Tag-Ordering Polling Protocols in RFID Systems

Yan Qiao, Shigang Chen, Tao Li, and Shiping Chen

***Abstract*—Future RFID technologies will go far beyond today's widely used passive tags. Battery-powered active tags are likely to gain more popularity due to their long operational ranges and richer on-tag resources. With integrated sensors, these tags can provide not only static identification numbers but also dynamic, real-time information such as sensor readings. This paper studies a general problem of how to design efficient polling protocols to collect such real-time information from a subset M of tags in a large RFID system. We show that the standard, straightforward polling design is not energy-efficient because each tag has to continuously monitor the wireless channel and receive $O(|M|)$ tag IDs, which is energy-consuming. Existing work is able to cut the amount of data each tag has to receive by half through a coding design. In this paper, we propose a tag-ordering polling protocol (TOP) that can reduce per-tag energy consumption by more than an order of magnitude. We also reveal an energy-time tradeoff in the protocol design: per-tag energy consumption can be reduced to $O(1)$ at the expense of longer execution time of the protocol. We then apply partitioned Bloom filters to enhance the performance of TOP, such that it can achieve much better energy efficiency without degradation in protocol execution time. Finally, we show how to configure the new protocols for time-constrained energy minimization.**

***Index Terms*—Energy efficiency, polling protocol, radio frequency identification (RFID), time efficiency.**

I. Introduction

TRADITIONAL barcodes can only be read in close ranges. RFID tags replace barcodes with electronic circuits that can transmit identification numbers wirelessly over a distance. The longer operational range makes them popular in transportation [1], object tracking [2]–[4], and supply chain management [5], [6]. A typical RFID system consists of one or multiple readers and numerous tags. Each tag carries a unique identifier (ID). Tags do not communicate amongst themselves; they communicate directly with the reader.

Passive tags are most widely used today. They are cheap, but do not have internal power sources. They rely on radio waves emitted from the reader for power, and have small operational ranges of a few meters, which seriously limit their applicability. For example, consider a large warehouse in a distribution center of a major retailer, where hundreds of thousands of tagged commercial products are stored. In such an indoor environment, if we use passive tags, hundreds of RFID readers may have to be installed in order to access tags in the whole area, which is not only costly but also causes interference when nearby readers communicate with their tags simultaneously. It is not a good solution to use a mobile reader and walk through the whole area whenever we need information from tags. To automate warehouse management in large scale, a much better choice is to use battery-powered active tags because of their long transmission ranges. The lifetime of these tags is determined by how their battery power is used. Energy conservation must be one of the top priorities in any protocol design that involves active tags.[1]

With richer on-tag resources, active tags are likely to gain more popularity in the future, particularly when their prices drop over time as manufactural technologies are improved and markets are expanded. These tags may be integrated with miniaturized sensors [7]–[12]. Not only will they report their IDs, but also they can report dynamic, real-time information about the operation status of the tags or the conditions of the environment.

This paper studies a general problem of how to design efficient polling protocols to collect information from a subset of tags in a large RFID system. For example, consider a large chilled food storage facility, where each food item is attached with an RFID tag that has a thermal sensor. An RFID reader periodically collects temperature readings from tags to check whether any area is too hot (or too cold), which may cause food spoil (or energy waste).[2] Because each area in the facility may be packed with many food items, the temperature readings from these close-by tags are highly redundant. Hence, it is not necessary for the reader to collect information from all tags in the system. The reader may select a subset of tags each time to collect temperature information. In the second example, an RFID reader periodically accesses the residual energy levels of on-tag batteries to see if some tags (or their batteries) need to be replaced. If the reader has information about which tags are new and which ones are old, it may choose to only query the old tags. In the third example, tags may be attached to cell phones/tablets/laptops in a large retail store, military weapons in a big warehouse, or medical equipment in a hospital. We may periodically query for the existence of the items. However, querying all items each time can be time and energy consuming and may even be unnecessary. One strategy is to query more important or expensive items at high frequency, which means a subset of items may be queried each time. As we will see

Manuscript received July 25, 2014; revised February 13, 2015; accepted March 24, 2015; approved by IEEE/ACM Transactions on Networking Editor Y. Liu. Date of publication May 05, 2015; date of current version June 14, 2016. This work was supported in part by the National Science Foundation under Grants CNS-1409797 and CNS-1115548.

Y. Qiao and T. Li were with the University of Florida, Gainesville, FL, USA, and are now with Google Inc., Mountain View, CA, USA.

Shigang Chen is with the University of Florida, Gainesville, FL, USA (e-mail: sgchen@cise.ufl.edu).

Shiping Chen is with the University of Shanghai for Science and Technology, Shanghai, China.

Digital Object Identifier 10.1109/TNET.2015.2423311

[1]Passive tags are powered by the reader's signal and thus their energy consumption is less of a concern, but their ranges are short.

[2]If a tag reports an abnormal temperature, the reader may instruct the tag to keep transmitting beacons, which guide a mobile signal detector to locate the tag.

later in this paper, it costs less energy to query a smaller number of tags. On the other hand, it is a harder problem to collect information from a *subset* of tags than from *all* tags because the reader has to make sure that tags that are not under query do not transmit—their transmissions will interfere with the transmissions made by tags of interest, causing unnecessary energy waste.

Much existing research focused on designing *ID-collection protocols* that read IDs from all tags in an RFID system [13]–[24]. In recent years, some interest is shifted to other functions such as estimating the number of tags in a system [2], [13], [25]–[32], detecting the missing tags [33]–[36] or misplaced tags [37], continuous monitoring [38], [39], etc. The primary performance objective in most papers is to minimize the execution time to read all tag IDs or perform other functions. Energy efficiency, particularly, how to reduce energy consumption by the tags, is an under-studied subject. There exists prior work on energy-efficient protocols for estimating the number of tags [40], or anti-collision protocols that minimize the energy consumption of a mobile reader [41], [42]. To the best of our knowledge, we were the first to investigate energy-efficient polling protocols for collecting information from tags in a large RFID system.

In this paper, we first show that the standard, straightforward polling design is not energy-efficient because each tag has to continuously monitor the wireless channel and receive $O(m)$ tag IDs, which is energy-consuming if the number m of tags that the reader needs to collect information from is large. We then show that a coded polling protocol (CP) is able to cut the amount of data each tag has to receive by half, which means that energy consumption per tag is also reduced by half. This is still far away from our objective of reducing energy consumption to $O(1)$. We propose a novel tag-ordering polling protocol (TOP) that can reduce per-tag energy consumption by more than an order of magnitude when comparing with the coded polling protocol. We also reveal an energy-time tradeoff in the protocol design: per-tag energy consumption can be reduced to $O(1)$ at the expense of longer protocol execution time. We then apply partitioned Bloom filters to enhance the performance of TOP, such that it can achieve much better energy efficiency without degradation in protocol execution time. Finally, we show how to configure the new protocols for time-constrained energy minimization.

The rest of the paper is organized as follows. Section II gives the system model and the problem statement. Section III describes a baseline protocol. Section IV gives a coded polling protocol. Sections V–VI propose a new energy-efficient polling protocol and analyze its performance. Sections VII–VIII use partitioned Bloom filters to design an enhanced polling protocol. Section IX discusses possible practical issues implementation of the proposed protocols. Section X presents numerical results. Section XI investigates the related work. Section XII gives the conclusion.

II. System Model and Problem Statement

A. RFID System

We consider a large RFID system using active tags. Each tag carries a unique ID and one or more sensors. It also has the capability of performing certain computations as well as communicating with the RFID reader wirelessly. The reader and the tags transmit with sufficient power such that they can communicate over a long distance. We assume that the RFID reader knows the IDs of all tags in the system by executing an ID-collection protocol, and it has enough power supply.

Unlike passive tags, there is currently no widely accepted standard for active tags. These tags come in great variety of performance specifications. Their prices can range from over \$100 to a few dollars and even lower; for example, at thickness of 7.5 mm, they can be as cheap as \$0.1–\$0.5 (depending on the purchasing quantities), working at 2.4 GHz with 80 m reading distance and a reading rate of over 100 tags per second, as well as up to 1 year battery life (quoted from alibaba.com). Active tags may carry internal batteries (typically small lithium ones), harvest and store energy from the surrounding environment through solar, thermal or piezoelectric means [43], [44], or be wirelessly rechargeable, e.g., using technologies similar to that adopted by the Moo tags [45].

Active tags use their own energy to transmit. A longer reading range can be achieved by transmitting at higher power. They are also richer in resources for implementing advanced functions. Their relatively higher price becomes less of a concern if they are used for expensive merchandizes or reused many times. Moreover, the price is expected to fall if large quantities are used as we have already witnessed (see the example in the previous paragraph).

But active tags also have a problem. They are powered by batteries. Replacing batteries for tens of thousands of tags is a laborious operation, considering that the tagged products may stack up, making tags not easily accessible. Even though today's commercial active tags often claim lifetime of several years, they assume that tags are idle most of the time and little data are exchanged between tags and readers for each communication, which is not truly in this paper where every tag may have to receive tens of thousands of IDs from a reader if a traditional polling protocol is used. Note that a tag's lifetime is determined by how often and how much the tags have to communicate with the reader, and the lifetime will drastically decrease with extensive use. Therefore, in the context of this paper, a new polling protocol that can reduce communication volume by an order of magnitude or more and thus lengthen the tag lifetime will be welcome. This is particularly true if tags are designed to harvest energy from environment or sporadically charged wirelessly. Since the time between energy replenishment can vary significantly, energy conservation becomes critical to ensure uninterrupted operation.

B. Problem Statement

Let N be the set of tags in the system and $n = |N|$. Let M be a subset of tags, $m = |M|$, and $M \subseteq N$. Our objective is to design efficient polling protocols that collect information from tags in M. A polling protocol may be scheduled to execute periodically. M may change over time so that different subsets of tags are queried, but the reader always knows the IDs of tags in M before each polling; otherwise it is impossible to differentiate between a tag in M and a tag in N-M. How to determine M based on N is out of the scope of this paper.

Collecting information from a subset M is different from doing so from all tags in [11]. In fact, most of our effort in this paper is to find an efficient way to *inform* tags in M to report their information and *tell* others to stay out, such that the amount of data received/sent by each tag is $O(1)$, instead of $O(m)$ in a traditional polling protocol, which is very energy-expensive in informing the tags about who are in the subset M. The work in [11] does not need to address the issue of informing tags in M to report, and thus their method cannot be applied to solve our problem.

C. Performance Objectives

We have two performance objectives. The primary performance objective is to achieve *energy efficiency*. We want to minimize the average amount of energy that a tag spends during one execution of a polling protocol. The energy expenditure by a tag has two components: 1) energy for transmitting its information (e.g., 32 bits) to the reader, and 2) energy for receiving the polling request and other information from the reader. The former is a small, fixed amount of energy that must be spent. The latter is dependent on the protocol design as we will see shortly. It is a variable amount of energy that should be minimized. Simple protocol designs will result in all tags in the system, including those not in M, to be continuously active and unnecessarily receive a large amount of data from the reader for an extended period of time. How to minimize such energy cost is the focus of this paper.

