Energy-Aware Tag Anti-Collision Protocols for RFID Systems*

Vinod Namboodiri†, *Member, IEEE*, Lixin Gao‡, *Senior Member, IEEE*

***Abstract*— Energy consumption of portable RFID readers is becoming an important issue as applications of RFID systems pervade many aspects of our lives. Surprisingly, however, these systems are not energy-aware with the focus till date being on reducing the time to read all tags by the reader. In this work, we consider the problem of tag arbitration in RFID systems with the aim of designing energy aware anti-collision protocols. We explore the effectiveness of using multiple time slots per node of a binary search tree through three anti-collision protocols. We further develop an analytical framework to predict the performance of our protocols and enable protocol parameter selection. We demonstrate that all three protocols provide significant energy savings both at the reader and tags (if they are active tags) compared to the existing Query Tree protocol, while sharing the deterministic property of the latter. Further, we show that our protocols provide similar benefits even with correlated tag IDs.**

***Index Terms*— RFID MAC, Energy Aware Anti-Collision, Binary Tree Protocol**

I. Introduction

Radio Frequency Identification (RFID) is a technology by which radio frequency (RF) communication is used to store and retrieve data through a RF compatible integrated circuit. The main components of an RFID system are: a **reader**, including an antenna, which is the device used to read and/or write data to RFID tags; and a **tag** which is the device with an integrated circuit on which the reader acts. The tag can derive its energy for operation and transmission either from the reader's signal (passive tag) or through its own battery supply (active tag) [1]. RFID technology has found applications in inventory tracking, data handling, object identification and more. By writing data to tags and reading and/or modifying them later as required, RFID presents a dynamic front compared to the static bar code system.

For commercial systems located at industrial premises, readers are typically mounted on static locations and connected to wall socket power supplies. Tag reading throughput is important when scanning tagged items on a large scale or in continuous streams. There are, however, numerous other applications like inventorying items in supermarkets, garages, refrigerators, closets and more, where a user carries a portable reader that is battery-operated and scans tags on a much smaller scale. A mobile phone with reader functionality being used for the above applications is a typical instance of this scenario. Scanning of tags by portable readers results in faster depletion of battery energy supply, requiring more frequent recharging or replacement.There are two main requirements in such applications: tag reading time should not be perceptible to the user (or any delay is within acceptable limits), and the energy consumption of the reader should be minimized. Minimizing reader energy consumption would enable greater operating lifetimes, thus increasing its portability, and make RFID technology more pervasive. Energy consumption has not received serious attention in RFID systems previously. With more and more applications requiring mobile readers, however, energy has become an important resource to conserve, if and when possible. Energy awareness in these systems, resulting in greater operating lifetimes, can help accelerate the growth of this technology and its adoption for the numerous applications already envisaged.

There are many research issues in RFID systems both at the design and deployment stages. At the deployment stage the issue of tag detection is critical to ensure reliable operation of the system [1], [2]. On the other hand from a design perspective, compact antenna design is important to achieve desired reliability, performance, and low-cost production [3]. However, one of the most interesting research issues in these systems is arbitration while reading tags, also called the tag anti-collision problem. The reader sends an interrogation signal or query to collect the IDs of tags to identify the item and then possibly read the data stored on these tags. With many tags responding to the interrogation signal of the reader, it is important to be able to read the tag IDs of all. Several anti-collision protocols for RFID have been proposed in literature before, and can be broadly classified as based on Aloha and its variants or binary tree search [4]. Aloha based protocols rely on reducing collisions by separating tag responses temporally (or time slots within a frame if using frame-slotted Aloha). They are probabilistic in nature and simple [5], thus enabling simpler reader and tag implementations. However, they do not offer any guarantees on the time required to read all tags. In contrast, the binary tree search protocols are deterministic in nature [6], [7]. They are able to read all tags by successively querying nodes at different levels of the tree, with tag IDs distributed on the tree based on their prefix. The guarantee that all tags IDs will be read within a certain time frame has made the binary search protocol desirable in many applications. The binary tree search procedure, however, uses up a lot of reader queries and tag responses by relying on colliding responses of tags to determine which sub-tree to query next. This results in higher energy consumption at the readers and tags (if they are active tags).

We present three novel anti-collision protocols that combine the ideas of a binary tree search protocol with that of frame-slotted Aloha. These schemes are deterministic and energy-aware. The Multi-Slotted (MS) scheme relies on using multiple-slots per query to reduce the chances of collision among tag responses. The Multi-Slotted with Selective Sleep (MSS) scheme further explores the benefits of using sleep commands to put resolved tags to sleep during the arbitration process. These two protocols have a probabilistic flavor to them because tags choose a reply slot in a query frame randomly. The Multi-Slotted with Assigned Slots (MAS) scheme gives more structure to the tag responses by assigning tags in each

*A Preliminary version of this paper appeared in *Proceedings of the 5th Annual IEEE International Conference on Pervasive Computing and Communications* (IEEE PERCOM 2007), White Plains, NY, March 2007.

† *Affiliated to Wichita State University; can be contacted at vinod.namboodiri@wichita.edu*

‡ *Affiliated to University of Massachusetts Amherst; can be contacted at lgao@ecs.umass.edu*

[1]Semi-passive tags also exist that use battery supply for powering up but rely on the reader's signal for transmission

sub-tree of the search tree to a specific slot of the query frame. This makes the protocol fully deterministic, including the reply behavior of tags.

All three protocols are capable of adjusting the frame size used per query to maximize energy savings at the reader by reducing collisions among tag responses. The frame size can be chosen based on a specified average time constraint within which all tags IDs must be read. We develop an analytical framework for predicting the performance of these protocols in terms of the average time slots, average reader energy consumption and average tag energy consumption to read all tag IDs. We numerically evaluate our analytical expressions and demonstrate 20-60% reductions in energy consumption at the reader (depending on the scheme used) compared to an existing binary search protocol, the Query Tree protocol. Moreover, in spite of the focus on reducing reader energy consumption, energy savings at active tags is a useful byproduct of our schemes. We further compare these schemes for the case where tag ID distribution is unknown, possibly correlated. We show that even in this case, most of the benefits of our schemes are retained.

The rest of the paper is organized as follows: Section II provides some background on RFID anti-collision protocols with a brief description of the Query Tree Protocol and gives an overview of related work in the area. Section III presents our anti-collision protocols in detail. Section IV presents the energy models used at the reader and active tags, and the derivation of an analytical framework for the protocols. In Section V we numerically evaluate our schemes through the developed analytical framework and compare their performance and discuss how parameter selection for our protocols can be done under practical settings. In Section VI we extend our evaluations to the case where encountered tag IDs might come from an unknown distribution, and are possibly correlated. Concluding remarks are made in Section VII.

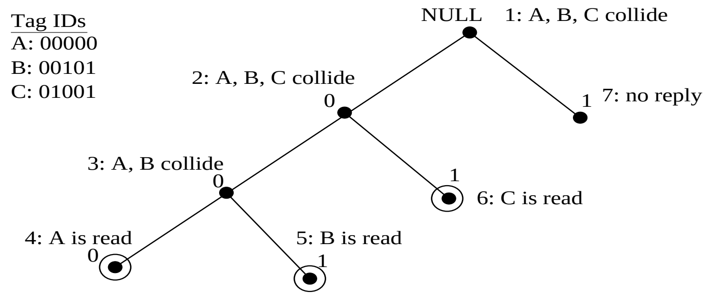


Fig. 1. Example of Query Tree Protocol. The complete IDs of tags A, B, and C belong at the bottom of the tree. However, these tags are uniquely identified in the initial few levels based on their prefixes as shown by circled nodes on the tree.

No.	Reader Query	A (00000)	B (00101)	C (01001)	Result
1	NULL	R	R	R	collision
2	0	R	R	R	collision
3	00	R	R	NR	collision
4	000	R	NR	NR	A read
5	001	NR	R	NR	B read
6	01	NR	NR	R	C read
7	1	NR	NR	NR	no tags read

TABLE I
STEPS IN QUERY TREE PROTOCOL WITH THREE TAGS WITH 5 BIT IDS; NULL REPRESENTS AN EMPTY STRING QUERY TO WHICH ALL TAGS RESPOND, R DENOTES A RESPONSE AND NR DENOTES NO RESPONSE TO A QUERY.

II. Background and Related Work

In this section we provide a more detailed account of existing RFID anti-collision protocols. We also describe related work and how our contribution differs from existing anti-collision protocols.

A. Background

RFID anti-collision protocols can be categorized into Aloha based protocols and tree based protocols [4]. Aloha based protocols reduce the probability of tag collisions by spreading out tag responses over time[5]. In the basic Aloha protocol, tags can respond at any time [8], while in slotted Aloha, tags respond only at the beginning of time-slots [9], [10]. The frame slotted Aloha protocol further groups time slots into frames within which tags must respond [11], [12]. The frame-slotted protocol shows the best performance for RFID systems among Aloha based protocols due to a tag responding in a time slot with its ID only once every frame [4]. Aloha based protocols, however, are not deterministic and thus do not have analytically provable upper bounds on the time to read all tags. The simplicity of Aloha based protocols, along with fixed size responses (which helps assign time slot durations efficiently) from tags for reader queries, has led to their adoption in some early RFID based systems and associated standards (e.g. [13]). This has resulted in a fair share of work done with these protocols in the research community (e.g. [14], [15], [16]). For many RFID based systems, however, it is important that an anti-collision protocol is deterministic, *ie.* there be a guarantee that all tags will be read within certain time limits. Tags are within range of readers for only a limited amount of time and any unread tag could be costly in the context of the application.

On the other hand, tree based anti-collision protocols provide guarantees on the time required to read all tags in range, and are thus deterministic. Algorithms for arbitration using trees have been around for a while [5]. The authors of [17] were one of the first to consider these for reading tags in RFID systems. They analyze a B-ary tree search algorithm extensively, producing analytical results for average time slots, reader messages, and tag responses required to read tags. The two most widely used tree based search protocols are the Binary Tree (BT) protocol [6] and the Query Tree (QT) protocol [7], and are based on classical collision resolution algorithms [18], [19]. Both schemes work by splitting tag IDs bitwise using queries from the reader until all tags are read. In the BT scheme, after a reader query, the reader announces the outcome: collision, identification or no-answer, based on which tags modify their behavior for future queries until all tags are read. The BT scheme relies on tags remembering results of previous inquiries by the reader which makes the tags susceptible to their power supply. Further, QT has been shown to outperform BT in previous work [20]. Thus our main focus will be on the QT protocol only and we describe it briefly below. We will use the QT protocol as a reference to judge how good our proposed protocols are in the later sections of this work.

