

University of Massachusetts Amherst

ScholarWorks@UMass Amherst

---

Electrical and Computer Engineering Faculty  
Publication Series

Electrical and Computer Engineering

---

2010

## Energy-Aware Tag Anticollision Protocols for RFID Systems

Vinod Namboodiri

*University of Massachusetts Amherst*

Lixin Gao

*University of Massachusetts Amherst, lgao@ecs.umass.edu*

Follow this and additional works at: [https://scholarworks.umass.edu/ece\\_faculty\\_pubs](https://scholarworks.umass.edu/ece_faculty_pubs)

---

### Recommended Citation

Namboodiri, Vinod and Gao, Lixin, "Energy-Aware Tag Anticollision Protocols for RFID Systems" (2010).  
*IEEE Transactions on Mobile Computing*. 2.  
<https://doi.org/10.1109/PERCOM.2007.14>

This Article is brought to you for free and open access by the Electrical and Computer Engineering at ScholarWorks@UMass Amherst. It has been accepted for inclusion in Electrical and Computer Engineering Faculty Publication Series by an authorized administrator of ScholarWorks@UMass Amherst. For more information, please contact [scholarworks@library.umass.edu](mailto:scholarworks@library.umass.edu).

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

Vinod Namboodiri, Lixin Gao

Department of Electrical and Computer Engineering

University of Massachusetts at Amherst

Email: {vnambood, lgao}@ecs.umass.edu

## Abstract

*Energy consumption of mobile readers is becoming an important issue as applications of RFID systems pervade different 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. The problem of tag arbitration in RFID systems is considered with the aim of trading off time for energy savings at the reader. The approach of using multiple time slots per node of a binary search tree is explored through three anti-collision protocols that aim to reduce the number of colliding responses from tags. This results in fewer reader queries and tag responses and, hence, energy savings at both the reader and tags (if they are active tags). An analytical framework is developed to predict the performance of our protocols, with the numerical evaluation of this framework validated through simulation. It is shown that all three protocols provide significant energy savings when compared to the existing Query Tree protocol while sharing the deterministic and memoryless properties of the latter.*

## 1 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)<sup>1</sup>. RFID technology has found applications in inventory tracking, data handling, object identification and more. By writing specific data to tags and reading and/or modifying them later as required, RFID

<sup>1</sup>Semi-passive tags also exist that use battery supply for powering up but rely on the reader's signal for transmission

presents a dynamic front compared to the static bar code system.

For commercial systems located at industrial premises, readers usually are fixed and connected to the AC mains supply with tags moving by them. Tag reading throughput is important when scanning continuous streams of tagged items. There are, however, numerous other applications like inventorying items in supermarkets, garages, refrigerators, closets and more, where a user carries a reader that is battery-operated and scans tags. A mobile phone with reader functionality being used for the above applications is a typical instance of this scenario. Larger number of reader queries required to read all tag IDs results in faster depletion of the batteries requiring more frequent recharging or replacement. The main requirement in such applications is that the tag reading time should not be perceptible to the user (or any delay is within acceptable limits) and the energy consumption of the reader is minimized. In such cases, a delay in reading tags can be traded off for smaller energy consumption at the reader. This would enable greater operating lifetimes for the reader and make RFID technology more pervasive. Energy consumption has not been looked at in RFID systems previously, but with more and more applications requiring mobile readers, it has become an important resource to conserve if 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.

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 the tags so that it can then read the data stored in these tags selectively when required, or simply to identify the item. 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, which 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 by time (or time slots within a frame if using frame-slotted ALOHA). They are probabilistic in nature and simple [3], thus enabling simpler reader and tag implementations. However, they do not offer any guarantees and a tag ID may not be read for a very long time. In contrast, the binary tree search protocols are deterministic in nature [2, 8]. 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). The design of anti-collision protocols for RFID systems is challenging since the tags cannot be assumed to be able to communicate with each other directly and may not be capable of storing states of the arbitration process in their memory.