Our secondary performance objective is to reduce *protocol execution time*. RFID systems use low-rate communication channels. For example, in the Philips I-Code system, the rate from a reader to a tag is about 27 Kbps and the rate from a tag to a reader is about 53 Kbps. Low rates, coupled with a large number of tags, often cause long execution times for RFID protocols. To apply such protocols in a busy warehouse environment, it is desirable to reduce protocol execution time as much as possible.

D. Communication Model

Communication between the reader and tags is time-slotted. The reader's signal synchronizes the clocks of tags. Let $t_{\rm tag}$ be the length of a time slot during which the reader is able to broadcast a tag ID, and $t_{\rm inf}$ be the length of a time slot during which a tag is able to transmit its information.

Sensor motes can communicate with each other. This is fundamental to their ability of forming networks. Like passive tags, active tags do not communicate with each other. They all communicate directly with the reader. Tags try to lower their energy consumption by transmitting at low power. With a high-quality antenna, a reader is able to receive weak signals from tags. With low-quality antennas, although tags can receive strong signals from the reader, they cannot receive each other's weak signals. Therefore, coordination amongst tags through overhearing cannot be applied.

III. Basic Polling Protocol (BP)

In a standard, straightforward way of designing a polling protocol, we simply let the RFID reader broadcast the tag IDs in M one by one. After it transmits an ID, it waits for a time slot of $t_{\rm inf}$ during which the corresponding tag transmits its information. Each tag continuously listens to the wireless channel. Whenever it receives an ID from the reader, the tag compares the received ID with its own ID. If they match, the tag will transmit its information and then go to sleep until the next scheduled execution of the protocol.

In the above protocol, each tag in M will have to receive $\frac{m}{2}$ IDs on average from the reader before it transmits. Each tag not in M will have to receive all m IDs. The amount of energy spent by a tag in receiving such data grows linearly with respect to m. It takes a constant amount of energy for a tag to receive an ID and another constant amount of energy for it to transmit its information. The energy cost of the whole system is thus $O(nm)$. The protocol execution time is $m(t_{\rm tag}+t_{\rm inf})$.

We use a numerical example to explain the energy cost. Consider a military base that has a large warehouse storing 50,000 weapons, ammunition magazines, and other equipment, which are tagged with RFID tags. Among them, there are 1,000 sensitive devices, from which an RFID reader needs to access information in order to make sure that they are in good conditions or simply to confirm their presence (against unauthorized removal). Let e_r be the amount of energy a tag spends in receiving an ID and e_s be the amount of energy a tag spends in transmitting its information. The total energy consumed by all tags for transmitting is 1,000 e_s, and the total energy consumed by all tags for receiving is about 50,000,000 e_r. Even though e_r may be smaller than e_s, the total amount of energy spent by tags in receiving can be much greater than the amount spent in transmitting.

IV. Coded Polling Protocol

We show that a coded polling protocol (CP) [46] is able to reduce the amount of data each tag has to receive by half. The protocol assumes that each tag ID carries an *identification number* and a CRC (*cyclic redundancy code*) for error detection. This requirement is satisfied by the EPCglobal Gen-2 standard [47], [48], where each 96 bit tag ID contains a CRC checksum. The CRC is computed based on the identification number and a generator. When a tag receives an ID from a wireless channel, it computes a CRC based on the received identification number and then compares the result with the received CRC. If they are the same, we say the ID contains a *valid* CRC.

CRC has the following property: If x and y are two tag IDs with valid CRCs, then $x \oplus y$ also has a valid CRC. The same property does not hold for $x \oplus \hat{y}$, where $\hat{y}$ contains the same bits in y but in the reverse order. For example, if $y = 10110$, then $\hat{y} = 01101$. We call $\hat{y}$ the *reversal* of y.

In the coded polling protocol, the RFID reader first arranges the IDs in M in pairs. Each pair consists of two IDs that are arbitrarily selected from M. Consider an arbitrary pair, x and y, which are called each other's *paring ID*. We define the *polling code* of the pair as $c = x \oplus \hat{y}$.

Instead of sending out the IDs in M one after another, the reader broadcasts the polling code of each pair one after another. After each broadcast of a polling code $c = x \oplus \hat{y}$, the reader waits for two time slots, during which tag x and tag y will transmit. More specifically, when an arbitrary tag z receives the polling code c, it first computes $z \oplus c$, and checks whether the CRC in the reversal of $z \oplus c$ is valid. If it is, the tag will transmit its information. Otherwise, the tag computes $\hat{z} \oplus c$, and checks whether the CRC in $\hat{z} \oplus c$ is valid. Again, if it is valid, the tag will transmit. Otherwise, the tag will not transmit. We show that only tag x and tag y will transmit.

First, consider the case of $z = x$. The tag first computes $z \oplus c = x \oplus x \oplus \hat{y} = \hat{y}$. The reversal of $\hat{y}$ is y. The CRC in any tag ID (including y) is valid. Hence, tag x will transmit. Moreover, it now knows its pairing ID, y. If x is greater than y, the tag will transmit in the first slot after receiving the polling code; otherwise, it will transmit in the second slot.

Second, we consider the case of $z = y$. The tag first computes $y \oplus c = y \oplus x \oplus \hat{y}$. Its reversal is likely to have an invalid CRC; the chance for an arbitrary number to contain a valid CRC is very small. Then, the tag computes $\hat{z} \oplus c = \hat{y} \oplus x \oplus \hat{y} = x$, which contains a valid CRC. Consequently, y will transmit. Since it now knows its pairing ID, x, it also knows in which slot it should transmit.

Finally, consider the case of $z \neq x$ and $z \neq y$. The tag computes the reversal of $z \oplus c = z \oplus x \oplus \hat{y}$ and then computes $\hat{z} \oplus c = \hat{z} \oplus x \oplus \hat{y}$. Both of them are likely to have invalid CRCs.

A minor problem is that $y \oplus c$ in the second case and $z \oplus c$ or $\hat{z} \oplus c$ in the third case still have a small probability to contain a valid CRC. However, the reader can easily prevent this from happening. It knows all tag IDs. It can precompute all polling codes and check whether a valid CRC happens in the above cases by chance when it is not supposed to. If this is true for a pair of tags, x and y, the reader must break up the pair, and use them to form new pairs with other IDs in M. Such an approach is effective because the probability for this to happen is exceedingly small when CRC is sufficiently long.

Because each polling code represents two tag IDs, the number of polling codes in CP is $\frac{m}{2}$. Hence, when comparing with the basic polling protocol, CP reduces the number of broadcasts made by the reader by half, and it also reduces the amount of data that each tag has to receive by half. This not only saves energy for tags, but also reduces the protocol execution time to $\frac{m}{2}t_{\text{tag}} + mt_{\text{inf}}$.

V. Tag-Ordering Polling Protocol (TOP)

Although CP is more efficient, the expected amount of energy that each tag spends in receiving remains $O(m)$. In this section, we propose a new tag-ordering polling protocol that reduces such energy cost to $O(1)$.

A. Motivation

In the basic polling protocol, an RFID reader broadcasts m IDs in time slots of length t_{tag}. All tags must continuously monitor the wireless channel in order to know whether their own IDs are in the broadcast. In CP, the reader broadcasts $\frac{m}{2}$ polling codes also in time slots of length t_{tag}. Again, all tags must continuously monitor the wireless channel. They have to keep receiving and processing the polling codes. Each tag in the basic protocol has to receive up to m IDs. Even though CP is more efficient, a tag still has to receive up to $\frac{m}{2}$ codes.

We want to remove the necessity for any tag to keep monitoring the wireless channel. Ideally, a tag should stay in an energy-conserving standby mode for most of time, and only wake up at the right time slot to receive information about itself, such as whether it is polled and, if so, when it should transmit. To further reduce the amount of data that tags have to receive, we let the reader broadcast a so-called *reporting-order* vector V, instead of IDs in M. Each ID in M is mapped to a bit in V through a hash function; the bit is set as one to encode the ID in the vector. A tag only needs to check a specific bit in V at a location determined by the hash of its ID. This bit is called the *representative bit* of the tag. If its value is one, the tag is polled by the reader for reporting, i.e., the tag belongs to M; if its value is zero, the tag is not polled. The vector V also carries information about the order in which the polled tags will report their data. Each bit whose value is one in V represents a polled tag. If a tag finds that there are i ones in V preceding its representative bit, it knows that it should be the $(i+1)$th tag in M to report its information. With such an ordering, it becomes possible for tags in M to report at different times and avoid collision.

However, this basic idea has two problems. First, there should be at least m bits in V to encode m IDs in M. The energy cost of receiving V remains $O(m)$. How can a tag find out the number of ones in V preceding its representative bit without having to receive the whole vector? Second, hash collision causes two issues. If a tag not in M is hashed to the same bit in V as a tag in M does, it will find its representative bit to be one, causing false positive. If two tags in M are mapped to the same bit in V, they will transmit at the same time, causing report collision. In the rest of this section, we design a new tag-ordering polling protocol (TOP) to solve these problems. It consists of three phases: *ordering phase*, *polling phase*, and *reporting phase*. In the ordering phase, the reader broadcasts the vector V so that each tag knows whether it is polled and where it is located in the reporting order. The polling phase resolves the issues of false positive and report collision. Finally, in the reporting phase, tags in M report their information in the order defined by V without collision.

B. Protocol Description

1) Ordering Phase: The RFID reader does not broadcast any IDs or indices. It only broadcasts the reporting-order vector, V. If V cannot fit in one time slot of length t_{tag}, the reader breaks the vector into segments and broadcasts each segment in a time slot of t_{tag}. In addition, the reader also broadcasts the vector size v.

Knowing the vector size, a tag t is able to hash its ID and find out the location of its representative bit in V. Because the segment size is fixed, t also knows which segment its representative bit belongs to. This segment, denoted as V_t, is called the *representative segment* of tag t. A tag will stay in the standby mode and be active only when receiving its representative segment.

If a tag finds that its representative bit is zero, it knows for sure that it is not a member in M. If a tag finds that its representative bit is one, it may be a member in M or a non-member that is mapped to a bit which a member in M is also mapped to. The latter case causes *false positive*. Because the reader knows all IDs in the system, it can pre-compute the set F of non-member tags that cause false positive.

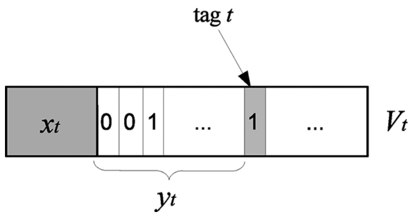


Fig. 1. V_t is the representative segment of tag t, x_t is the total number of ones in all previous segments, and y_t is the number of ones in V_t that precede tag t's representative bit. I_t is the position of t in the reporting order. $I_t = x_t + y_t$.