The QT protocol is a memoryless anti-collision protocol with tags requiring no additional memory other than that required to store their ID [7]. The algorithm consists of rounds of queries and responses. In each round, a reader sends a query as a prefix. Tags with this prefix respond back with the remaining bits of their ID. When more than one tag responds to a prefix query, a collision results, and the reader realizes that there are at least two tags with the same prefix. The reader then extends the prefix by a '0' or '1' bit and continues the query with this longer prefix. A tag is resolved or uniquely identified when a query has only one tag responding. A query whose prefix matches that of the resolved tag ID is never sent again because no further tags with that prefix remain unread. A working example of this protocol is shown in Figure 1 where a binary tree with three tags A, B, and C. Table I shows the steps of the arbitration process in more detail. Each tag logically belongs at the bottom level of the tree based on their complete IDs, but can be uniquely identified at higher levels. The reader queries the tree in depth-first fashion to read all tags, requiring 7 reader messages in all corresponding to each query. Total tag responses sent are 11. The corresponding activity of each node in the tree is shown to the immediate left or right. Circled nodes on the tree show when the highest level at which each tag can be uniquely identified.

B. Related Work

Many researchers have explored further extensions to the basic QT protocol as presented above. The authors of [7] state an optimization (called Short-Long Queries) where tags respond to reader queries with only a subset of their ID. Tags are made to respond with their complete ID only when the reader is sure that only one tag matches the prefix. The authors of [21] look at reducing the number of collisions of tag responses by modifying the QT protocol to adaptively adjust the prefix it queries taking into account tag IDs that have already been read before, but still present in the reader's interrogation zone. There have also been studies on how the QT protocol can be improved to handle tag IDs which could have some common prefixes [22], [20]. All these work can be used in conjunction with our protocols in a similar fashion. Our main focus in this work is to make the QT protocol more energy-aware. To the best of our knowledge, the only pieces of work that focus on energy with RFID anti-collision protocols are [23], [24]. The work in [23] compares existing variants of pure and frame-slotted Aloha protocol in terms of energy efficiency for their suitability for use in RFID-enhanced wireless sensor networks (WSNs). This is orthogonal to our work which considers the QT protocol (a B-ary tree protocol) and how it can be made energy-aware from a design perspective. The authors of [24] presented ways to minimize power consumption of passive tags by allowing the reader to detect a collision early and signal the tags to stop sending more bits of their ID. This differs from the QT protocol in that when the tags are sending their IDs, the reader, once detecting a collision bit, gives the tags a signal to stop sending. This results in reduced number of sending operations in the worst case. This optimization is complementary to our work and can be used to provide power savings at passive tags which is critical for their working range.

There are some similarities between our proposed protocols and prior work in the usage of multiple-slots. The work in [25] briefly mentions a multi-slotted scheme that uses less time slots than the typical (single slotted) B-ary tree search algorithm. The number of slots per query in this scheme depends on the value of B, while in our work the number of slots per query can be selected as required, independent of B. The EPC Class 1 Gen 1 standard specifies a ping command which is used to read tags by filtering their IDs for a sub-sequence of bits and collecting responses into 'bins' [26]. The work by authors of [27] present a binary tree based protocol where tags respond to a query by the reader with some probability that is decided based on the number of contending tags, and relies on a continuous estimation of tag count as the protocol proceeds. These latter two pieces of work have some similarity to our MAS scheme in their approach to gain structured responses to queries. Our work, by varying the number of slots per query, focuses on energy awareness

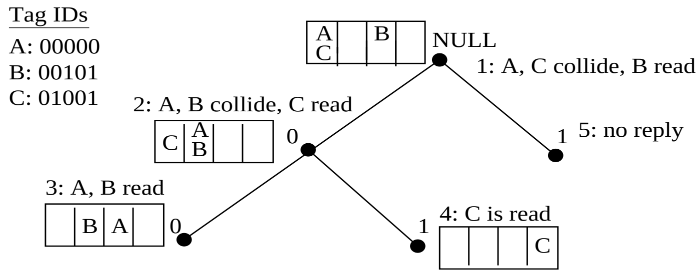


Fig. 2. Example of Multi-Slotted Scheme. Tags select one slot in the query frame randomly when the query matches their prefix.

in anti-collision protocols, whereas the goal of above mentioned work is to minimize tag identification delay. We study different variants of the multi-slotted approach to provide insight on what properties are desirable in the design of energy aware tree-based anti-collision protocols.

III. Energy-Aware Multi-Slotted Anti-Collision Protocols

Our approach to develop energy-aware anti-collision protocols for RFID systems is to reduce the number of tag response collisions. We do this by combining the contention resolution properties of the QT protocol and frame-slotted Aloha. The QT protocol relies on colliding responses to queries for determining the location of tag IDs. No effort is, however, made to use such queries (with colliding tag responses) to possibly resolve tag IDs directly. The idea of our protocols is to allow tags to transmit responses within a slotted time frame and thus, try to avoid collisions with responses from other tags. Thus, in our approach, each query is used to find colliding sub-trees as well as read tag IDs. This two-pronged approach allows more tags to be read with the same number of queries as before, reducing the number of reader queries to be sent by the reader, while at the same time requiring fewer tag responses in all. Using multiple-slots per query possibly increases the time required to read all tags. We investigate the energy consumption of our multi-slotted schemes in relation to the QT protocol in later sections. In this section we describe three different variants of our multi-slotted approach. These variants offer insight into the design choices to be made when contemplating how anti-collision protocols can be more energy-efficient or energy-aware. Our evaluations later in this paper compare these variants with the aim of determining what variant can be most useful to reduce energy consumption.

A. Multi-Slotted Scheme

The Multi-Slotted (MS) scheme works as follows. At each node of the B-ary tree[2], F slots are used to read tag responses. Tags randomly choose a slot to respond. If all tags with the prefix of the node are read successfully within the F slots without collisions, the sub-trees of that node are not queried further. If there is at least one collision in the responses, sub-trees from that node are queried as before and so on. Some tag IDs may be read (and some not, the ones that collided) in such a frame, but since the reader does not know to which subtrees the colliding tags belong to, it still has to query all the sub-trees. This is because the reader has no way of telling the tags that were read, to stop responding. These tags would thus still respond to further queries until their prefix is ignored by future queries.

[2]Even though we use a generalized B-ary tree for explanations, the examples used to demonstrate the schemes assume a binary tree (i.e. B = 2).

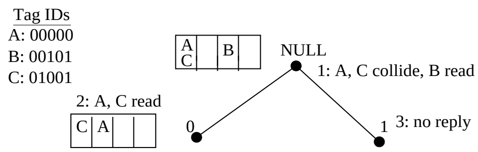


Fig. 3. Example of Multi-Slotted Scheme with Selective Sleep. Tags select one slot in a frame randomly. This protocol puts identified tags to sleep through a sleep command.

A binary tree with with tags A, B and C is shown in Figure 2. The corresponding query frame (this example uses frame size $F = 4$) and activity of each node in the tree is shown to the immediate left or right. The reader queries the tree in depth first fashion to read all tags within a frame whose responses have not collided with those of any other tag in a slot. For this example, the reader initially queries the root node with 'NULL' prefix to which all tags respond. It happens that tag B chooses a unique slot and is read while responses of tags A and C collide on choosing the same slot. At the colliding slot, the reader is unaware of tags that responded, and still has to query all sub-trees. In the next query with prefix '0', all three tags respond, and now another slot has a collision of responses from tags A and B. Note that even though tag B was identified in the first step, it wasn't put to sleep by the reader, and hence, still responds if its prefix is queried. The reader subsequently descends to the next level with a query prefix of '00' at which point only A and B match the prefix. As they chose unique slots, they are now identified. The reader also does not need to query any further sub-trees as it has read all tags successfully. Next, continuing the depth first querying pattern, a query with prefix '01' enables identifying tag C as well. Interestingly, an additional query with prefix '1' is required as the reader is uncertain if any tags exist in that sub-tree after the collision in the query frame at the root node. Total number of reader queries required are 5 while total tag responses sent is 9.

B. Multi-Slotted Scheme with Selective Sleep

The Multi-Slotted Scheme with Selective Sleep (MSS) explores an alternative approach by incorporating the use of selective sleep commands in the MS scheme. In this scheme, as tags are read at nodes of the tree in collision frames (frames with at least one collision), they are sent to sleep by the reader. Sleep commands are not used on tags read in a collision free frame because the subtrees of that prefix will not be queried further anyway. This approach of putting tags to sleep does not help reduce the number of reader queries directly because the reader still does not know what further prefix the colliding tags had. This, however, helps indirectly by reducing the number of contending tags. Reduction in tag responses result in possibly fewer collisions at nodes of the tree, requiring fewer queries by the reader. The communication and time costs of issuing sleep commands warrant attention. Our subsequent analysis along with numerical evaluations presented later will explore how worthwhile the usage of selective sleep commands are in reducing reader energy consumption.

The working example for the MSS scheme is shown in Figure 3. At the root node, all three tags respond to a query with 'NULL' prefix, and a collision occurs between the responses of tag A and C. Tag B is identified and put to sleep. For the next query with prefix '0', tags A and C choose unique slots and hence both are identified as well. As the reader is unaware of where the the colliding responses came from in the query at root node, it still has to send a query with prefix '1' to check for tags in that sub-tree. The number reader

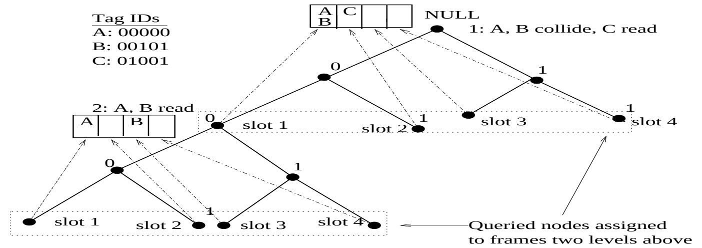


Fig. 4. Example of Multi-Slotted Scheme with Assigned Slots. Tags in level $L + log_B F$ are assigned to respond to individual slots in a frame at level L when queried as opposed to a random selection of a slot. The scheme can also be visualized as an F-ary tree with prefix incremented by $log_B F$ bits after each query.

queries required is 3 along with 1 sleep command for tag B, thus requiring 4 reader messages in all. Total tag responses sent are 5.

Term	Definition
Frame	A collection of time slots
Frame Size (F)	Number of time slots within a frame
B	The base of tag ID's corresponding to a B-ary tree.
Level (L)	The distance of a node in the binary tree from root node.
m	Number of tags in the field

TABLE II
KEY TERMS AND THEIR DEFINITIONS

C. Multi-Slotted Scheme with Assigned Slots

The previous two multi-slotted search schemes were designed to reduce the number of queries by the reader and responses by the tags. Some opportunities were lost, however, to reduce them even more by failing to utilize the tags that were read within the F slots. A single collision in F slots was enough to force the reader to probe all further sub-trees of the node as if all slots had collisions. This was because the reader had no idea about the prefix of tags whose responses had collided - tags randomly chose a slot to send their responses. Here we describe the Multi-Slotted Scheme with Assigned Slots (MAS) that uses a structured assignment of slots to nodes of the B-ary tree at different levels. This enables the reader to make use of the non-colliding responses as well to learn which sub-trees it need not probe further.