We present three such novel anti-collision protocols that combine the ideas of a binary tree search protocol with frame-slotted ALOHA to provide for deterministic schemes that are energy-aware. The Multi-Slotted (MS) scheme relies on using multiple-slots per query to reduce the chances of collision among the 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 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 is also chosen based on a specified average time constraint within which all tags IDs must be read. These schemes are simple and do not require the often resource constrained tags to store any state of the communication between themselves and the reader. 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 further validate our analysis by simulations and demonstrate significant reductions in energy consumption at the reader 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.

The rest of the paper is organized as follows: Section 2 gives an overview of related work in the area of RFID arbitration of tags with a brief description of the Query Tree Protocol. Section 3 presents our anti-collision protocols in detail. Section 4 presents the energy models used at the reader and active tags and the derivation of an analytical framework for the protocols along with their numerical evaluation. The validation of our analysis through simulation is presented in Section 5 and demonstrates the energy savings possible by our protocols. Concluding remarks and possible future work are given in Section 6.

## 2 Related Work

The simplicity of Aloha based protocols had led to their adoption in early RFID based systems. For most RFID based systems, however, it is important that an anti-collision protocol is deterministic, i.e 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. The two most popular binary search protocols are the Binary Tree (BT) protocol [2] and the Query Tree (QT) protocol [8]. Both schemes work by splitting tag IDs using queries from the reader 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. Thus our main focus will be on the QT protocol only and we describe it briefly below. We will be using 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 deterministic tag anti-collision protocol which is memoryless with tags requiring no additional memory except that required to store their ID [8] (our protocols share these properties as well). The algorithm consists of rounds of queries and responses. In each round, a reader sends a query as a prefix and tags with this prefix respond back with the remaining bits of their ID. When more than one tag responds to a prefix query the reader comes to know 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. That tag is never queried again because the reader will move onto other prefixes with a query never sent whose prefix matches that of the resolved tag ID. A working example of this protocol is shown in Figure 1. A binary tree with tags A, B and C is shown with the tags positioned at the highest level at which they can be uniquely identified. 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 is 11. The corresponding activity of each node in the tree is shown to the immediate left or

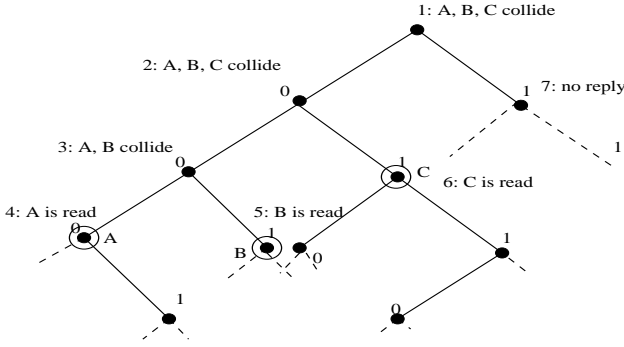


Figure 1. Example of Query Tree Protocol

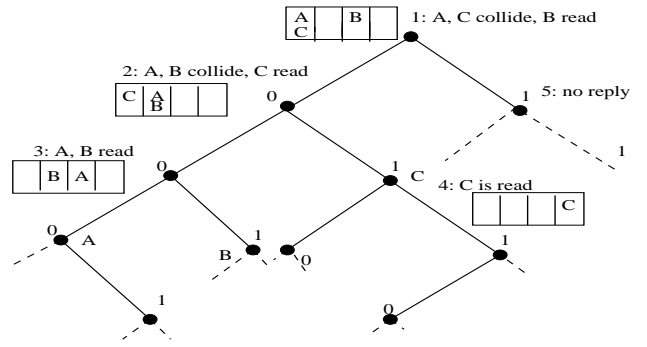


Figure 2. Example of Multi-Slotted Scheme

right.