When the reader broadcasts any segment of V, it includes in the same time slot *the total number of ones in the previous segments*. For an arbitrary tag t, let I_t be the number of ones in V preceding the representative bit of t. When tag t receives V_t, it can computes I_t as the sum of (a) the number of ones in the previous segments and (b) the number of ones in V_t before its representative bit. See Fig. 1 for illustration. As we will see later, the value of I_t specifies when tag t will transmit during the reporting phase.

If two tags in M are mapped to the same bit in V, they will have the same I_t value and thus transmit at the same time during the reporting phase, causing collision. Because the reader has all IDs in M, it knows exactly which tags will be mapped to the same bit. This makes it easy to resolve collision. The reader simply removes all but one tag that are mapped to a bit, and puts them in a set C. These tags, together with tags in F, will not participate in the reporting phase. They are handled separately in the polling phase.

2) Polling Phase: In this phase, the reader issues two types of polling requests. For each tag in C, it sends a *positive polling request*. For each tag in F, it sends a *negative polling request*. To distinguish these two types, the reader must transmit a one bit flag together with a tag ID in each request, specifying whether the polling is positive or negative and which tag is polled.

Tags that find their representative bits to be ones in the previous phase must continuously listen to the channel during the polling phase. After sending a positive request, the reader waits for a time slot to receive information. The tag that finds its ID in the request will transmit its information in this slot. This tag, which belongs to C, will not participate in the reporting phase. After sending a negative request, the reader does not wait before sending out the next request. The tag that finds its ID in a negative request knows that it must belong to F and hence should not further participate in the protocol execution.

The total number of polling requests is $|F|+|C|$. By choosing an appropriate size for the reporting-order vector, we can make sure that $|F| + |C| = O(1)$ (see Section VI). Note that only tags in M and F have to listen to the channel in this phase. Tags in $N - M - F$, which may contain the majority of tags in the system, have already known that they do not belong to M and thus do not need to participate in the protocol execution.

3) Reporting Phase: A tag participates in the reporting phase only if it satisfies the following two conditions: (1) it finds that its representative bit is one in the ordering phase, and (2) it does not find its ID in the requests of the polling phase.

The reporting phase consists of $m - |C|$ time slots. In each time slot, one tag in $M - C$ transmits its information. Recall that each tag in M learns its index in the reporting order during the ordering phase. The tag will transmit in the reporting phase at the time slot of the same index.

4) Timing: Before executing the protocol, the RFID reader uses its broadcasting signal to synchronize the clocks of the tags. The reader computes the vector V and breaks it into segments. Suppose each time slot of length t_{tag} can carry 96 bits. We may set the segment size to be 80 bits and use the remaining 16 bits to carry the total number of ones in the previous segments.[3] The reader is able to compute the execution time T_1 of the ordering phase, which is the number of segments multiplied by t_{tag}.

Since the reader knows all IDs in the system, it can precompute the set F of tags that cause false positive and the set C of tags that should not participate in the reporting phase in order to avoid collision. Based on F and C, the reader can compute the execution time T_2 of the polling phase, which is $|F| \times t_{\text{tag}} + |C| \times (t_{\text{tag}} + t_{\text{inf}})$.

Suppose all tags wake up at each scheduled execution of the protocol. The reader computes and broadcasts the values of T_1 and T_2 right before the ordering phase, so that the tags know when each phase of the protocol will begin. They will remain in the standby mode unless they have to receive their representative segments, participate in the polling phase, or transmit their information in the reporting phase.

If the system requires on-demand polling of tag information instead of periodic execution, there are two possible solutions to wake the tags up in the first place. The first one is "pseudo-on-demand" polling, where tags still wake up periodically, but the reader only issues the polling request when needed. The second approach is to attach a wake-up circuit to each tag, and use the two-stage wake-up scheme proposed in [49] to activate the tags. In this approach, tags respond almost immediately to the polling event. However, the wake-up circuit requires the reader to be close enough so that the radio power is strong enough to trigger the wake-up event. As a result, we may have to deploy extra readers to cover all the tags.

VI. Performance Analysis of TOP

A. Energy Cost

We show how to configure TOP such that the energy cost per tag is $O(1)$. The energy cost of a tag has four components: 1) receiving v, T_1 and T_2, 2) receiving a segment of V in the ordering phase, 3) listening to the channel during the polling phase, and 4) transmitting information in a slot at the reporting phase (or at the polling phase if the tag is in C). The first two components incur small, constant energy expenditure to every tag in the system. The fourth component also incurs small, constant energy cost, but only to the tags in M. The third component incurs energy cost only to tags in F and M. In the **worst** case, a tag has to listen to all $|C|+|F|$ polling requests from the reader. Suppose it takes one unit of energy to receive a polling request. The total energy cost of a tag, denoted as Ω, is

$$\Omega \leq |C| + |F| + O(1). \tag{1}$$

[3]Using 16 bit to carry the number of ones in previous segments will limit the value of m to (0, 65,535]. To get rid of this limitation, we can use $\lceil \log_2 m \rceil$ bits instead and broadcast the value of $\lceil \log_2 m \rceil$ to tags at the beginning of protocol. However, for the sake of simplicity, we use 16 bits in this paper to help demonstrate the main idea.

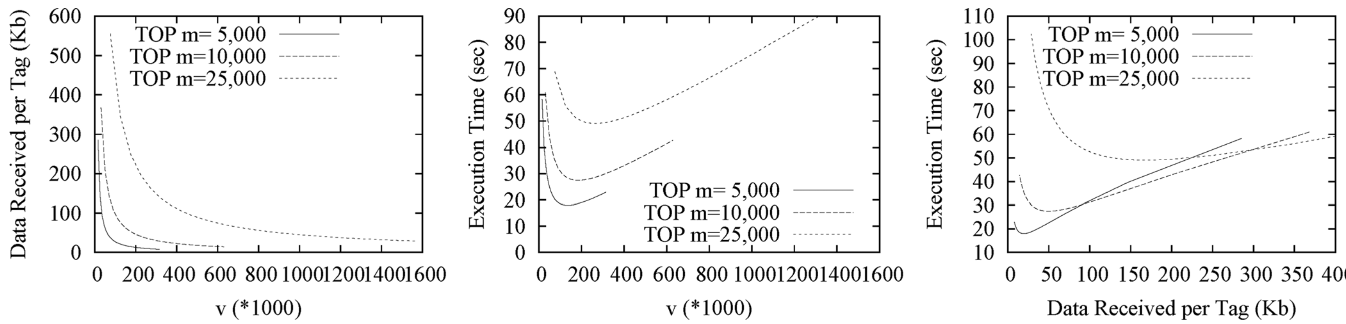


Fig. 2. First plot: Energy cost per tag with respect to v. Second plot: Protocol execution time with respect to v. Third plot: Energy-time tradeoff controlled by v.

We treat $|C|$ and $|F|$ as random variables and derive their expected values. Recall that v be the number of bits in the reporting-order vector V. Let b_i be the value of the ith bit in $V, 0 \leq i < v$. For each tag in M, the reader maps it to a random bit in V and sets the bit to one. After encoding all m tags in V, the probability for b_i to be one is

$$\text{Prob}\{b_i = 1\} = 1 - \left(1 - \frac{1}{v}\right)^m \approx 1 - e^{-m/v}. \quad (2)$$

The bits, $b_0, b_2, \ldots, b_{v-1}$, are independent of each other. Thus, the expected number of ones in V is $\sum_{i=1}^{v} Prob\{b_i = 1\}$. The value of $|C|$ is equal to m subtracted by the number of ones in V. Hence, we have

$$E(|C|) = m - \sum_{i=1}^{v} \text{Prob}\{b_i = 1\} \approx m - v(1 - e^{-m/v}). \quad (3)$$

A tag not in M will cause false positive when its representative bit is one. The probability for this to happen is $\text{Prob}\{b_i = 1\}$. Hence,

$$E(|F|) = (n - m)\text{Prob}\{b_i = 1\} \approx (n - m)(1 - e^{-m/v}). \quad (4)$$

Both $E(|C|)$ and $E(|F|)$ are monotonically decreasing functions of v. We show that $E(|C|) = O(1)$ if v is sufficiently large. Let $v = \frac{m^2}{2}$. From Taylor expansion, we know that

$$\begin{aligned} 1 - e^{-m/v} &= \frac{m}{v} - \frac{1}{2!}\left(\frac{m}{v}\right)^2 + \frac{1}{3!}\left(\frac{m}{v}\right)^3 - \frac{1}{4!}\left(\frac{m}{v}\right)^4 \ldots \\ &\geq \frac{m}{v} - \frac{1}{2!}\left(\frac{m}{v}\right)^2. \end{aligned}$$

Applying it to (3), we have

$$E(|C|) = m - v(1 - e^{-m/v}) \leq \frac{1}{2!}\frac{m^2}{v} = 1. \quad (5)$$

Next we show that $E(|F|) = O(1)$ if v is sufficiently large. If $n = m, E(|F|) = 0$. Now assume $n > m$. Let $v = -\frac{m}{\ln(1 - \frac{1}{n-m})}$. Applying it to (4), we have

$$E(|F|) = (n - m)(1 - e^{-m/v}) = 1. \quad (6)$$

Therefore, if we choose $v = \max\{\frac{m^2}{2}, -\frac{m}{\ln(1-\frac{1}{n-m})}\}$, we have

$$E(\Omega) \leq E(|C|) + E(|F|) + O(1) \leq 1 + 1 + O(1) = O(1).$$

We conclude that TOP can be configured such that the expected energy cost per tag is $O(1)$. As we will see shortly, the protocol execution time increases when v becomes too large. To strike a balance between energy cost and protocol execution time, we may choose a value of v much smaller than $\max\{\frac{m^2}{2}, -\frac{m}{\ln(1-\frac{1}{n-m})}\}$. In Section X, we use simulations to study the performance of TOP under practical values of v. For example, when $v = 24$ m, the amount of data that a tag receives in TOP is more than an order of magnitude smaller than what a tag has to receive in CP.

We characterize the energy cost in the polling phase by counting the amount of data (in Kilobits) that a tag has to receive. Numerical results are shown in the first plot of Fig. 2, where $n = 50{,}000$ and $m = 5{,}000$, 10,000, or 25,000, corresponding to three curves in the plot. Clearly, as v increases, the energy cost decreases.