The reader chooses a frame size F as before, but now such that $F = B^d$ where d is a positive integer (note we require $F \geq B$ now; also $log_B F = d$). Each node u at level L of the tree (with F slots each) then allocates one slot to each node[3] v in its sub-tree at level $L + log_B F$. Thus, the tags whose prefix matches that of node v will respond to the reader's interrogation at node u by transmitting on the slot assigned to it in node u. Each tag knows the slot it must transmit on as follows. Based on the query prefix q at node u, tags will transmit on $qxx \cdots x$ slot, where x's are the $log_B F$ bits after prefix q in the tag IDs. Thus, all tags with prefix $qxx \cdots x$ will transmit with their slots determined by the value of $xx \cdots x$. Consider an example with $B = 2$, $F = 4$, and $q = 00$. That is the query is at a node u corresponding to 00 with $L = 2$ on the binary tree. u has four nodes (the v nodes of above description) in

[3]Remember that the term node refers to vertices on the binary tree and not tags which are distributed under different nodes based on their IDs. Also note that nodes at higher levels carry smaller sub-prefixes of tag IDs compared to nodes at lower levels and a tag matching the prefix of a node u will also match the prefix of any nodes v in u's sub-tree.)

its subtree at level $L + log_B F = 2 + 2 = 4$. Each of these nodes v will then have tags matching its prefix reply on the corresponding slot at node u. The number of x's will be $log_B F = 2$. Thus, each of the v nodes will correspond to prefixes $q00$, $q01$, $q10$ and $q11$. The tags with prefix $q00$ will transmit on slot 1 (in frame at u), tags with prefix $q01$ will transmit on slot 2 and so on.

In the protocol, starting at level $L = 0$, queries are sent only at levels $log_B F$, $2log_B F$, $3log_B F$ and so on. The scheme can be visualized as an F-ary tree with a query at each node responded to by tags from F sub-treesat a level $log_B F$ below the current level. That is, in such an F-ary tree, the prefix is increased by $log_B F$ bits after each query. A working example for the MAS scheme is shown in Figure 4. For the query at root node with 'NULL' prefix, the sub-tree at level 2 with both tags A and B is assigned slot 1 and hence their responses collide. Tag C is read as its subtree at level 2 is assigned as slot 3. In the subsequent query at level 2 with prefix '00', the subtrees of tags A and B at level 4 are assigned slots 1 and 3, and they are read. No additional queries are required due to the collision at root node (prefix 'NULL') as the reader knows no tags exist in the tree corresponding to the empty slots (3 and 4) in the query frame. Thus, 2 reader messages and 5 tag responses are required.

IV. Analysis of Multi-Slotted Schemes

Here we develop an analytical framework of the three multi-slotted schemes described in the previous section. This analytical framework will be used to predict the performance of the protocols as well as help in selecting an ideal frame size F as described in later sections. The analytical expressions developed will then be studied through numerical evaluation in the following section to ascertain the effectiveness of the protocols in terms of energy consumption (at both the reader and tags) and time to read all tags. We begin by presenting our energy model at the reader as well as a tag.

A. Energy Model

The energy consumption at the reader is a function of the number of queries it sends and the number of slots spent in the receive mode. Similarly, the energy consumption at an active tag is a function of the number of queries received by the tag and the number of responses it sends back. We will neglect the energy spent in modes other than transmit and receive in our models for simplicity. In our analysis we will assume that the time slot in which a reader query or message is sent is of the same duration as that of a tag response. The energy consumption model of the reader is based upon a half-duplex operation where the reader transmits energy and its query for a specific period and then waits in receive mode with no more energy transmission until the end of the frame. Such a pulse based, half-duplex operation is also termed as sequential (SEQ) operation[4] [4]. It is assumed that all tags (including passive tags) are sophisticated enough to count down slots immediately after the query is transmitted by the reader, and respond within the slotted time frame without assistance from the reader.

Let P_{Rtx} and P_{Rrx} represent the power required by the reader to transmit and receive respectively. Similarly, let P_{Ttx} and P_{Trx} represent the power required by an active tag to transmit and receive respectively. Since at a node of the binary tree, there is one slot for a query from the reader and F slots in which the reader awaits responses, the reader energy consumption can be expressed as $q(m)\,[P_{Rtx} + P_{Rrx}\,F]$ where $q(m)$ is the number of queries required to read all m tags. The total energy consumption of all active tags can be expressed as $q(m)\,P_{Trx} + u(m)\,P_{Ttx}$, $u(m)$ is the total number of tag responses. An exception to the above expression is for the MSS scheme where we have to account for the energy required at the reader to issue sleep commands (1 slot) and for the tags to receive these (1 slot). Thus for the MSS scheme the reader energy consumption is $q(m)\,[P_{Rtx} + P_{Rrx}\,F] + z(m)\,P_{Rtx}$ and total for all active tags is $q(m)\,P_{Trx} + u(m)\,P_{Ttx} + z(m)\,P_{Trx}$, where $z(m)$ is the number of sleep commands issued by the reader.

B. Analysis

Our focus will be on an average case analysis of energy consumption. Thus, we will determine expected values of $q(m)$ and $u(m)$, namely $\bar{q}(m)$ and $\bar{u}(m)$ for all three protocols (and $\bar{z}(m)$ for MSS scheme) and use those to find the average energy consumption at both the reader and active tags for each protocol. Note that we are finding the total average energy consumption at tag side due to the collective nature of the tag arbitration process[5]. We will also derive expressions for the average number of time slots required to read all tags, $\bar{t}(m)$.

There are m tags to be read in the field. These m tag IDs are uniformly distributed at the bottom level of the B-ary tree. Consequently, the probability of k of the m tags having the same prefix as the query of size L bits, is given by the binomial distribution

$$P(k, m, L) = \binom{m}{k} p^k (1-p)^{m-k}, \qquad L > 0 \tag{1}$$

with p being the probability that a tag's ID has the prefix of the queried node, and is a function of L.

Analysis overview for all three schemes

Based on the probability of a specific number of tags responding at a node in the tree, and given the frame size, we can calculate the probability of collision at that node. This can be extended to find the probability of visiting a node, as a node is visited if and only if a query at its its parent node suffers a collision. We then use the linearity property of the expectation of random variables to sum the probability of visiting a node over all nodes in the tree to obtain the expected number of nodes visited in the tree. As each visit of a node in the tree corresponds to a reader query, this provides us with the expected number of reader queries sent. We use the above strategy to also calculate the expected number of time slots required to read all tags. This is a simple extension as each reader query consists of $F + 1$ time slots. Finally, the expected total number of tag responses is found by summing up the expected number of tags responses after queries at each level of the tree.

*1) **MS Scheme Analysis**:* To analyze for the expected number of nodes visited in the tree with the MS scheme we begin by finding the probability of collision at a node. As explained above, due to the

[4]If the reader were required to emit the same transmission energy for tags to respond back (which is the case with passive tags when not using SEQ operation), then our strategy of using multiple slots would require more energy than the QT protocol unless the number of time slots required is fewer. Thus, the utility with passive tags will be much lesser and the benefits will mainly remain limited to active tags only. This indicates that when energy awareness is required, the mode of operation of RFID systems becomes important as well.

[5]The corresponding value per active tag can be found easily by dividing by the number of tags.

nature of the querying process, a node is visited if and only if there is a collision at the parent node.

We can obtain the probabilities for the following behavior at a node when the tree is probed: no reply or idle, I, at least one collision in frame, C and collision free replies in frame or success, S.
At level L of the B-ary tree, there are B^L nodes. The probability that an ID has the prefix of the queried node is $p = B^{-L}$ and the probability that it does not have the prefix of the query is $(1-B^{-L})$. Thus, using Equation 1 the probability of no reply is

$$P_I(m, L) = P(0, m, L) = (1 - B^{-L})^m, \qquad L > 0. \tag{2}$$

Let $P_{NC}(F, k)$ represent the probability that there are no collisions when k tags transmit within a frame of size F. The probability $P_{NC}(F, k)$ can be calculated as follows. From the k tags, the first tag can choose a slot to transmit without collisions with probability F/F given F possible time slots. The next tag can now choose $F-1$ of the remaining slots with probability $F-1/F$ to avoid collision and so on. Thus,

$$P_{NC}(F, k) = \frac{F}{F} \times \frac{F-1}{F} \times \cdots \times \frac{F-k+1}{F} = \frac{F!}{F^k (F-k)!}. \tag{3}$$

Collision free replies are received at a node in the tree if there is either only one tag in the sub-tree or all tags use different slots in the frame when responding. Then, the probability of no collisions in such a frame is,

$$\begin{aligned} P_S(m, L) &= \sum_{k=1}^{m} P(k, m, L)\ P_{NC}(F, k) \\ &= \sum_{k=1}^{min(m,F)} P(k, m, L)\ P_{NC}(F, k), \qquad L > 0 \end{aligned} \tag{4}$$

where the final equality is due to the fact that if more than F tags transmit in a frame of size F, the probability of no collision is zero.

Since the sum of probability of a collision free frame, probability of a frame with no responses, and probability of a frame with colliding responses equals one (as these are the only three possibilities, and are mutually disjoint),

$$\begin{aligned} P_C(m, L) &= 1 - P_S(m, L) - P_I(m, L) \\ &= 1 - (1 - B^{-L})^m \\ &\quad - \sum_{k=1}^{min(m,F)} P(k, m, L)\ P_{NC}(F, k), \qquad L > 0. \end{aligned} \tag{5}$$

Additionally, for $L = 0$, $P_C(m, L) = 1 - P_{NC}(F, m)$.

Now we analyze the average number of nodes of the tree the querying algorithm visits, by making use of the probabilities calculated above. This will provide a handle on the number of reader queries and time slots required to read all the tags.

At any level L, all nodes have an equal probability of being visited with our assumption of uniformly distributed tag IDs in the tag ID space. Moreover, as mentioned above, all nodes except the root node at level 0 are visited only if their parents are collision nodes. It could happen that some tags with the prefix of a query do not suffer any collision when responding. But since the tags respond randomly in any slot, there is no way of the reader knowing which subtree the colliding tags belong to. Thus, one or more collisions in the F slots at a node requires all child nodes to be queried.

Based on these facts, the probability of a node being visited at level L is

$$P_V(m, L) = \begin{cases} 1 & L = 0 \\ P_C(m, L-1) & L > 0 \end{cases}. \tag{6}$$

This comes about due to the fact that to visit a node, all the predecessor nodes in the tree have to be visited. For any tag to be read at a node, there has to be a collision at all previous nodes in the tree starting at the root. This implies that the probability of visiting all previous nodes above the parent of a node is unity, and the probability of visiting the node only depends on the probability of collision at its parent.