B. Execution Time

The protocol execution time also consists of four components. To begin with, it takes the reader a small, constant time to broadcast v, T_1 and T_2. The time for the ordering phase is $\frac{v}{l}t_{\text{tag}}$, where l is the segment size. The time for the polling phase is $|F| \times t_{\text{tag}} + |C| \times (t_{\text{tag}} + t_{\text{inf}})$. The time for the reporting phase is $|M - C| \times t_{\text{inf}}$. Hence, the total execution time is

$$T = \left(\frac{v}{l} + |F| + |C|\right) t_{\text{tag}} + m \times t_{\text{inf}} + O(1). \quad (7)$$

From (3) and (4), the expected protocol execution time is

$$\begin{aligned} E(T) &= \left[\frac{v}{l} + (n - m)(1 - e^{-m/v}) + m \right. \\ &\quad \left. - v(1 - e^{-m/v})\right] t_{\text{tag}} + m \cdot t_{\text{inf}} + O(1) \\ &\approx \left[\frac{v}{l} + \frac{(n - m)m}{v}\right] t_{\text{tag}} + m \cdot t_{\text{inf}} + O(1). \quad (8) \end{aligned}$$

The second plot of Fig. 2 presents the protocol execution time (excluding the constant $O(1)$) when $n = 50{,}000$, $m = 5{,}000$, 10,000, or 25,000, $t_{\text{tag}} = 3297$ μs, and $t_{\text{inf}} = 906$ μs; see Section X for how they are determined. Interestingly, as v increases, the execution time first decreases and then increases. We can find the optimal value of v that minimizes the execution time from $\frac{\delta E(T)}{\delta v} = 0$.

Combining the results in the first and second plots, we can figure out the *tradeoff relation* between energy cost and protocol execution time, which is presented in the third plot. As v becomes large, the energy cost decreases at the expense of increased execution time.

C. Choosing v for Time-Constrained Energy Minimization

Recall the performance objectives of TOP are energy efficiency and time efficiency. However, as shown in Fig. 2, we may not be able to achieve the best performance in both metrics using one configuration. Below we study how to configure TOP for time-constrained energy minimization.

Consider a warehouse with a large number of RFID-tagged goods. Suppose the system administer wants to maximize the tags' battery lifetime, but there is a requirement on the execution time of a polling operation because excessively long execution time increases the chance of interfering with other scheduled tasks. From the previous analysis, we know that the protocol execution time is treated as a random variable. Let T be the execution time of TOP, B be a pre-defined time bound, and α be a probability value, $0 < \alpha < 1$. The time constraint can be specified in a probabilistic way,

$$\mathrm{Prob}\{T \leq B\} \geq \alpha. \tag{9}$$

Our performance objective is to find the optimal value of v that minimizes the energy cost, subject to the above constraint.

As shown in the first plot of Fig. 2, the energy cost decreases as the size of the reporting-order vector, v, increases. Hence, our goal becomes finding the largest v that satisfies (9). In the following, we derive $\mathrm{Prob}\{T \leq B\}$ as a function of v. Based on this function, we will be able to compute the optimal value of v.

Let d be the total number of ones in V after encoding tags in $M, 0 < d \leq m$. The probability that x bits are ones, expressed as $\mathrm{Prob}\{d = x\}$, can be calculated by the *balls and bins algorithm*, which will be given in the next subsection. For now we denote the function for computing $\mathrm{Prob}\{d = x\}$ as $p_d(m, v, x)$.

After encoding tags in M, the reader removes colliding tags to C. The value of $|C|$ is equal to m subtracted by the number of ones in V. Hence,

$$\mathrm{Prob}\{|C| = c\} = \mathrm{Prob}\{d = m - c\} = p_d(m, v, m - c). \tag{10}$$

When a tag not in M is mapped to a bit that is one, false positive happens. The reader puts all false positive tags to F. When there are x bits that are ones in V, the conditional false positive probability is $\frac{x}{v}$. Thus,

$$\mathrm{Prob}\left\{\text{false positive}|\, d = x\right\} = \frac{x}{v}.$$

Obviously, when $d = x$, the total number of false positive tags follows a binomial distribution $\mathrm{Bino}\left(n - m, \frac{x}{v}\right)$.

$$\mathrm{Prob}\left\{|F| = f|\, d = x\right\} = \binom{n-m}{f}\left(\frac{x}{v}\right)^f\left(1 - \frac{x}{v}\right)^{n-m-f}.$$

Let S be the union of C and F, so $|S| = |C| + |F|$. The probability distribution of $|S|$ is

$$\begin{aligned}\mathrm{Prob}&\left\{|S| = s\right\}\\ &= \sum_{c=0}^{s}\mathrm{Prob}\left\{|F| = s - c|\, |C| = c\right\}\cdot\mathrm{Prob}\left\{|C| = c\right\}\\ &= \sum_{x=1}^{m}\mathrm{Prob}\left\{|F| = s - m + x|d = x\right\}\cdot\mathrm{Prob}\{d = x\}\\ &= \sum_{x=1}^{m}\binom{n-m}{s-m+x}\left(\frac{x}{v}\right)^{s-m+x}\\ &\quad\times\left(1 - \frac{x}{v}\right)^{n-s-x} p_d(m, v, x).\end{aligned} \tag{11}$$

Adopt (7) and ignore $O(1)$, a small constant time for the reader to broadcast v, T_1 and T_2, which is negligibly small when comparing with other components on the right side of (7). We have

$$\mathrm{Prob}\{T \leq B\} = \sum_{s=0}^{s_{\max}}\mathrm{Prob}\{|S| = s\} \tag{12}$$

where $s_{\max} = \frac{B - mt_{\mathrm{inf}}}{t_{\mathrm{tag}}} - \frac{v}{l}$. We denote the right side of (12) as $P_t(v, B)$, which is the probability for the protocol execution time to be bounded by B under a certain value of v. It is computable as a function of v and B after (11) is applied and parameters m and n are given.

We want to find the largest value of v that satisfies the inequality, $P_t(v, B) \geq \alpha$. Our numerical computation shows that, given a fixed value of $B, P_t(v, B)$ is not a monotonic function with respect to v. Hence, we cannot directly apply the bisection search method to find the largest v that satisfies $P_t(v, B) \geq \alpha$. We may use the False Position algorithm [50] to find the optimal value of v. The computation overhead is reasonable. For $n =$ 10,000, $m =$ 1,000, $B = 4$ seconds, and $\alpha = 99\%$, it takes an Apple MacBook Pro (2.4 GHz CPU and 4 GB memory) 3 seconds to find the optimal $v = 60,160$. And for $n =$ 10,000, $m =$ 1,000, $B = 3$ seconds, and $\alpha = 99\%$, it takes the same computer 16 seconds to find that no v can satisfy the requirement, because $B = 3$ seconds is smaller than the minimum execution time that TOP can achieve.

As a related problem, if v and α are given, we can also use $P_t(v, B)$ to compute the time bound that TOP can achieve. More specifically, given a value of v, we are able to find the smallest B that satisfies $P_t(v, B) \geq \alpha$ through bi-section search: Recall that $P_t(v, B)$ is the formula for $\mathrm{Prob}\{T \leq B\}$, the probability for the protocol execution time to be bounded by B. Clearly, it is an increasing function of B with $P_t(v, 0) = 0$ and $P_t(v, +\infty) = 1$. We choose a small value B_1 (e.g., 0) such that $P_t(v, B_1) < \alpha$ and a large value B_2 such that $P_t(v, B_2) \geq \alpha$. Let $B_3 = \lceil\frac{B_1 + B_2}{2}\rceil$. If $P_t(v, B_3) < \alpha$, assign B_3 to B_1; otherwise, assign B_3 to B_3. Hence, the search range $[B_1, B_2]$ is cut by half. Repeat the above process until $B_1 = B_2$, which gives the smallest bound B that satisfies $P_t(v, B) \geq \alpha$. Let $n =$ 10,000 and $m =$ 1,000. Fig. 3 shows the smallest bound B with respect to v when $\alpha = 90\%$, 95% and 99%, which correspond to the three curves in the figure.

D. Computing $p_d(m, v, x)$—The Balls and Bins Algorithm

Problem: Suppose we throw m balls into v empty bins. Each ball is thrown to a random bin, and each bin can hold unlimited number of balls. We want to find the probability that after m balls are thrown, x bins are not empty, denoted as $p_d(m, v, x)$.

Solution: There are many solutions to this problem. We now provide a recursive one. Assume after we throw m balls, there are x non-empty bins, $1 \leq x \leq m$. When $x > 1$, there are two possibilities of where the mth ball goes: (1) If the mth ball is placed to a previously empty bin, there should be $x - 1$ non-

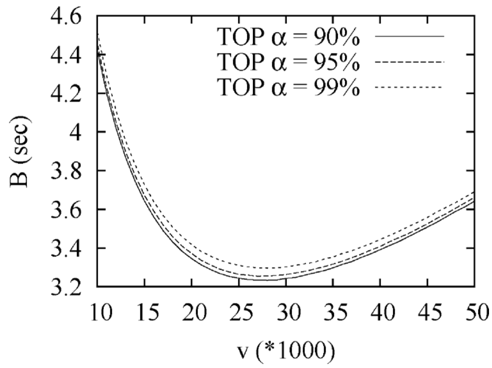


Fig. 3. Bound B that satisfies $\text{Prob}\{T \le B\} \ge \alpha$ with respect to v. Parameters: $n = 10,000, m = 1,000$.

empty bins after $m-1$ balls were thrown, and the possibility for this to happen is $\frac{v-x+1}{v}$; (2) Otherwise if the mth ball goes to a previously non-empty bin, there must be x non-empty bins after $m-1$ balls were thrown, and the possibility of this option is $\frac{x}{v}$. Thus,

$$p_d(m,v,x) = \begin{cases} 1; x = m = 1. \\ \frac{x}{v}p_d(m-1,v,x) + \frac{v-x+1}{v}p_d(m-1,v,x-1); \\ 1 \le x \le m \text{ and } x \le v. \\ 0; \text{all other cases.} \end{cases}$$

$p_d(m,v,x)$ can be calculated from simple dynamic programing.

VII. Enhanced Tag-Ordering Polling Protocol (ETOP)

A. Motivation

If we do not want to significantly increase execution time, we cannot choose a large value for v. In this case, we must find other means to lower energy cost. The key is to reduce the number of IDs that have to be transmitted in the polling phase. Namely, we should reduce the number of tags in F and C. Let's first focus our discussion on false positive. Consider an arbitrary tag $t \notin M$. Its representative segment is V_t. Let q be the number of tags in M that are also mapped to V_t. False positive occurs if t and one of those q tags have the same representative bit. The probability for this to happen is $1-(1-\frac{1}{l})^q$, where l is the number of bits in V_t.

To further reduce the false-positive probability, we can implement each segment of V as a Bloom filter [51], [52]. The reader uses multiple hash functions to map each tag to $k(>1)$ *representative bits* in V, instead of just one in TOP. More specifically, for each member $t' \in M$, the reader first maps it to a representative segment $V_{t'}$ through a hash function whose range is $[0, \frac{v}{l})$. Then the reader further maps t' to k representative bits in $V_{t'}$ and set them to ones.

After all members in M are encoded in the segments of V, the reader broadcasts the segments in the ordering phase. A tag t only listens for its representative segment V_t and then checks its representative bits. If any representative bit is zero, the tag **cannot** be in M. If all representative bits are ones, the tag may be a member in M or a false positive. In the case of false positive, even though the tag does not belong to M, every one of its representative bits is set because it is also a representative bit of a member tag in M. The probability for this to happen is

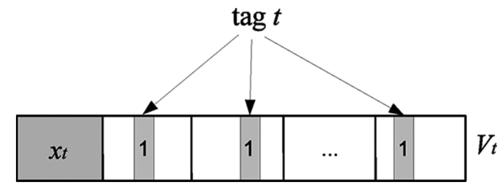


Fig. 4. V_t is the representative segment of tag t. V_t is evenly divided into k partitions, each having $\lfloor \frac{l}{k} \rfloor$ bits. Tag t has one representative bit in every partition.