Based on the probability of a node being visited, we can calculate the expected number of nodes visited in the binary search tree by summing over all nodes at all levels L, with each level having B^L nodes. We assume an infinite tree where L can be as large as ∞ at this point and later introduce a correction for finite trees since the tag ID length is fixed and finite. Now the average number of slots required to read all tags then is the product of number of time slots per node (and 1 slot for reader probe) and the average number of nodes visited. Thus,

$$\begin{aligned} \bar{t}_{MS}(m) &= (F+1)[1 + \sum_{L=1}^{\infty} B^L P_V(m, L)] \\ &= (F+1)[1 + \sum_{L=1}^{\infty} B^L P_C(m, L-1)] \\ &= (F+1)[1 + \sum_{L=0}^{\infty} B^{L+1} P_C(m, L)]. \end{aligned} \tag{7}$$

Using Equation (5) gives

$$\begin{aligned} \bar{t}_{MS}(m) = (F+1)[1 &+ B(1 - P_{NC}(F, m)) \\ &+ B \sum_{L=1}^{\infty} B^L \{1 - (1 - B^{-L})^m \\ &- \sum_{k=1}^{min(m,F)} P(k, m, L)\ P_{NC}(F, k)\}]. \end{aligned} \tag{8}$$

Next, we determine the average number of queries by the reader. This is the same as analyzed above in Equation (8), without the multiplicative factor $F + 1$,

$$\begin{aligned} \bar{q}_{MS}(m) = 1 + B(1 - P_{NC}(F, m)) + B \sum_{L=1}^{\infty} B^L \{1 - (1 - B^{-L})^m \\ - \sum_{k=1}^{min(m,F)} P(k, m, L)\ P_{NC}(F, k)\}. \end{aligned} \tag{9}$$

Next, the average total number of tag responses is found by summing up the expected number of tag responses per level of the tree. The expected number of responses at a level is the difference between the total number of tags and expected number of tags resolved at all levels above this level.

Let $\bar{s}(m, n)$ be the average number of tags whose ID was read by the tree search algorithm for all levels less than and including n. Since tags read in a collision frame (a frame with at least one collision) stay active when queried, we will count these tags as resolved only when they are read in a collision free frame, which they will be eventually with the prefix ignored subsequently. This does not change the result and allows us to use the expression developed above for P_{NC} simplifying the analysis in this section.

At any node w at level n in the tree, the expected number of tags resolved or read is

$$\bar{r}(m, n) = \begin{cases} \sum_{k=1}^{min(m,F)} k\, P(k, m, n)\ P_{NC}(F, k) & n > 0 \\ m P_{NC}(F, m) & n = 0 \end{cases},$$

which corresponds to the tags read in collision free frames. Thus, summing over all nodes w at a level n

$$\bar{s}(m,n) = \sum_{w=0}^{B^n-1} \bar{r}(m,n) = B^n \bar{r}(m,n) = \begin{cases} B^n \sum_{k=1}^{min(m,F)} k\, P(k,m,n)\ P_{NC}(F,k) & n > 0 \\ mP_{NC}(F,m) & n = 0 \end{cases}. \tag{10}$$

By counting all the tags read at level n, we ensure we have counted all tags that will be read above level n in the tree as well. This is because, any tag expected to be read at levels above n, will be read at level n also.

At the root level, or $L = 0$, a query receives m responses. At all levels L below this, the number of responses is m minus the number of resolved tag IDs in levels 0 through $L-1$. Thus the average number of tag replies or responses can be quantified as

$$\bar{u}_{MS}(m) = m + \sum_{L=1}^{\infty}(m - \bar{s}(m, L-1)), \tag{11}$$

where $\bar{s}(m, L-1)$ is the number of tags resolved up to and including level $L-1$.

The above analysis uses an infinite sum for number of levels L. This sum can be limited to some value $L^{'}$ with the quality of approximation depending on the value of $L^{'}$. It is shown in [25] that the average number of nodes visited when reading all tags with a binary search tree can be computed within a tolerance ϵ, if

$$L^{'} \geq -log_B\left(\frac{\epsilon}{2m(m-1)}\right).$$

This result relies on the fact that the probability of collision at a node becomes very small after a certain number of levels under the assumption of uniformly distributed tag IDs. Thus, even for a large number of bits in tag IDs, we can get a good approximation using the limit of $L^{'}$ above. Our results in this work use $\epsilon = 0.01$. Note that using this approximation reduces computation required compared to using $L^{'}$ equal to number of bits in tag IDs.

Now, using Equation (10) with the upper limit $L^{'}$ gives, with some manipulation of level indices,

$$\bar{u}_{MS}(m) = 2m - mP_{NC}(F,m) + \sum_{L=1}^{L^{'}-1}[m - B^L \sum_{k=1}^{min(m,F)} k\, P(k,m,L)\ P_{NC}(F,k)], \tag{12}$$

*2) **MSS Scheme Analysis**:* This section analyzes the multi-slotted scheme presented before with the addition of a selective sleep command issued by the reader during the tag reading process. Tags read in a collision free frame will never be queried again and hence need not be put to sleep. If some tags are resolved at each level in frames with at least one collision (termed collision frames hereon) and they are put to sleep, the probability of collision should decrease with each further level probed. The main difference in the analysis of MSS scheme compared to the MS scheme is to decrease the total number of tags that reply at a node by the expected number of nodes that have been put to sleep after being read in collision frames, and make the necessary adjustments in decreasing probability of collisions. Thus the analytical expressions developed below rely on a recurrence on the number of tags read in collision frames before a certain level.

We begin by analyzing the average number of time slots required by the MSS scheme to read all tags. Let $\bar{s}(m,n)$ denote the expected number of tags read in collision frames at all levels from the root node to n, including n[6].

Now we can obtain the probabilities for the following behavior at a node when the tree is probed: no reply, I, collision C and collision free replies S.

We obtain:

$$P_I(m,L) = P(0, m - \bar{s}(m, L-1), L), \qquad L > 0 \tag{13}$$

where $m - \bar{s}(m, L-1)$ is the number of tags left after subtracting all tags resolved in collision frames at or above level $L-1$. Collision free replies are received if there is either only one tag in sub-tree or all tags use different slots in frame when responding. Let $P_{NC}(F,k)$ represent the probability that there are no collisions when k tags transmit within a frame of size F. Then,

$$P_S(m,L) = \sum_{k=1}^{m-\bar{s}(m,L-1)} P(k, m-\bar{s}(m,L-1), L)\ P_{NC}(F,k) = \sum_{k=1}^{min(m-\bar{s}(m,L-1),F)} P(k, m-\bar{s}(m,L-1), L)\ P_{NC}(F,k),\ L > 0 \tag{14}$$

where the final equality is due to the fact that if more than F tags transmit in a frame of size F, the probability of no collision is zero.

Since the sum of probability of collision free responses, no responses and colliding responses equals one,

$$P_S(m,L) = 1 - P_S(m,L) - P_I(m,L) = 1 - P(0, m-\bar{s}(m,L-1), L) - \sum_{k=1}^{min(m-\bar{s}(m,L-1),F)} P(k, m-\bar{s}(m,L-1), L)\ P_{NC}(F,k),\ L > 0. \tag{15}$$

Additionally, $P_C(m,L) = 1 - P_{NC}(F,m)$ for $L = 0$.

$\bar{s}(m,n)$ can be computed by summing up the expected number of of collision free replies in a collision frame at a node at level n over all nodes w at that level. This works because all tags resolved at levels above n will also be resolved at level n.

$$\bar{s}(m,n) = \sum_{w=0}^{B^n-1} \bar{r}(m,n) = B^n \bar{r}(m - \bar{s}(m,n-1), n), \tag{16}$$

where $\bar{r}(m - \bar{s}(m,n-1), n)$ is the expected number of tags read in a collision frame at a node at level n also taking into account the reduced number of tags at that level.

Now the expected number of tags read in a collision frame at a node is the product of expected number of slots with exactly one tag response among the F slots and the probability of a collision frame. The expected number of tags read in a frame is given by $F[\binom{k}{1}p^k(1-p)^{m-k}]$ with $p = 1/F$, where k is the number of tags responding to the query. By summing over all possible k, we get

$$\bar{r}(m - \bar{s}(m,n-1), n) = \sum_{k=1}^{m-\bar{s}(m,n-1)} k(1-1/F)^{k-1} P(k, m-\bar{s}(m,n-1), n)(1 - P_{NC}(F,k)),$$

[6] $\bar{s}(m,n)$ is not an integer, and hence we round it off to the nearest integer value when required in the equations given below. Simulation results of the protocol show negligible effects of this approximation when compared to analytical expressions derived here

which gives

$$\bar{s}(m,n) = B^n \sum_{k=1}^{m-\bar{s}(m,n-1)} k(1-1/F)^{k-1} \times P(k, m-\bar{s}(m,n-1), n)(1-P_{NC}(F,k)) \quad (17)$$

with $\bar{s}(m,0) = m(1-1/F)^{m-1}(1-P_{NC}(F,m))$. Then the probability of a node being visited

$$P_V(m-\bar{s}(m,L-1),L) = \begin{cases} 1 & L=0 \\ P_C(m,L-1) & L=1 \\ P_C(m-\bar{s}(m,L-2),L-1) & L>1 \end{cases}$$

By using similar arguments to those used to obtain Equation 7, now the average number of slots required to read all tags then is the product of number of time slots per node (and 1 slot for reader probes) and the average number of nodes visited, plus the average number of slots used by the reader to issue sleep commands during the arbitration process, say $\bar{z}(m)$. Thus,

$$\begin{aligned} \bar{t}_{MSS}(m) &= (F+1)\sum_{L=0}^{\infty}\sum_{w=0}^{B^L-1} P_V(m-\bar{s}(m,L-1),L) + \bar{z}(m) \\ &= (F+1)[1+\sum_{L=1}^{\infty} B^L P_V(m-\bar{s}(m,L-1),L)] + \bar{z}(m) \\ &= (F+1)[1+B.P_C(m,0) \\ &\quad + B\sum_{L=1}^{\infty} B^L P_C(m-\bar{s}(m,L-1),L)] + \bar{z}(m). \quad (18) \end{aligned}$$

Using 15 gives

$$\begin{aligned} \bar{t}_{MSS}(m) &= (F+1)[1+B(1-P_{NC}(F,m)) \\ &\quad + B\sum_{L=1}^{\infty} B^L\{1-P(0,m-\bar{s}(m,L-1),L) + \bar{z}(m) \\ &\quad - \sum_{k=1}^{min(m-\bar{s}(m,L-1),F)} P(k,m-\bar{s}(m,L-1),L)\ P_{NC}(F,k)\}]. \quad (19) \end{aligned}$$

The expected number of slots used by the reader to issue sleep commands, $\bar{z}(m)$, equals the total number of tags that have to be put to sleep which equals $\bar{s}(m,\infty)$ or $\bar{s}(m,L^{'})$ if $L^{'}$ is the limit used for number of levels.