$\left(1-\left(1-\frac{1}{l}\right)^{kq}\right)^k$, where q is the number of tags in M whose representative segments are also V_t. For example, if $l = 80, k = 3$, and $q = 2$, the false-positive probability is just 3.8×10^{-4}, much lower than $1-\left(1-\frac{1}{l}\right)^q = 2.5 \times 10^{-2}$ in TOP under the same parameters.

Bloom filters can reduce the false-positive probability. But it is more difficult to use them to carry the reporting order, based on which the tags will take turn to transmit during the reporting phase. In TOP, we use the number of ones that precede the representative bit of a tag to determine the tag's position in the reporting order. Bloom filters use multiple representative bits to encode each member. The representative bits of different members may overlap in an arbitrary way. Hence, we cannot simply use all bits whose values are ones to represent tags in M because there is no one-to-one mapping between them.

In the following, we design an *enhanced tag-ordering polling protocol* (ETOP) to solve the above problem. ETOP uses *partitioned Bloom filters*, which not only reduce false positive and encode the reporting order, but also reduce $|C|$ as well as overall execution time of the protocol.

B. Protocol Description

The main difference between ETOP and TOP is that ETOP implements each segment of V as a partitioned Bloom filter instead of a simple bit array. When we describe the protocol of ETOP, we focus on the difference while omitting the details that it shares in common with TOP.

In a partitioned Bloom filter, the l bits of a segment are evenly divided into k partitions. Each partition has $\lfloor \frac{l}{k} \rfloor$ bits. See Fig. 4 for illustration. For every member tag t in M, the reader applies a hash function on its ID to obtain a number of hash bits. The reader uses $\lceil \log_2 v \rceil$ hash bits to map t to a representative segment V_t, and then uses $k \lceil \log_2 \frac{l}{k} \rceil$ hash bits to further map t to one representative bit in every partition of the segment. Like a classical Bloom filter, the partitioned Bloom filter sets k representative bits for each encoded member; unlike a classical Bloom filter, a partitioned Bloom filter spreads the k representative bits in k different partitions.

After receiving its representative segment, a tag checks the k representative bits to determine if it is a member of M. False positive cases are handled by the reader in the polling phase as usual.

How does a tag t know its position in the reporting order? First we consider the reporting order among tags that are encoded in the same segment V_t. Since every tag has exactly one representative bit in each partition of V_t, we may be able to use one of the partitions to carry the order information. In other words, if there is a partition P^* whose number of ones is equal to the number of tags encoded in V_t, we know that there must be a one-to-one mapping between these tags and the '1' bits in P^*. We can use the order of '1' bits in P^* as the reporting order of the corresponding tags. We will explain later how the reader makes sure that such a partition exists. When the reader sends out V_t, in the same time slot it also sends the total number x_t of tags that are encoded in all previous segments of V. The position of tag t in the reporting order can be computed from x_t and the information in P^*, which we will further explain shortly.

How to make sure that any segment of V always has a partition whose number of ones is equal to the number of tags encoded in the segment? The reader has to do some extra work. After encoding all tags in M, the reader examines the partitions one by one for each segment. If there is not such a partition, the reader removes an encoded tag and places it in the set C, which will be explicitly polled in the polling phase. The reader keeps removing tags until it finds a partition that satisfies the above requirement. Note that the requirement is always satisfied when the number of tags encoded in a segment is one.

After receiving its representative segment V_t, a tag $t \in M$ computes its position in the reporting order as follows: It finds out a partition P^* in V_t that has the largest number of ones. This partition must have a one-to-one mapping between '1' bits and encoded tags. Let y_t be the number of ones in P^* that precedes the representative bit of t. The tag computes its position in the reporting order as $y_t + x_t$. Recall that x_t is the number of tags that are encoded in the previous segments. It is received together with V_t in the same time slot.

The polling phase and the reporting phase of ETOP are identical to their counterparts in TOP.

VIII. Performance Analysis of ETOP

A. Energy Cost

We show that ETOP can be configured such that the energy cost per tag is $O(1)$. ETOP has the same upper bound formula for per-tag energy cost as TOP does, which is shown in (1), but it has different values of $|C|$ and $|F|$. In the following, we derive $|C|$ and $|F|$ for ETOP. Let m_i be the number of tags in M that are encoded in the ith segment, $0 \le i < \frac{v}{l}$. Each tag in M has a probability of $\frac{l}{v}$ to be mapped to the ith segment. Hence, m_i follows a binomial distribution Bino $\left(m, \frac{l}{v}\right)$.

$$\text{Prob}\{m_i = x\} = \binom{m}{x}\left(\frac{l}{v}\right)^x\left(1-\frac{l}{v}\right)^{m-x}. \tag{13}$$

Let C_i be a subset of C, containing the tags that are removed from the ith segment. We know the following facts: (1) When $m_i = 0, |C_i| = 0$. (2) When $m_i = 1, |C_i| = 0$. (3) When $m_i \ge 1, |C_i| \le m_i - 1$. Hence, we must have

$$\begin{aligned} E\left(|C_i|\right) &< (m_i - 1)\cdot(1 - \text{Prob}\{m_i = 0\} - \text{Prob}\{m_i = 1\}) \\ &= (m_i - 1)\cdot\left(1-\left(1-\frac{l}{v}\right)^m - \frac{ml}{v}\left(1-\frac{l}{v}\right)^{m-1}\right). \end{aligned}$$

Since $\left(1-\frac{l}{v}\right)^m > 1 - \frac{ml}{v}$, we have

$$E\left(|C_i|\right) < \frac{m_i(m-1)^2l^2}{v^2} < \frac{m_i m^2 l^2}{v^2}.$$

$|C|$ is the sum of all $|C_i|$s, $0 \le i < \frac{v}{l}$. We know $\sum_{i=1}^{v/l} m_i = m$. So,

$$E\left(|C|\right) = \sum_{i=1}^{v/l} E(|C_i|) < m\frac{m^2l^2}{v^2} = \frac{m^3l^2}{v^2}.$$

If we let $v = \sqrt{m^3l^2}, E(|C|) < 1$.

Consider an arbitrary tag not in M. Without loss of generality, suppose it is mapped to the ith segment. In any partition of the segment, the probability for it to share a representative bit with a tag in M is $1 - \left(1-\frac{k}{l}\right)^{m_i}$. The probability for that to happen in all partitions is $\left[1-\left(1-\frac{k}{l}\right)^{m_i}\right]^k$. Hence, the probability for the tag to cause false positive, denoted as p_f is

$$\begin{aligned} p_f &= \sum_{q=0}^{m}\text{Prob}\{m_i = q\}\left[1-\left(1-\frac{k}{l}\right)^q\right]^k \\ &< (1-\text{Prob}\{m_i = 0\})\left[1-\left(1-\frac{k}{l}\right)^m\right]^k \\ &\approx (1-e^{-lm/v})(1-e^{-km/l}). \end{aligned}$$

The expected valus of $|F|$ is

$$\begin{aligned} E\left(|F|\right) &= (n-m)\cdot p_f \\ &< (n-m)(1-e^{-lm/v})(1-e^{-km/l}). \end{aligned} \tag{14}$$

If we let $v = -\frac{ml}{\ln\left(1-\frac{1}{(n-m)(1-e^{-km/l})}\right)}$ and apply it to (14), we have $E(|F|) < 1$. Now, if we choose $v = \max\{\sqrt{m^3l^2}, -\frac{ml}{\ln\left(1-\frac{1}{(n-m)(1-e^{-km/l})}\right)}\}$, the expected energy cost $E(\Omega) \le E(|C|) + E(|F|) + O(1) < 1 + 1 + O(1) = O(1)$. Therefore, ETOP can also be configured such that the energy cost per tag is $O(1)$.

B. Execution Time

Following the same analysis as in Section VI-B, it is easy to see that ETOP has the same formula for protocol execution time as TOP does: $T = \left(\frac{v}{l} + |F| + |C|\right)t_{\text{tag}} + m \times t_{\text{inf}} + O(1)$, but the values of $|C|$ and $|F|$ are different. Our simulation results in Section X show that ETOP has smaller execution time than TOP.

C. Choosing v for Time-Constrained Energy Minimization

Following the same reasoning in Section VI-C, we define the time bound for ETOP to be

$$\text{Prob}\{T \le B\} \ge \alpha \tag{15}$$

where T is the execution time of ETOP, B is a pre-defined time bound, and α is a probability value, $0 < \alpha < 1$. The objective is to find the largest value v that minimizes the energy cost, subject to the constraint (15). In the following, we derive a computable formula for $\text{Prob}\{T \le B\}$, which can be found in (23) and (24). Based on the formula, we will be able to find the optimal value v.

Let m_i be the number of tags in M that are encoded in the ith segment, denoted as $V_i, 0 \le i < \frac{v}{l}$. Obviously, m_i follows a binomial distribution Bino $\left(m, \frac{l}{v}\right)$,

$$\text{Prob}\{m_i = x\} = \binom{m}{x}\left(\frac{l}{v}\right)^x\left(1-\frac{l}{v}\right)^{m-x}.$$

Let n_i be the number of tags not in M that are mapped to the ith segment, $0 \leq n_i \leq n-m$. Obviously, n_i follows the binomial distribution Bino $\left(n-m, \frac{l}{v}\right)$.

$$\text{Prob}\{n_i = z\} = \binom{n-m}{z} \left(\frac{l}{v}\right)^z \left(1-\frac{l}{v}\right)^{n-m-z}.$$

Let C_i be a subset of C, containing the tags that are removed from V_i; Let F_i be a subset of F, consisting the false positive tags that are mapped to V_i; Let S_i be the union of C_i and F_i, thus $|S_i| = |C_i| + |F_i|$, and,

$$\begin{aligned}&\text{Prob}\left\{|S_i| = s \middle| m_i = x, n_i = z\right\}\\ &\quad= \sum_{c=0}^{s} \text{Prob}\left\{|F_i| = s-c \middle| |C_i| = c, m_i = x, n_i = z\right\}\\ &\qquad \cdot \text{Prob}\left\{|C_i| = c \middle| m_i = x\right\}.\end{aligned} \tag{16}$$

Firstly, we show how to calculate $\text{Prob}\{|C_i| = c| m_i = x\}$. After encoding m_i tags in V_i, let d_{ij} be the number of ones in the jth partition, $1 \leq j \leq k$. As a tag in M has exactly 1 representative bit in each partition, $0 \leq d_{ij} \leq \min\left\{m_i, \frac{l}{k}\right\}$. The reader removes a tag to C_i only if it shares a representative bit with another tag in the partition that contains the largest number of ones. As a result, $|C_i| = m_i - \max_{j\in[1,k]} d_{ij}$. When $y \geq 1$, we have

$$\begin{aligned}&\text{Prob}\left\{\max_{j\in[1,k]} d_{ij} = y \middle| m_i = x\right\}\\ &= \prod_{j=1}^{k} \text{Prob}\left\{d_{ij} \leq y \middle| m_i = x\right\}\\ &\quad - \prod_{j=1}^{k} \text{Prob}\left\{d_{ij} \leq y-1 \middle| m_i = x\right\}\\ &= \left(\sum_{d=0}^{y} \text{Prob}\left\{d_{ij} = d \middle| m_i = x\right\}\right)^k\\ &\quad - \left(\sum_{d=0}^{y-1} \text{Prob}\left\{d_{ij} = d \middle| m_i = x\right\}\right)^k\\ &= \left(\sum_{d=0}^{y} p_d\left(x, \frac{l}{k}, d\right)\right)^k - \left(\sum_{d=0}^{y-1} p_d\left(x, \frac{l}{k}, d\right)\right)^k\end{aligned} \tag{17}$$

where $p_d\left(x, \frac{l}{k}, d\right) = \text{Prob}\left\{d_{ij} = d \middle| m_i = x\right\}$ is the conditional probability that a partition containing $m_i = x$ tags happens to have d ones. The calculation of $p_d(\cdot)$ can be found in Section VI-D. Hence, the conditional distribution of $|C_i|$ is

$$\begin{aligned}&\text{Prob}\left\{|C_i| = c \middle| m_i = x\right\}\\ &= \left(\sum_{d=0}^{x-c} p_d\left(x, \frac{l}{k}, d\right)\right)^k - \left(\sum_{d=0}^{x-c-1} p_d\left(x, \frac{l}{k}, d\right)\right)^k.\end{aligned} \tag{18}$$

Secondly, we derive

$$\text{Prob}\left\{|F_i| = s-c \middle| |C_i| = c, m_i = x, n_i = z\right\}.$$

A tag not in M maps itself to k partitions and choose one bit randomly from each partition. If all these bits are ones, false positive happens. The conditional false positive probability is,

$$\begin{aligned}&\text{Prob}\left\{\text{false positive in } V_i \middle| m_i = x\right\}\\ &\qquad= \left(\sum_{d=0}^{x} \frac{kd}{l} p_d\left(x, \frac{l}{k}, d\right)\right)^k.\end{aligned} \tag{19}$$

When $|C_i| = c, \max_{j\in[1,k]} d_{ij} = m_i - c$, hence,

$$\begin{aligned}&\text{Prob}\left\{\text{false positive in } V_i \middle| |C_i| = c, m_i = x\right\}\\ &= \frac{1}{p_d\left(x, \frac{l}{k}, x-c\right)} \left[\left(\sum_{d=0}^{x-c} \frac{kd}{l} p_d\left(x, \frac{l}{k}, d\right)\right)^k\right.\\ &\quad \left. - \left(\sum_{d=0}^{x-c-1} \frac{kd}{l} p_d\left(x, \frac{l}{k}, d\right)\right)^k\right],\end{aligned} \tag{20}$$

denoted as p_{fc}, which represents the false positive probability when $m_i = x$ tags are encoded in the ith segments and $|C_i| = c$ tags are moved to the collision set C. When n_i tags in $N - M$ are mapped to V_i, the conditional distribution of $|F_i|$ follows the binomial distribution Bino(n_i, p_{fc}), thus,

$$\begin{aligned}&\text{Prob}\left\{|F_i| = s-c \middle| |C_i| = c, m_i = x, n_i = z\right\}\\ &\qquad= \binom{z}{s-c} p_{fc}^{s-c}(1-p_{fc})^{z-s+c}.\end{aligned} \tag{21}$$

From (18) and (21), we can derive $\text{Prob}\{|S_i| = s| m_i = x, n_i = z\}$. Thus, the probability distribution of $|S_i|$ is

$$\begin{aligned}&\text{Prob}\{|S_i| = s\}\\ &= \sum_{x=0}^{m}\sum_{z=0}^{n-m}\sum_{c=0}^{s} \text{Prob}\left\{|S_i| = s \middle| m_i = x, n_i = z\right\}\\ &\quad \cdot \text{Prob}\{m_i = x\} \cdot \text{Prob}\{n_i = z\}.\\ &= \sum_{x=0}^{m}\sum_{z=0}^{n-m}\sum_{c=0}^{s} \left[\left(\sum_{d=0}^{x-c} p_d\left(x, \frac{l}{k}, d\right)\right)^k\right.\\ &\quad \left. - \left(\sum_{d=0}^{x-c-1} p_d\left(x, \frac{l}{k}, d\right)\right)^k\right]\\ &\quad \cdot \binom{z}{s-c} p_{fc}^{s-c}(1-p_{fc})^{z-s+c}\\ &\quad \cdot \binom{m}{x}\left(\frac{l}{v}\right)^x\left(1-\frac{l}{v}\right)^{m-x}\\ &\quad \cdot \binom{n-m}{z}\left(\frac{l}{v}\right)^z\left(1-\frac{l}{v}\right)^{n-m-z}.\end{aligned} \tag{22}$$

Let S be the union of C and F. We have $|S| = |C| + |F|$ and $|S| = \sum_{i=1}^{v/l} |S_i|$. As $S_1, S_2, \ldots, S_{v/l}$ are independent of each other, the probability distribution of $|S|$ is the convolution of $|S_i|$. Hence,

$$\text{Prob}\{|S| = s\} = \text{Prob}\{|S_1| = s\} * \ldots * \text{Prob}\left\{\left|S_{\frac{v}{l}}\right| = s\right\},$$

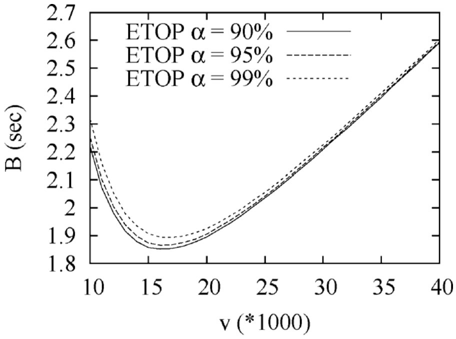


Fig. 5. Bound B that satisfies $\text{Prob}\{T \le B\} \ge \alpha$ with respect to v. Parameters: $n = 10,000, m = 1,000$.

where $*$ is the convolution operator. With the help of Fourier Transform, we have

$$\text{Prob}\{|S| = s\} = \hat{\text{FFT}}\left[\left(\text{FFT}\left(\text{Prob}\{|S_i| = s\}\right)\right)^{v/l}\right] \quad (23)$$

where FFT is the Fast Fourier Transform, and $\hat{\text{FFT}}$ is the inverse Fast Fourier Transform. Adopting (7), we have

$$\text{Prob}\{T \le B\} = \sum_{s=0}^{s_{\max}} \text{Prob}\{|S| = s\} \quad (24)$$

where $s_{\max} = \frac{B - mt_{\inf}}{t_{\text{tag}}} - \frac{v}{l}$. The right side is denoted as $P_t'(v, B)$, which is the probability for the protocol execution time to be bounded by B under a certain value of v. It is computable as a function of v and B after (22) is applied and parameters m and n are given. Given a value of B, we can find the largest v that satisfies $P_t'(v, B) < \alpha$ using the False Position algorithm [50]. For example, when $n = 10,000$, $m =$ 1,000, $B = 2$ seconds, and $\alpha = 99\%$, the optimal value of v is 23,200.

As a related problem, if v and α are given, we can use $P_t'(v, B)$ to compute the time bound that ETOP can achieve. More specifically, given a value of v, we are able to find the smallest B that satisfies $P_t'(v, B) \ge \alpha$ through bi-section search as described in Section VI-C. Fig. 5 shows the time bound of ETOP with respect to v when $\alpha = 90\%$, 95% and 99%, which correspond to the three curves in the figure.

IX. Implementation

A. Tag Requirement

We acknowledge that the proposed protocol cannot be directly supported by the off-the-shelf EPC C1G2 tags, which dominate today's market. EPC C1G2 is however for passive tags. This paper is about energy efficiency for active tags. Unfortunately, there is no widely accepted standard. There is a wide variety of active tags with different performance specifications and different price ranges. Below we list the requirements on the tags in order to implement the proposed protocols.

- Being able to receive 96 bit segments. This is for receiving system parameters and polling requests in the *ordering phase* and *polling phase*;
- Simple hash operations to determine the tag's reporting order;

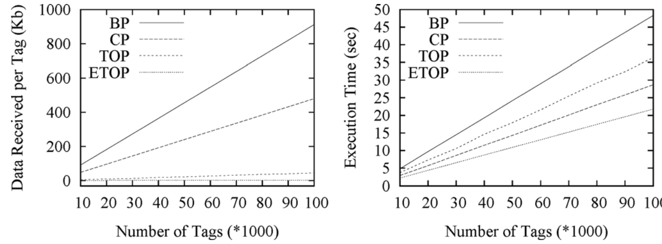


Fig. 6. Energy and time comparison. Parameters: $m = 0.1n$, $v = 24m$ for TOP and ETOP. Note that the horizontal '0' line is not at the bottom in order to make the ETOP curve visible.

- Transmitting sensor information in the *polling phase* and *reporting phase*.

B. Channel Error

Channel error may corrupt the data exchanged between the reader and tags. For example, if a negative polling request is corrupted, the tag that is not supposed to participate in the reporting phase will transmit and cause collision in the reporting phase. A segment of V sent from the reader may be corrupted so that tags encoded in this segment will not report their information. There exist other scenarios of corruption in the execution of TOP or ETOP. They cause two effects: 1) A tag in M does not transmit its information in the slot when it is supposed to transmit, and 2) it transmits but collides with another tag that is not supposed to transmit in the slot. To detect these cases, when a tag transmits, we require it to include a CRC checksum that is computed from the concatenation of the information bits and the tag's ID. When the reader expects information from a tag in a time slot, if the slot turns out to be empty or the data received in the slot do not carry a correct CRC, the reader knows that information from the tag is not correctly received. At the end of the protocol, all missed information can be retrieved by polling the tags directly.

X. Simulation Results

In this section, we evaluate the performance of our new protocols, the tag ordering polling protocol (TOP) and the enhanced tag ordering polling protocol (ETOP). We compare them with the basic polling protocol (BP) and the coded polling protocol (CP). Our evaluation uses two performance metrics: 1) the average number of bits that each tag has to receive during the protocol execution, and 2) the overall execution time.

We only consider energy consumption of tags in receiving information for two reasons. First, this is the major, variable portion of the energy cost per tag. As we will see shortly, each tag may have to receive hundreds of thousands of bits during protocol execution, whereas it only sends a small, fixed amount, e.g., 32 bits. Second, the energy cost for tags in M to transmit their information is the same for all protocols. Omitting them does not affect the comparison.