Next, in calculating the energy consumption at the reader for the MSS scheme we are interested in the average number of queries $\bar{q}_{MSS}(m)$ and the average number of sleep commands issued, $\bar{z}(m)$. The latter was analyzed above. The average number of queries at the reader is simply the average number of nodes visited. Thus, based on our analysis in the previous subsection

$$\begin{aligned} \bar{q}_{MSS}(m) &= [1+B(1-P_{NC}(F,m)) \\ &\quad + B\sum_{L=1}^{\infty} B^L\{1-P(0,m-\bar{s}(m,L-1),L) \\ &\quad - \sum_{k=1}^{min(m-\bar{s}(m,L-1),F)} P(k,m-\bar{s}(m,L-1),L)\ P_{NC}(F,k)\}]. \quad (20) \end{aligned}$$

For the total average number of tags responses $\bar{u}_{MSS}(m)$, we begin by computing $\bar{s}(m,n)$ as above which is the number of tags read in collision frames, the ones that will be put to sleep.

$$\bar{s}(m,n) = B^n \sum_{k=1}^{m-\bar{s}(m,n-1)} k(1-1/F)^{k-1} \times P(k, m-\bar{s}(m,n-1), n)(1-P_{NC}(F,k)) \quad (21)$$

with $\bar{s}(m,0) = m(1-1/F)^{m-1}(1-P_{NC}(F,m))$.

Now, we need to find the number of tags resolved at each level, say $\bar{y}(m,n)$ by taking into account the tags put to sleep as well. This equals the sum of number of tags read in collision frames plus the tags read in collision free frames. Thus, we get

$$\bar{y}(m,n) = \bar{s}(m,n) + \sum_{w=0}^{B^n-1} \bar{r}(m,n) = B^n \bar{r}(m-\bar{s}(m,n-1),n), \quad (22)$$

where $\bar{r}(m-\bar{s}(m,n-1),n)$ is the expected number of tags read at a node at level n from collision free frames also taking into account the reduced number of tags at that level. We already have the tags read in collision frames as $\bar{s}(m,n)$ calculated above.

Now the expected number of tags read at a node is the expected number of slots with exactly one tag response among the F slots. The expected value of tags read then is given by $F[\binom{k}{1}p^k(1-p)^{m-k}]$ with $p=1/F$, where k is the number of tags responding to the query. By summing over all possible k, we get

$$\bar{r}(m-\bar{s}(m,n-1),n) = \sum_{k=1}^{m-\bar{s}(m,n-1)} k(1-1/F)^{k-1} P(k,m-\bar{s}(m,n-1)) \quad (23)$$

with $\bar{r}(0,m) = m(1-1/F)^{m-1}$.
Thus we have

$$\bar{y}(m,n) = \bar{s}(m,n) + B^n \sum_{k=1}^{m-\bar{s}(m,n-1)} k(1-1/F)^{k-1} P(k,m-\bar{s}(m,n-1),n)$$

with $\bar{y}(m,0) = \bar{s}(m,0) + m(1-1/F)^{m-1}$.

Then, the average number of tag responses is given by

$$\begin{aligned} \bar{u}_{MSS}(m) &= m + \sum_{L=1}^{\infty} m - \bar{y}(m,L-1) \\ &= 2m - [\bar{s}(m,0) + m(1-1/F)^{m-1}] + \sum_{L=1}^{L^{'}-1} m - [\bar{s}(m,L) \quad (24) \\ &\quad + B^L \sum_{k=1}^{m-\bar{s}(m,L)} k(1-1/F)^{k-1}.P(k,m-\bar{s}(m,L-1),L)] \end{aligned}$$

limiting the number of levels in the tree to $L^{'}$ as before.

*3) **MAS Scheme Analysis**:* With this scheme, for a query at level L, the probability of a tag responding is $B^{-(L+log_B F)}$ corresponding to fact that there are $B^{(L+log_B F)}$ nodes at level $L+log_B F$ and each tag has uniform probability to have the prefix of any of these nodes. The probability of k tags responding at their assigned slot in level L then is

$$P(k,m,L) = \binom{m}{k} p^k (1-p)^{m-k} \quad (25)$$

with $p = B^{-(L+log_B F)}$.
The probability of a collision at an assigned slot is then

$$\begin{aligned} P_C(m,L) &= P(k>1,m,L) = 1 - P(0,m,L) - P(1,m,L) \\ &= 1 - (1-B^{-(L+log_B F)})^m \\ &\quad - mB^{-(L+log_B F)}(1-B^{-(L+log_B F)})^{m-1}. \quad (26) \end{aligned}$$

A node w at level $L+log_B F$ is visited only if there is a collision at slot assigned to its tags at level L.

Then the probability of visiting a node in the tree is then given by

$$P_V(m,L) = \begin{cases} 1 & L=0 \\ P_C(m,L-log_B F) & L>0,\ L \bmod log_B F = 0 \\ 0 & \text{Otherwise} \end{cases}$$

The above equation takes into account the fact that only those levels are queried starting from level 0 that are multiples of $log_B F$ with the nodes on the other levels skipped.Each visit to a node in the tree uses up F slots of time. Thus the average running time can be calculated by multiplying the average number of nodes visited in the tree by $F+1$, where the 1 is due to the slot required by the reader probe.

$$\begin{aligned}\bar{t}_{MAS}(m) &= (F+1)\sum_{L=0}^{\infty}\sum_{w=0}^{B^L-1} P_V(m,L) \\ &= (F+1)[1+\sum_{L=log_B F}^{\infty} B^L P_C(m, L-log_B F)]J \\ &= (F+1)[1+\sum_{L=0}^{\infty} B^{L+log_B F} P_C(m,L)]J. \quad (27)\end{aligned}$$

where

$$J = \begin{cases} 1 & L \bmod log_B F = 0 \\ 0 & \text{Otherwise} \end{cases}.$$

Using 26 gives

$$\bar{t}_{MAS}(m) = (F+1)[1+B^{log_B F}\sum_{L=0}^{\infty} B^L\{1-(1-B^{-(L+log_B F)})^m - mB^{-(L+log_B F)}(1-B^{-(L+log_B F)})^{m-1}\}]J. \quad (28)$$

The upper limit of L, L' can again be calculated as before to transform the infinite tree to a finite tree.

The average number of reader queries is simply equal to the number of nodes queried of the binary tree including the leaves (as analyzed above in 28, without the multiplicative factor F),

$$\bar{q}_{MAS}(m) = [1+B^{log_B F}\sum_{L=0}^{\infty} B^L\{1-(1-B^{-(L+log_B F)})^m - mB^{-(L+log_B F)}(1-B^{-(L+log_B F)})^{m-1}\}]J, \quad (29)$$

where J is defined as before.

Let $\bar{s}(m,n)$ be the average number of tags whose ID was read by the tree search algorithm for all levels less than and including n. If $P_r(m,n) = P(1,m,n)$ be the probability of a single, collision free reply at a slot (among F such slots) in node w and level n,

$$\bar{s}(m,n) = F\sum_{w}^{B^n-1} P_r(m,n) = FB^n P_r(m,n) = B^{n+log_B F} P_r(m,n).$$

As before, the above equation results because if a tag is resolved at a level higher than n in the tree, it will also be resolved at level n. So just counting the resolved tags at level n suffices. The factor F results due to each node in tree having F slots with uniform probability of collision.

The average total number of tag responses $\bar{u}_{MAS}(m)$ can be found as follows. At the top level (level 0) of tree, all m tags reply to a probe. The number of replies at all lower levels considered (ones with $L \bmod log_B F = 0$) is equal to m minus the number of tags resolved received at all levels above it.

Thus, with

$$\begin{aligned}\bar{s}(m,n) &= B^{n+log_B F} P(1,m,n) \\ &= B^{n+log_B F}[mB^{-(n+log_B F)}(1-B^{-(n+log_B F)})^{m-1}] \\ &= m(1-B^{-(n+log_B F)})^{m-1} \quad (30)\end{aligned}$$

TABLE III

POWER CONSUMPTION VALUES USED FOR READER AND TAGS

Reader		Active Tag	
P_{Rtx}	P_{Rrx}	P_{Ttx}	P_{Trx}
825 mW	125 mW	35 mW	28 mW

we have

$$\begin{aligned}\bar{u}_{MAS}(m) &= m + \sum_{L=log_B F}^{L'} (m - \bar{s}(m, L-log_B F)J \\ &= m + \sum_{L=0}^{L'-log_B F}[m - B^{L+log_B F} P_r(m,L)]J \\ &= m + \sum_{L=0}^{L'-log_B F}[m - (m(1-B^{-(L+log_B F)})^{m-1})]J. \quad (31)\end{aligned}$$

V. NUMERICAL EVALUATIONS

Here we numerically evaluate the analytical expressions (coupled with our energy model) derived before for average reader energy consumption, total active tag energy consumption, and average number of time slots required for reading all tags by the MS, MSS and MAS schemes. Note that the numerical evaluation of the MS or MSS scheme with $F = 1$ provides the results for the QT protocol. We will be concerned only with the value of $B = 2$ henceforth since it is the most widely used value in practice. The power consumption values used in the evaluation are shown in Table III[7].

We begin by studying how the energy consumption of the reader and tags, and time slots required vary with the size of query frame, F and number of tags, m. Subsequently, we will look at the energy savings these protocols provide compared to QT and show the possible trade-off involved in terms of additional delay.

We look at values of m ranging from 8 to 128[8]. We believe this range of m to be most applicable for our target scenario of portable RFID readers. We study the MS and MSS schemes for F from 1 to 32, while we look at values of F from 2 to 128 as powers of two for the MAS scheme.

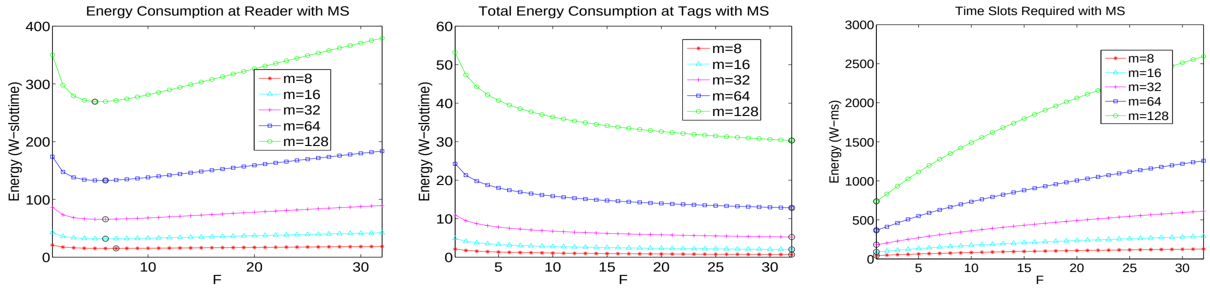
A. Evaluation of MS, MSS and MAS Schemes

Figures 5, 6, and 7 show the reader and tag energy consumption, and time slots required for all three schemes.

The reader energy consumption decreases with increasing F to a point, after which it increases again. Initially, increasing F reduces the collisions among tag responses and results in a decrease in energy consumption. As F is increased further, however, the benefit due to decrease in collisions is offset by the greater energy consumption due large number of slots required per frame. It can be seen that the MAS scheme incurs the least energy consumption among all schemes and grows slowly with increasing F. Interestingly, the reader energy consumption of MAS suggests that reductions in collisions is more pronounced for some F than others.The MSS scheme is more energy-efficient than MS and allows us to conclude that minimizing

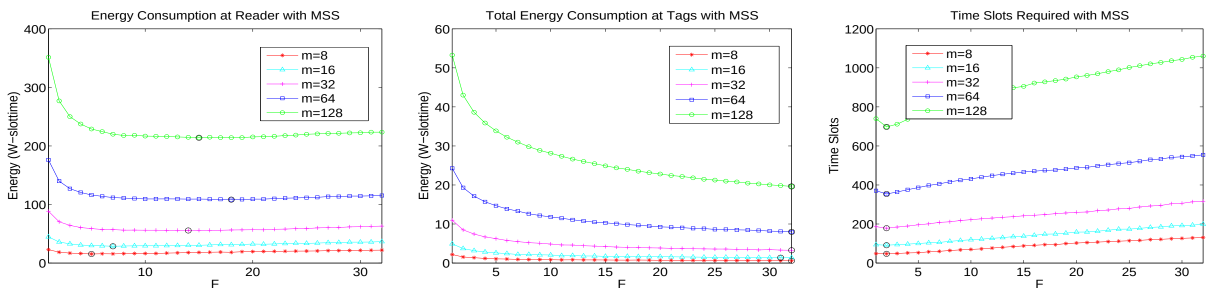
[7]These values have been obtained from the specifications of the Philips MF RC530 Reader IC and Infineon TDA5250 Active Tag and Wireless Sensor. In general, it is the ratio of transmit to receive power that matters than absolute values. The greater this ratio, greater the energy savings compared to the QT protocol.

[8]We do not show the results for $m < 8$ to keep the plots easy to read; the results for these number of tags follow the same trends as of those shown here, except for the special case of $m = 1$ which is discussed later in this section.



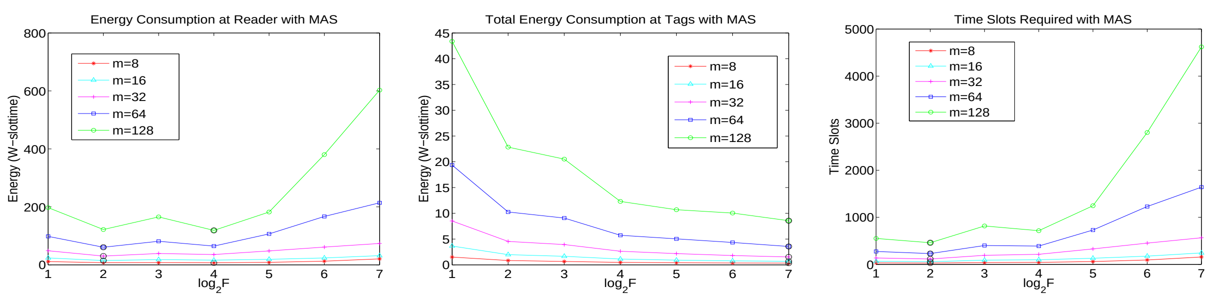
(a) Reader Energy Consumption (b) Tag Energy Consumption (c) Time Slots Required

Fig. 5. Reader energy consumption, tag energy consumption, and total timeslots required with MS for varying F and m. Circled markers show the point of minimum value.



(a) Reader Energy Consumption (b) Tag Energy Consumption (c) Time Slots Required

Fig. 6. Reader energy consumption, tag energy consumption, and total timeslots required with MSS for varying F and m. Circled markers show the point of minimum value.



(a) Reader Energy Consumption (b) Tag Energy Consumption (c) Time Slots Required

Fig. 7. Reader energy consumption, tag energy consumption, and total timeslots required with MAS for varying F and m. Note that for MAS, F has to be a power of B. Circled markers show the point of minimum value.

collisions by putting tags to sleep has greater benefits and outweighs the extra energy required to issue sleep commands. From a more fundamental perspective, it is interesting to note that with the MS scheme, the optimal value of F for reader energy consumption decreases slowly with increasing m. The reason is that with greater number of tags contending, multiple slots within a frame are not as useful due to the higher load per frame. Thus, using smaller F wastes less time slots (and energy), until the queries descend to a level where multiple slots becomes more useful. In fact, for some ratios of $\frac{P_{Rtx}}{P_{Rrx}}$, $F = 1$ can become optimal as the number of tags increases. This scenario, however, rarely occurs in practice - in the application setting under consideration with portable RFID systems, the binary trees are always sparse with small m. Further, the ratio $\frac{P_{Rtx}}{P_{Rrx}}$ tends to be large (similar to the one shown in Table III) with the reader transmit power much greater than the receive power. With the MSS scheme, the optimal values of F for reader energy consumption does not decrease with increasing m. This is because sleep commands reduce tag contention beginning at a much higher level than the MS scheme providing greater benefits with large F. The structured assignment of slots enables similar benefits with the MAS scheme.

Increasing the value of F is always beneficial for tag energy savings. Larger F decreases collisions among tag responses, with the energy burden of using a large F lying solely on the reader. The tag energy consumption behavior is similar to reader energy consumption trend for the three schemes with MAS using up the least energy due to fewest collisions.

There is a general trend of greater number of time slots required with increase in F. Surprisingly, the rate of increase in time slots required for small values of F is very slow with MSS and MAS hinting at the possibility that they may have utility for faster tag identification as well. For the MSS scheme, using $F = 2$ takes less time slots to identify tags than with $F = 1$ for the values of m considered. This is not seen with the MS scheme due to its insufficient reduction in collisions. We show our comparisons with QT in greater detail later in this section.

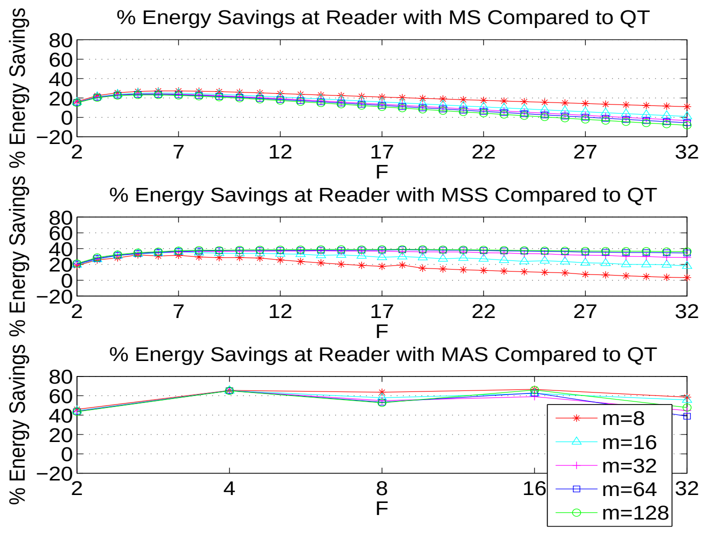


Fig. 8. Percentage energy savings at the reader for all three schemes compared to the QT protocol

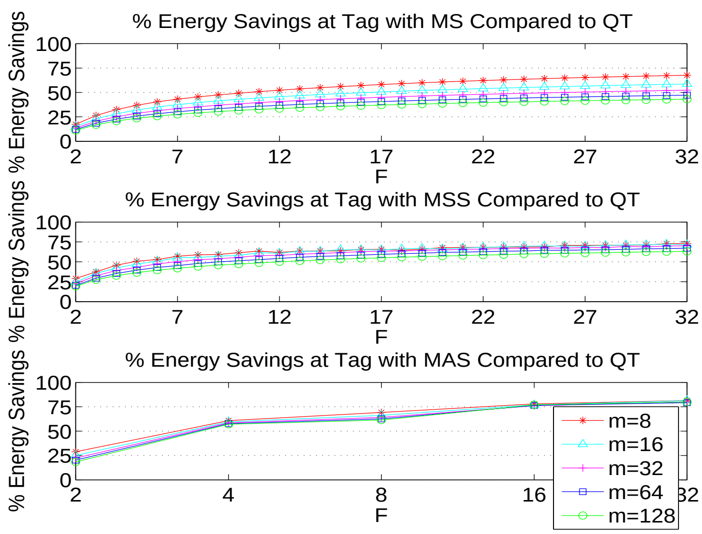


Fig. 9. Percentage energy savings at tags for all three schemes compared to the QT protocol

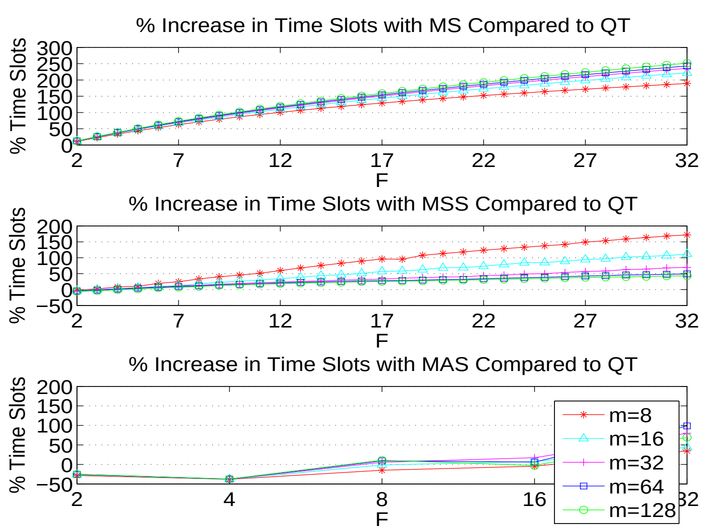


Fig. 10. Percentage increase in time slots for all three schemes compared to the QT protocol

B. Comparison of Our Schemes with QT

Here we look at the following for all three schemes compared to the QT protocol: average energy savings at the reader and active tags as well as the possible increase in time slots required to read all tags as a function of F. We could have shown these values for only the optimal values of F for each scheme, but as we explain later in this section, there are scenarios where it may be difficult to select this optimal value. Thus, considering energy savings for different F is a useful step toward understanding the behavior of the protocols. The optimal values of F for these plots remain the same as shown in plots Figures 5, 6 and 7 for the three schemes.

Energy Savings at the Reader: Figure 8 shows the plots for all three schemes. The MS and MSS scheme save upwards of 30% energy at the reader while the MAS scheme roughly doubles the savings over these schemes when compared to QT. The fewer number of nodes visited in the tree (confirmed separately, but not shown here directly), along with the structured responses of tags, results in more tags read per query and thus the result. Interestingly, the number of tags do not seem to have much impact on the magnitude of energy savings. This is significant because it allows energy savings without accurate knowledge of the number of tags in the field, a very common case.

Total Energy Savings at Active Tags: The energy savings to read all active tags by the three schemes are shown in Figure 9. All three schemes show increasing benefits in reducing energy consumption at active tags with increasing values of F. The greater the number of slots, the greater the number of tag responses read by the reader at higher levels of the binary tree, and thus fewer the responses required. The MAS scheme achieves the greatest energy savings, with a rapid increase in % savings with increasing F.