We use the following parameters to configure the simulation: each tag ID is 96 bits long, information reported from a tag to the reader is 32 bits long, and each segment in ETOP is 80 bits long and divided into 4 partitions, i.e., $k = 4$. Thirty-two bit information size should be sufficient in expressing the sensor data from a tag in many applications, such as battery status or environmental condition (e.g., humidity, temperature, or pressure).

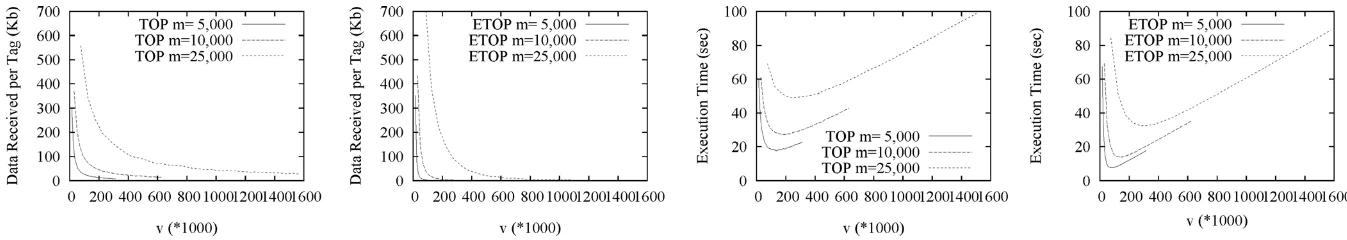


Fig. 7. First plot: Energy cost of TOP with respect to v. Second Plot: Energy cost of ETOP with respect to v. Third Plot: Execution time of TOP with respect to v. Fourth Plot: Execution time of ETOP with respect to v.

The transmission time is based on the parameters of the Philips I-Code specification [53]. The rate from a tag to the reader is 53 Kb/sec; it takes 18.88 μs for a tag to transmit one bit. Any two consecutive transmissions (from the reader to tags or vice versa) are separated by a waiting time of 302 μs. The value of $t_{\inf}$ is calculated as the sum of a waiting time and the time for transmitting the information, which is 18.88 μs multiplied by the length of the information. For 32 bit information, $t_{\inf} = 906$ μs. The transmission rate from the reader to tags is 26.5 Kb/sec; it takes 37.76 μs for the reader to transmit one bit. The value of $t_{\rm tag}$ is calculated as the sum of a waiting time and the time for transmitting a 96 bit ID. The result is 3927 μs.

A. Varying Number n of Tags

We first vary the number n of tags in the system from 10,000 to 100,000. We set $v = 24m$ and $m = 0.1n$, i.e., 10% of all tags are selected by the reader to report information. Fig. 6 compares four protocols in terms of energy cost and protocol execution time. The left plot shows energy costs. TOP and ETOP reduce energy consumption by one or multiple orders of magnitude. For example, when $n =$ 100,000, per-tag energy cost in TOP is 9.4% of the cost in CP, and 5.0% of the cost in BP. Per-tag energy cost in ETOP is just 0.52% of the cost in CP, and 0.28% of the cost in BP. The right plot shows the execution time comparison. TOP requires 25% less time than BP, but 27% more time than CP. ETOP requires 55% less time than BP and 24% less time than CP.

In summary, CP reduces both energy cost and execution time nearly by half when comparing with BP. TOP makes great improvement over CP in terms of energy cost, but has modestly higher execution time. ETOP considerably outperforms CP in terms of both energy cost and execution time.

B. Varying Size v of Reporting-Order Vector

Next, we show how the value of v influences the performance of TOP and ETOP. We set $n =$ 50,000 and $m =$ 5,000, 10,000, or 25,000. We vary v from $4m$ to $64m$ and use simulation to find energy cost per tag and protocol execution time. Fig. 7 shows the simulation results. The first two plots present the average amount of data each tag receives in TOP and ETOP, respectively. The curves match the theoretical results we have given in Section VI. When v is reasonably large, e.g., $v \geq 7m$, ETOP consumes less energy than TOP. The third and fourth plots present the protocol execution time of TOP and ETOP, respectively. ETOP also requires less time than TOP when $v \geq 7m$.

XI. Related Work

Much existing work on RFID systems is to design anti-collision ID-collection protocols, which read IDs from all the tags in the system. They mainly fall into two categories. One is *ALOHA-based* [13], [16]–[18], [21], [22], [54], and the other is *Tree-cased* [14], [15], [19], [20]. The *ALOHA-based protocols* work as follows: The reader broadcasts a query request. With a certain probability, each tag chooses a time slot in the current frame to transmit its ID. If there is a collision, the tag will continue participating in the next frame. This process repeats until all tags are identified successfully.

The *tree-based protocols* organize all IDs in a tree of ID prefixes [15], [19], [20]. 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 follows 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 [14], [19], [55].

In recent years, RFID research has been greatly broadened along a number of directions: estimating the number of tags in a system through various statistical measurements that are functions of the number tags [2], [13], [26]–[29], [31], [32], detecting the missing-tag event probabilistically or pinpointing exactly which tags are missing [33]–[36], locating tags that are out of their original places [37], continuously collecting the IDs of tags in a dynamic system where tags are moving in and out [30], [39], using indoor RFID tags for localization or navigation purposes [7], [56]–[58], investigating the security issues presented to resource-scarce tags such as authentication and lightweight encryption [59]–[66], and searching a reader's interrogation zone to find the subset of wanted tags [67].

Most related is the work by Chen *et al.* to collect sensor-produced data from all tags in a sensor-augmented RFID system [11]. Their protocols do not need to inform specific tags to report information. Instead, their focus is on minimizing the total time for all tags to transmit their data successfully to the reader. Moreover, energy consumption is not a concern in protocol design. Without any mechanism to separate the operations of a subset from others, their protocols cannot be applied to solve our problem. A follow-up work by Yue *et al.* complements [11] with an additional protocol to allow the collection of sensor information from RFID tags in multi-reader scenarios [68]. However, it makes a strong assumption that the tags are synchronized at the bit level.

XII. Conclusion

In this paper, we propose two energy-efficient polling protocols, TOP and ETOP, for large-scale RFID systems. These protocols are designed to collect real-time information from a subset of tags in the system. Our primary objective is to lower energy consumption by tags in order to extend their lifetime. The new protocols can be configured to achieve $O(1)$ energy cost per tag. Performance tradeoff between energy cost and execution time can be made by controlling the size of the reporting-order vector. Simulation results show that the new protocols are able to cut energy cost by more than an order of magnitude, when comparing with other protocols.

References

[1] H. Shen, Z. Li, L. Yu, and C. Qiu, "Efficient data collection for large-scale mobile monitoring applications," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 6, pp. 1424–1436, 2013.

[2] M. Kodialam, T. Nandagopal, and W. Lau, "Anonymous tracking using RFID tags," in *Proc. IEEE INFOCOM*, 2007.

[3] L. Yang, J. Cao, W. Zhu, and S. Tang, "A hybrid method for achieving high accuracy and efficiency in object tracking using passive RFID," in *Proc. IEEE Int. Conf. Pervasive Computing and Communications (PerCom)*, 2012, pp. 109–115.

[4] J. Han *et al.*, "Twins: Device-free object tracking using passive tags," in *Proc. IEEE INFOCOM*, 2014.

[5] R. Angeles, "RFID technologies: Supply-chain applications and implementation issues," *Inf. Syst. Manag.*, vol. 22, no. 1, pp. 51–65, 2005.

[6] J. Liu, B. Xiao, K. Bu, and L. Chen, "Efficient distributed query processing in large RFID-enabled supply chains," in *Proc. IEEE INFOCOM*, 2014, pp. 163–171.

[7] L. M. Ni, Y. Liu, Y. C. Lau, and A. Patil, "LANDMARC: Indoor location sensing using active RFID," *ACM Wireless Networks (WINET)*, vol. 10, no. 6, Nov. 2004.

[8] A. Ruhanen *et al.*, "Sensor-enabled RFID Tag handbook," in *BRIDGE, IST-2005-033546*, Jan. 2008.

[9] M. Miura, S. Ito, R. Takatsuka, T. Sugihara, and S. Kunifuji, "An empirical study of an RFID mat sensor system in a group home," *J. Networks*, vol. 4, no. 2, Apr. 2009.

[10] Y. Zhang, L. T. Yang, and J. Chen*, RFID and Sensor Networks: Architectures, Protocols, Security, and Integrations*. Boca Raton, FL, USA: CRC Press, 2010.

[11] S. Chen, M. Zhang, and B. Xiao, "Efficient information collection protocols for sensor-augmented RFID networks," in *Proc. IEEE INFOCOM*, 2011, pp. 3101–3109.

[12] Y. Qiao, S. Chen, and T. Li*, RFID as an Infrastructure*. New York, NY, USA: Springer, 2012.

[13] H. Vogt, "Efficient object identification with passive RFID tags," in *Proc. IEEE PerCom*, 2002.

[14] *Information Technology—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, Nov. 2003.

[15] 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. ISLPED*, 2004.

[16] S. Lee, S. Joo, and C. Lee, "An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification," in *Proc. IEEE MOBIQUITOUS*, 2005.

[17] B. Zhen, M. Kobayashi, and M. Shimizu, "Framed ALOHA for multiple RFID objects identification," *IEICE Trans. Commun.*, vol. 88, no. 3, pp. 991–999, 2005.

[18] J. R. Cha and J. H. Kim, "Dynamic framed slotted ALOHA algorithms using fast tag estimation method for RFID systems," in *Proc. IEEE CCNC*, 2006.

[19] J. Myung and W. Lee, "Adaptive splitting protocols for RFID tag collision arbitration," in *Proc. ACM MOBIHOC*, 2006.

[20] N. Bhandari, A. Sahoo, and S. Iyer, "Intelligent query tree (IQT) protocol to improve RFID tag read efficiency," in *Proc. IEEE ICIT*, 2006.

[21] V. Sarangan, M. R. Devarapalli, and S. Radhakrishnan, "A framework for fast RFID tag reading in static and mobile environments," *Int. J. Comput. Telecommun. Netw.*, vol. 52, no. 5, 2008.

[22] B. Sheng, Q. Li, and W. Mao, "Efficient continuous scanning in RFID systems," in *Proc. IEEE INFOCOM*, 2010.

[23] M. Chen, W. Luo, Z. Mo, S. Chen, and Y. Fang, "An efficient tag search protocol in large-scale RFID systems," in *Proc. IEEE INFOCOM*, 2013, pp. 899–907.

[24] L. Kong, L. He, Y. Gu, M. Wu, and T. He, "A parallel identification protocol for RFID systems," in *Proc. IEEE INFOCOM*, 2014.

[25] T. Li, S. Wu, S. Chen, and M. Yang, "Energy efficient algorithms for the RFID estimation problem," in *Proc. IEEE INFOCOM*, 2010.

[26] J. Zhai and G. N. Wang, "An anti-collision algorithm using two-functioned estimation for RFID tags," in *Proc. ICCSA*, 2005.