Time Slots Required: Since all schemes lead to energy savings at both the reader and tags, it becomes interesting to differentiate between them based on the number of time slots required by the reader to read all tags; the corresponding plots are shown in Figure 10. The MS scheme takes the most number of time slots of all three schemes and has the greatest rate of increase in time slots required. The MSS and MAS schemes have a slow rate of increase in time slots with increase of F. In fact, for $F = 2, 4$ these protocols use fewer time slots than the QT protocol itself for all $m \geq 2$ (values of $m = 2, 4$ have not been shown in the plots in this section, but were considered as well).

Discussion: Based on the comparisons above between the three schemes, it can be concluded that the MAS scheme provides the greatest energy savings at the reader as well as active tags for a given value of F with the least increase in time slots as compared to the other schemes[9]. The structured assignment of slots to specific subtrees allows the reader to distinguish between the tags responsible for collisions in slots reducing the number of nodes visited in the tree. Moreover, the MAS and the MSS schemes also reduce the time required to read all tags for small values of F.

C. Maximizing reader energy savings given a time constraint

Earlier in this section we showed the optimal values of F that should be used to maximize reader energy savings or tag energy savings and the corresponding number of expected time slots required to identify all tags. Here, we ask a more practical question: *Given an*

[9]We have further validated the results presented here using simulations to ensure correctness due to the usage of any approximations in our analysis (e.g. rounding off $\bar{s}(m, n)$)

average time constraint to read all tags (count possibly unknown), what value of F maximizes reader energy savings?

This can be easily solved based on the developed analytical framework. For example, consider the plot showing expected time slots required to read all tags by MS scheme in Figure 5c. Given a time bound, we can calculate a bound on F_{max} since time slots required increases monotonically with F. Next, we can find the F, $F \in [1, F_{max}]$ in Figure 5a that optimizes reader energy savings. Since m may be unknown, we can use some known upper bound on m, M and calculate F as above. Note that the same approach can be applied for the MAS scheme, even though the expected time slots does not monotonically increase with F, by not using $F = 8$ which is the outlier. For the MSS scheme, we need to consider values of $F \geq 2$ only because $F = 2$ allows faster identification than $F = 1$ as shown in Figure 6c. It can be seen that selecting values of F with the aim of maximizing reader energy savings provides tag energy savings as a useful 'byproduct'.

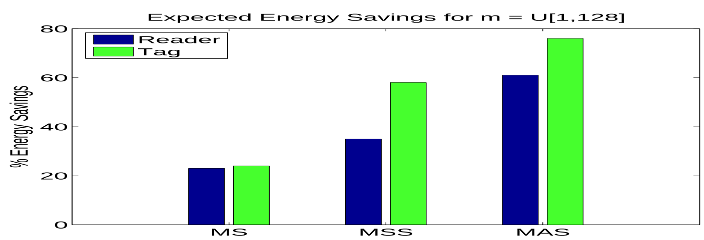


Fig. 11. Expected energy savings at the reader and tags with all three schemes when encountering number of tags uniformly distributed within a range of 1 to 128 (i.e. $m = U[1, 128]$) and an upper bound on tag count, $M = 128$.

When the given time bound is too small to complete reading M tags, small values of F can be chosen. For example, for the MS scheme, $F = 1$ is expected to take the shortest time, while for the MSS and MAS schemes, $F = 2, 4$ are better values for most values of m. Another possible approach would be to estimate the number of tags in the field as a first step before the identification process and selecting F accordingly [28]. This two step approach, however, will introduce additional delays and cost energy; it is interesting future work to consider the benefits of such approaches with our schemes.

To give a sense of energy savings at the reader and tags when encountering tags randomly with all three schemes, we select a value of F as outlined above with an upper bound $M = 128$. This F is subsequently used for all values of m ranging from 1 to 128. The results obtained correspond to the energy savings achievable when encountering tags uniformly distributed in that range[10]. We used times slots of size 1ms with a time constraint of 1100 ms to read all tags. The values of F used with MS, MSS and MAS schemes were 4, 15 and 16 respectively. The corresponding energy savings are shown in Figure 11. It can be observed that energy savings are significant in spite of relying only on an upper bound on number of tags. We observed that except at $m = 1$ (where $F = 1$ is the optimal value), all three schemes were able to save energy. In most practical settings, this scenario arises when the reader is pointed at the lone tag directly with no other tag in range. For this scenario, the reader could be equipped with a special push button that sends queries with $F = 1$. When the tag count is unknown, the above mentioned method of using an upper bound on m can be utilized.

[10]An example application scenario is the scanning of a box which might contain a certain number of tagged items, whose count varies within a certain range

VI. Evaluation for unknown distribution of Tag IDs

So far we have considered tag IDs that span the entire ID space and are assigned with a uniform distribution. The standards used in practice could mandate the use of different fields within an ID, with each field attached to a specific property. The resulting tag IDs, thus may not be uniformly distributed across the entire ID space. For example, in the Class-1 Generation-2 UHF RFID standard [13], a tag identifier is divided into four fields: tag class, designer identity, model number, and tag serial number. Except the tag serial number, all other fields may be correlated across tags with common prefixes. In this section we will study the performance of our schemes through simulation for an unknown distribution of tag IDs. Simulations allow us to study the performance of our protocols for varying, unknown levels of correlation among tag IDs, for specific values of F.

If the tag IDs are correlated, identification takes longer with binary tree based protocols like QT (and also our protocols) as is well documented in previous work [21], [22]. The reader has to descend the binary tree until it reaches a level below the common prefix to begin distinguishing tag IDs. It is expected that the random selection of slots within a frame of size F in the MS and MAS schemes will prove useful when identifying tags with any common prefixes by allowing tags to be read even at the 'common prefix' nodes in the binary tree .

A. Experimental Setup

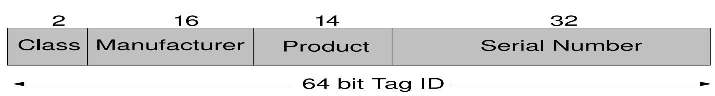


Fig. 12. Tag ID structure used in our non-uniform tag ID distribution experiments

Each data point shown in the simulation plots of Figures 13, 14 and 15 is the average of 1000 runs. 95% confidence intervals are shown, but are too small to be seen clearly. The power values used for the energy model are the ones previously used in our numerical evaluations as given in Table III. The tag id structure and field lengths used are shown in Figure 12. The serial number field of each ID was generated randomly from a uniform distribution. Within the first three fields, we generated possible values x from a geometric distribution with parameter β as follows

$$P(x) = \frac{(\beta - 1)\beta^r}{\beta^{2^k} - 1}, \;\; r = 0 \text{ to } 2^k - 1, \; 0 < \beta < 1 \qquad (32)$$

with x subsequently converted to binary format. When β tends to 1, the generated values tend towards a uniform distribution within the field. When β tends to 0, all generated values tend to be the same, resulting in maximum correlation among IDs. The value of β was varied to study the performance of all three schemes for different levels of correlation in the first three fields. In reality, tag IDs can be derived from any distribution. We specifically use the geometric distribution for varying parameter values to study effect of correlated tag ID distributions because of their known detrimental effects on performance of binary search tree based protocols. Using $\beta = 1 - 10^{-5}$ provides a tag ID distribution close to the uniform tag ID distribution considered earlier in the paper.

B. Comparison of our schemes with QT for various tag ID distributions

We use a constant value of $m = 32$ for the plots shown and consider F from 2 to 32 as powers of two for all three schemes (with MAS, that is a requirement). We found this value of m to be representative for all other values of m in the range 2 to 128.

***Comparison of Energy Savings at the Reader**:* Of the three schemes, MAS still provides the greatest energy savings across all values of F and β considered compared to QT and provides benefits similar to the case of uniform tag ID distribution. Reading correlated tag IDs require more queries, and hence greater energy with MAS, but there is a corresponding increase with QT as well which allows energy savings comparable to the uniform distribution case. The MSS scheme provides comparable benefits to MAS for all values of β except $\beta = 0.01$. Moreover, the energy savings at the reader do not decline with increasing F unlike MAS, showing the benefits of randomly selected slots within a frame. This benefit requires F to be large enough to provide opportunities to read some tags a frame. The MS scheme is the worst affected, with QT outperforming MS for smaller values of β. This happens due to a combination of two factors affecting MS scheme that is exacerbated due to common prefixes: reader wastes energy for multiple slots at a level, and identified tags keep responding when queried.

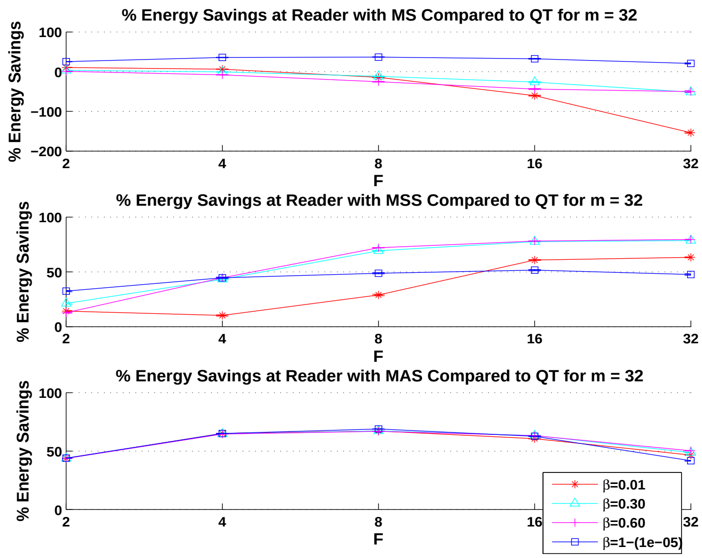


Fig. 13. Percentage energy savings at the reader for all three schemes compared to the QT protocol for various levels of common prefixes in tag IDs

***Comparison of Total Energy Savings at Active Tags**:* The MAS scheme still provides most energy benefits among all three schemes across all F and β, but the MSS scheme provides much greater energy savings for larger F, similar to the case with reader energy savings. The MS scheme is able to outperform QT for all values of β because tag energy consumption is unaffected by the reader using multiple slots in a frame, while deriving benefits from the fewer responses required.

***Comparison of Increase in Time Slots**:* In terms of time slots required to read all tags compared to QT, MAS protocol requires less time for small values of F. The MSS scheme typically requires less time than QT for all F except for highly correlated tag IDs with $\beta = 0.01$. The time slots required by the MS scheme increases rapidly with increase in F and will be useful only for small values of F and loose time constraints.

C. *Selection of F*

Here we consider what values of F are good choices for the three schemes when the tag ID distribution is unknown, possibly non-uniform. Without an analytical framework to predict performance

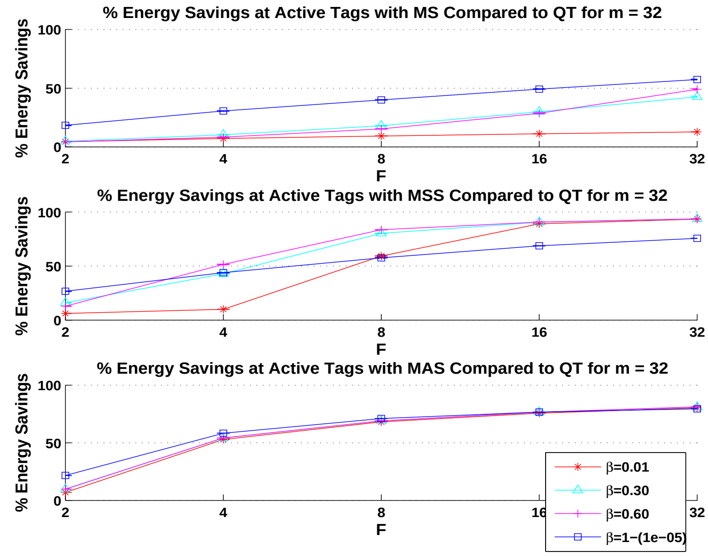


Fig. 14. Percentage energy savings at tags for all three schemes compared to the QT protocol for various levels of common prefixes in tag IDs

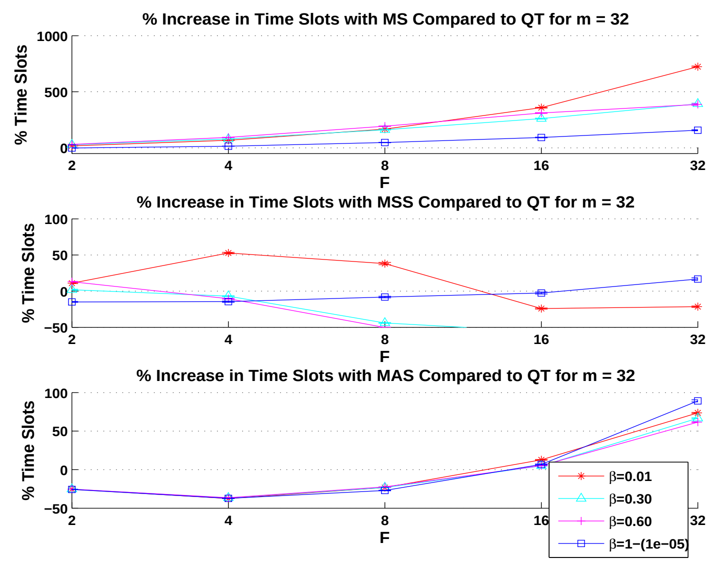


Fig. 15. Percentage increase in time slots for all three schemes compared to the QT protocol for various levels of common prefixes in tag IDs

for a specific number of tags (since the underlying ID distribution is unknown), all schemes might need to be a bit conservative in selecting values of F to ensure time constraints are met for all possible distributions.

For the MS scheme, low values of F are preferable due to rapid increase in time slots required and no energy saving benefits with larger F. For values of $F = 2$ or 4, Figure 13 shows that 5-20% reader savings are possible depending on the distribution. Similar energy savings at tags can be achieved with these values of F as well. In contrast, for the MSS scheme, values of F closer to the number of encountered tags take less time and provide more energy savings for different ID distributions. Thus, if m is known, F can be selected appropriately. When m is unknown, a value of F can be chosen based on time constraint only for an upper bound on m. This proves a bit difficult since the ideal value of F varies for different distributions and $F = 2$ or 4 are the safer choices.. With these values, close to 20-40% reader and tag energy savings can be obtained for most tag ID distributions. The energy savings with MAS scheme behaves similarly across different distributions considered and makes selection of F easier even for unknown m. A value of $F = 4$ is most suitable since it minimizes time slots required, while at the same time providing the maximum, or close to maximum energy savings around 60% for both the reader and tags.

VII. Conclusions

We presented an approach of using multiple slots per node of a binary search tree to reduce collisions among tag responses to provide for energy-aware RFID tag arbitration by the reader with provable upper bounds on identification time like the Query Tree Protocol. Three different variants of this approach were explored to provide insight on design choices to reduce energy consumption. Our analytical framework was used to predict the average case performance of these protocols for different input parameters with both known and unknown tag ID distributions. We showed that all three protocols were capable of reducing energy consumption at the reader as well as active tags. Further, two of the protocols demonstrated potential in reducing tag identification delay as well.

Acknowledgments

This work was supported in part by the NSF grant CCF-0329794.

References

[1] M. Jo, C.-G. Lim, and E. W. Zimmers, "RFID tag detection on a water object using a backpropagation learning machine," *KSII Transactions on Internet and Information Systems*, vol. 1, pp. 19–32, 2007.

[2] M. Jo and H. Youn, "Intelligent recognition of rfid tag position," *Electronics Letters*, vol. 44, no. 4, pp. 308–310, 2008.

[3] J.-J. Kang, D.-J. Lee, C.-C. Chen, J. Whitaker, and E. Rothwell, "Compact Mobile RFID Antenna Design and Analysis Using Photonic-assisted Vector Near-field Characterization," in *IEEE International Conference on RFID*, Las Vegas, NV, USA, April 2008, pp. 81–88.

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

[5] D. Bertsekas and R. Gallagher, *Data Networks*, 2nd ed. Prentice-Hall, Inc., 1992.

[6] M. A.-I. Center, "Draft protocol specification for a 900 MHz class of radio frequency identification tag," February 2003.

[7] C. Law, K. Lee, and K. Y. Siu, "Efficient memoryless protocol for tag identification," in *Proceedings of DIALM*, August 2000, pp. 75–84.

[8] N. Abramson, "The Aloha system - Another alternative for computer communications," in *Proceedings of Fall Joint Computer Conference AFIPS*, vol. 37, 1970, pp. 281–285.

[9] R. Metcalfe, "Steady state analysis of a slotted and controlled Aloha system with blocking," in *Proceedings of 6th Hawaii Conference System Science*, 1973.

[10] V. Anantharam, "The stability region of the finite-user slotted ALOHA protocol," *IEEE Transactions on Information Theory*, vol. 37, no. 3, pp. 535–540, May 1991.

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

[12] J. E. Wieselthier, A. Ephremides, and L. Michaels, "An exact analysis and performance evaluation of framed ALOHA with capture," *IEEE Transactions on Communications*, vol. 38, no. 2, pp. 125–137, 1989.

[13] "EPC radio frequency identity protocols class-1 generation-2 UHF RFID protocol for communications at 860 mhz-960mhz, version 1.0.9."

[14] C. Floerkemeier and M. Wille, "Bayesian transmission strategy for framed ALOHA based RFID protocols," in *Proceedings of IEEE International Conference on RFID (IEEE RFID 2007)*, March 2007.

[15] S.-R. Lee, S.-D. Joo, and C.-W. Lee, "An enhanced dynamic framed slotted aloha algorithm for RFID tag identification," in *Proceedings of Mobiquitous*, 2005, pp. 98–113.

[16] H. Vogt, "Efficient object identification with passive RFID tags," in *Proceedings of International Conference on Pervasive Computing*, 2002, pp. 98–113.

[17] D. Hush and C. Wood, "Analysis of tree algorithms for RFID arbitration," in *Proceedings of IEEE International Symposium on Information Theory*, 1998, p. 107.

[18] J. L. Capetanakis, "Tree algorithms for packet broadcast channels," *IEEE Transactions on Information Theory*, vol. 25, no. 5, pp. 505–515, September 1979.

[19] J. Moseley and P. Humblet, "A class of efficient contention resolution algorithms for multiple access channels," *IEEE Transactions on Communications*, vol. 33, pp. 145–151, 1985.

[20] M. A. Bonuccelli, F. Lonetti, and F. Martelli, "Exploiting ID knowledge for tag identification in RFID networks," in *Proceedings of Pe-WASUN'07*, October 2007.

[21] J. Myung and W. Lee, "Adaptive Binary Splitting: A RFID tag collision arbitration protocol for tag identification," *ACM Mobile Networks and Applications*, vol. 11, no. 5, pp. 711–722, 2006.

[22] A. Sahoo, S. Iyer, and N. Bhandari, "Improving RFID systems to read tags efficiently," KRSIT, IIT Bombay, Tech. Rep., June 2006.

[23] D. K. Klair, K.-W. Chin, and R. Raad, "An Investigation into the Energy Efficiency of Pure and Slotted Aloha Based RFID Anti-Collision Protocols," in *Proceedings of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (IEEE WoWMoM 2007*, Espoo, Finland, June 2007, pp. 1–4.

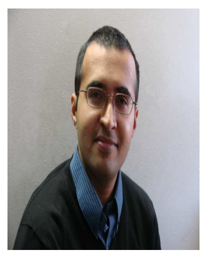
[24] F. Zhou, C. Chen, D. Jin, C. Huang, and H. Min, "Evaluating and optimizing power consumption of anti-collision protocols for applications in RFID systems," in *ISLPED '04: Proceedings of the 2004 international symposium on Low power electronics and design*, 2004, pp. 357–362.

[25] D. R. Hush, "A comparison of arbitration algorithms for RFID," Micron Communications, Inc., Boise, ID, USA, Tech. Rep., 1997.

[26] "EPC generation 1 tag data standards version 1.1 rev. 1.27, EPC Global Inc. standard specification," May 2005.

[27] D. Simplot-Ryl, I. Stojmenovic, A. Micic, and A. Nayak, "A hybrid randomized protocol for RFID tag identification," *Sensor Review*, vol. 26, no. 2, pp. 147–154, 2006.

[28] M. Kodialam and T. Nandagopal, "Fast and reliable estimation schemes in RFID systems," in *Proceedings of ACM MOBICOM*, Los Angeles, CA, USA, September 2006, pp. 322–333.



Vinod Namboodiri Vinod Namboodiri is currently an Assistant Professor at the Department of Electrical Engineering and Computer Science at Wichita State University, USA. He graduated with a Ph.D. in Electrical and Computer engineering from the University of Massachusetts Amherst in 2008. His research interests include designing MAC and routing protocols, and developing energy-efficient protocols and algorithms for different wireless technologies like Wireless LANs, RFID Systems, Wireless Sensor Networks, and Wireless Mesh Networks. He has served on the technical program committee of IEEE GLOBECOM and is an active reviewer for numerous journals and conferences in the wireless networking and mobile computing areas.

Lixin Gao Lixin Gao (S'93, M'96) is a professor of Electrical and Computer Engineering at the University of Massachusetts, Amherst. She received her Ph.D. degree in computer science from University of Massachusetts at Amherst in 1996. Her research interests include Internet routing, network security, network virtualization, multimedia networking, and energy efficient wireless communication. Between May 1999 and January 2000, she was a visiting researcher at AT&T Research Labs and DIMACS. She received an Alfred P. Sloan Fellow, and an NSF CAREER Award. She has served on numerous technical program committees/organizing committees of prestigious conferences such as ACM SIGCOMM, ACM SIGMETRICS, and IEEE INFOCOM, and on the Editorial Board of IEEE Transactions on Networking.