[27] M. Kodialam and T. Nandagopal, "Fast and reliable estimation schemes in RFID systems," in *Proc. ACM MOBICOM*, 2006.

[28] H. Han, B. Sheng, C. C. Tan, Q. Li, W. Mao, and S. Lu, "Counting RFID tags efficiently and anonymously," in *Proc. IEEE INFOCOM*, 2010.

[29] C. Qian, H. Ngan, Y. Liu, and L. Ni, "Cardinality estimation for large-scale RFID systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 9, pp. 1441–1454, 2011.

[30] Y. Zheng, M. Li, and C. Qian, "PET: Probabilistic estimating tree for large-scale RFID estimation," in *Proc. Int. Conf. Distributed Computing Systems (ICDCS)*, 2011, pp. 37–46.

[31] Y. Zheng and M. Li, "ZOE: Fast cardinality estimation for large-scale RFID systems," in *Proc. IEEE INFOCOM*, 2013.

[32] W. Gong, K. Liu, X. Miao, and H. Liu, "Arbitrarily accurate approximation scheme for large-scale RFID cardinality estimation," in *Proc. IEEE INFOCOM*, 2014.

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

[34] T. Li, S. Chen, and Y. Ling, "Identifying the missing tags in a large RFID system," in *Proc. ACM MobiHoc*, 2010.

[35] W. Luo, S. Chen, T. Li, and S. Chen, "Efficient missing tag detection in RFID systems," in *Proc. IEEE INFOCOM*, 2011, pp. 356–360.

[36] W. Luo, S. Chen, Y. Qiao, and T. Li, "Missing-tag detection and energy-time tradeoff in large-scale RFID systems with unreliable channels," *IEEE/ACM Trans. Netw.*, vol. 22, no. 4, pp. 1079–1091, 2014.

[37] K. Bu, B. Xiao, Q. Xiao, and S. Chen, "Efficient misplaced-tag pinpointing in large RFID systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 11, pp. 2094–2106, 2012.

[38] Q. Xiao, B. Xiao, and S. Chen, "Differential estimation in dynamic RFID systems," in *Proc. IEEE INFOCOM*, 2013, pp. 295–299.

[39] H. Liu, W. Gong, X. Miao, K. Liu, and W. He, "Towards adaptive continuous scanning in large-scale RFID systems," in *Proc. IEEE INFOCOM*, 2014.

[40] T. Li, S. Wu, S. Chen, and M. Yang, "Generalized energy-efficient algorithms for the RFID estimation problem," *IEEE/ACM Trans. Netw.*, vol. 20, no. 6, pp. 1978–1990, 2012.

[41] V. Namboodiri and L. Gao, "Energy-aware tag anti-collision protocols for RFID systems," in *Proc. IEEE PerCom*, 2007.

[42] D. Klair, K. Chin, and R. Raad, "On the energy consumption of pure and slotted aloha based RFID anti-collision protocols," *Comput. Commun.*, 2008.

[43] A. Hande, R. Bridgelall, and B. Zoghi, "Vibration energy harvesting for disaster asset monitoring using active RFID tags," *Proc. IEEE*, vol. 98, no. 9, pp. 1620–1628, Sep. 2010.

[44] M. Gorlatova, P. Kinget, I. Kymissis, D. Rubenstein, X. Wang, and G. Zussman, "Challenge: Ultra-low-power energy-harvesting active networked tags (EnHANTs)," in *Proc. ACM Mobicom*, 2009, pp. 253–260.

[45] "A batteryless programmable RFID-scale sensor device," The UMich Moo. [Online]. Available: https://spqr.eecs.umich.edu/moo/

[46] S. Chen, J. Li, and M. Zhang, "Polling Protocols in RFID Systems," Network Information Center, Univ. Shanghai for Science & Technology, Shanghai, China, Internal Technical Report, 2010.

[47] EPC Global, "EPC radio-frequency identity protocols class-1 generation-2 UHF RFID protocol for communications at 860 MHz–960 MHz version 1.2.0," 2008 [Online]. Available: http://www.gs1.org/gsmp/kc/epcglobal/uhfc1g2/uhfc1g2_1_2_0-standard-20080511.pdf

[48] X. Xu, L. Gu, J. Wang, and G. Xing, "Negotiate power and performance in the reality of RFID systems," in *Proc. Int. Conf. Pervasive Computing and Communications (PerCom)*, 2010, pp. 88–97.

[49] W. Chen *et al.*, "A two-stage wake-up circuit for semi-passive RFID tag," in *Proc. IEEE Int. Conf. ASIC*, 2009.

[50] R. Burden and J. Faires, *Numerical Analysis*. Boston, MA, USA: Brooks/Cole, Cengage Learning, 2011.

[51] B. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Commun. ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[52] A. Broder and M. Mitzenmacher, "Network applications of Bloom filters: A survey," *Internet Math.*, vol. 1, no. 4, pp. 485–509, 2004.

[53] Philips Semiconductors, I-CODE Smart Label RFID Tags, Jan. 2004 [Online]. Available: http://www.nxp.com/documents/data_sheet/SL092030.pdf

[54] Y. Qiao, S. Chen, T. Li, and S. Chen, "Energy-efficient polling protocols in RFID systems," in *Proc. ACM MobiHoc*, 2011.

[55] J. I. Capetenakis, "Tree algorithms for packet broadcast channels," *IEEE Trans. Inf. Theory*, vol. 25, no. 5, 1979.

[56] D. Zhang, J. Zhou, M. Guo, J. Cao, and T. Li, "TASA: Tag-free activity sensing using RFID tag arrays," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 4, pp. 558–570, 2011.

[57] K. Bu, X. Liu, J. Li, and B. Xiao, "Less is more: Efficient RFID-based 3D localization," in *Proc. IEEE MASS*, 2013, pp. 86–94.

[58] Y. Guo, L. Yang, B. Li, T. Liu, and Y. Liu, "Rollcaller: User-friendly indoor navigation system using human-item spatial relation," in *Proc. IEEE INFOCOM*, 2014.

[59] T. Dimitriou, "A secure and efficient RFID protocol that could make big brother (partially) obsolete," in *Proc. IEEE PerCom*, 2006.

[60] L. Yang, J. Han, Y. Qi, and Y. Liu, "Identification-free batch authentication for RFID tags," in *Proc. IEEE ICNP*, 2010.

[61] W. Gong, K. Liu, X. Miao, Q. Ma, Z. Yang, and Y. Liu, "Informative counting: Fine-grained batch authentication for large-scale RFID systems," in *Proc. ACM MobiHoc*, 2013, pp. 21–30.

[62] K. Sakai, M. Sun, W. Ku, and T. Lai, "Randomized skip lists-based private authentication for large-scale RFID systems," in *Proc. ACM MobiHoc*, 2013, pp. 277–280.

[63] M. Chen, S. Chen, and Q. Xiao, "Pandaka: A lightweight cipher for RFID systems," in *Proc. IEEE INFOCOM*, 2014.

[64] K. Bu, X. Liu, and B. Xiao, "Approaching the time lower bound on cloned-tag identification for large RFID systems," *Ad Hoc Networks*, vol. 13, pp. 271–281, 2014.

[65] L. Lu, Y. Liu, and X. Li, "Refresh: Weak privacy model for RFID systems," in *Proc. IEEE INFOCOM*, 2010.

[66] T. Li, W. Luo, Z. Mo, and S. Chen, "Privacy-preserving RFID authentication based on cryptographical encoding," in *Proc. IEEE INFOCOM*, 2012, pp. 2174–2182.

[67] Y. Zheng and M. Li, "Fast tag searching protocol for large-scale RFID systems," in *Proc. IEEE Int. Conf. Network Protocols (ICNP)*, 2011, pp. 363–372.

[68] H. Yue, C. Zhang, M. Pan, Y. Fang, and S. Chen, "A time-efficient information collection protocol for large-scale RFID systems," in *IEEE Proc. INFOCOM*, 2012, pp. 2158–2166.

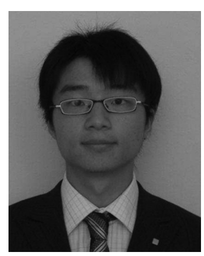
Yan Qiao received the B.S. degree in computer science and technology from Shanghai Jiao Tong University, China, in 2009, and the Ph.D. degree from the University of Florida, Gainesville, FL, USA, in 2014. Her advisor was Dr. Shigang Chen.

Her research interests include network measurement, algorithms and RFID protocols. She is now with Google Inc., Mountain View, CA, USA.

Shigang Chen received the B.S. degree in computer science from the University of Science and Technology of China, Shanghai, China, in 1993. He received the M.S. and Ph.D. degrees in computer science from University of Illinois at Urbana-Champaign, IL, USA, in 1996 and 1999, respectively.

After graduation, he had worked with Cisco Systems for three years before joining the University of Florida in 2002. He served on the technical advisory board for Protego Networks in 2002–2003. His research interests include computer networks, Internet security, wireless communications, and distributed computing. He is now an Associate Professor with the Department of Computer and Information Science and Engineering at the University of Florida, Gainesville, FL, USA. He has published more than 100 peer-reviewed papers, with about 5,000 citations. He holds 11 U.S. patents.

Dr. Chen received IEEE Communications Society Best Tutorial Paper Award in 1999 and NSF CAREER Award in 2007. He is an associate editor for IEEE TRANSACTIONS ON NETWORKING, *Elsevier Journal of Computer Networks* and IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. He also served as guest editor for the *Journal of Advances in Multimedia* in 2007, for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY in 2005, and for ACM/Baltzer *Journal of Wireless Networks (WINET)* in 2005. He has been serving on the steering committee of IEEE IWQoS since 2010. He was a TPC co-chair for IWQoS 2009, a TPC co-chair for the Computer and Network Security Symposium of ACM IWCMC 2009, and an area co-chair for the Network and Data Communications Track of the 10th International Symposium on Pervasive Systems, Algorithms and Networks (I-SPAN) 2009. He served as a TPC co-chair for the Computer and Network Security Symposium of IEEE IWCCC 2006, a vice TPC chair for IEEE MASS 2005, a vice general chair for QShine 2005, and a TPC co-chair for QShine 2004.



Tao Li received the B.S. degree in computer science from the University of Science and Technology of China, Shanghai, China, in 2007, and the Ph.D. degree from the University of Florida, Gainesville, FL, USA, in 2012. His advisor was Dr. Shigang Chen.

His research interests include network traffic measurement and RFID technologies.

Shiping Chen received the B.S. degree in electrical engineering from JiangXi University of China in 1984. He received the M.S. and Ph.D. degrees in computer science from the Institute of Computing Technology of the Chinese Academy of Sciences and Fudan University in 1990 and 2006, respectively.

He joined the University of Shanghai for Science and Technology, Shanghai, China, in 1990 and is currently a full Professor in the School of Optical-Electrical and Computer Engineering. He is also the Director of the network center of the university. His research interests include peer-to-peer networks, network communications, and database systems.