The authors of [8] also state some optimizations to the QT protocol. There have also been some optimizations targeting the minimization of 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 [12]. This optimization is supplemental to our work and can be used to provide power savings at passive tags which is critical for their working range. Another recent work with tag anti-collision protocols also looks 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 are still present in the reader's interrogation zone [9]. There have also been studies on how the QT protocol can be improved to handle tag IDs which could have some common prefixes [11]. 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 enabling an efficient trade off between the time to read all tags and energy consumption at the reader.

Algorithms for arbitration using trees have been around for a while [3]. The authors of [6] were one of the first to consider these for reading tags in RFID systems. They analyze a binary tree search algorithm extensively producing analytical results for the average times slots, reader messages and tag responses. The work in [5] also briefly mentions a multi-slotted scheme that uses less time slots than a binary tree search algorithm. The EPC Class 1 Gen 1 standard mentions a ping command which is used to read tags by filtering their IDs for a sub-sequence of bits and collecting responses into 'bins' [1]. These latter two pieces of work have some similarity to our MAS scheme in their approach to gain structured responses to queries. To the best of our knowledge, our work (by varying the number of slots per query) is the first that focuses on energy awareness in RFID systems, and specifically tag anti-collision protocols.

### 3 Energy-Aware Anti-Collision Protocols

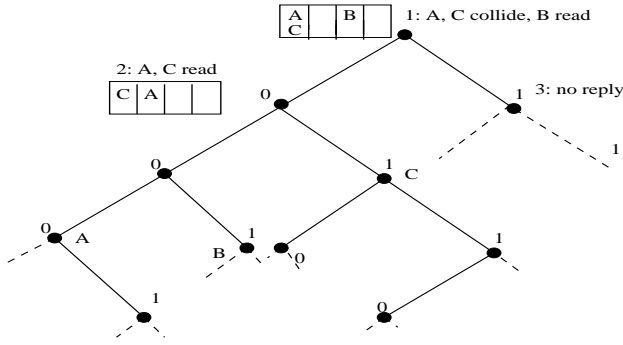
Our approach to energy-aware anti-collision protocols for RFID systems is to combine the deterministic nature of binary search algorithms along with the simplicity of frame-slotted ALOHA to reduce the number of tag response collisions. The QT protocol relies on colliding responses to queries that are sent to internal nodes of a tree to determine the location of tag IDs. No effort is, however, made to use these queries to get tag IDs as well. The idea of our protocol is to allow tags to transmit responses within a slotted time frame and thus, try to avoid collisions with responses from other tags. The queries are used to find colliding sub-trees as well as to read tag IDs. This two-pronged approach allows more tags to be read with the same number of queries as before, reducing the energy costs at the reader, while at the same time requiring fewer tag responses in all. The selection of number of time slots per query is based on the input of an average time constraint to read all tags, thus trading off time to achieve more energy savings if possible.

We look at multiple variants of this approach with the aim of finding the best among them or at least determining which protocols work best for particular scenarios.

#### 3.1 Multi-Slotted Scheme

The Multi-Slotted (MS) scheme works as follows. At each node of the  $B$ -ary tree<sup>2</sup>,  $F$  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 without collision, but since reader does not know to which subtrees the colliding tags belong to, it still has to query all the sub-trees.

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



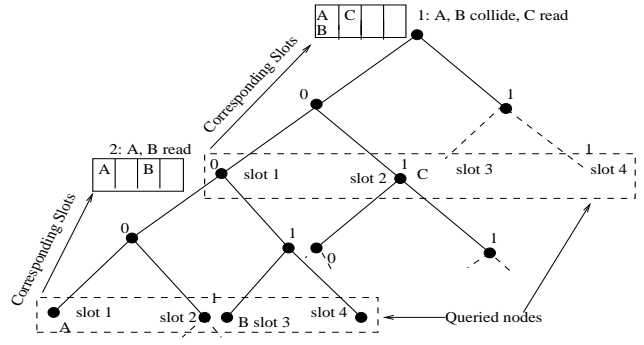
**Figure 3. Example of Multi-Slotted Scheme with Selective Sleep**

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.

A binary tree with frame size  $F = 4$  with tags A, B and C is shown in Figure 2 with the tags positioned at the highest level at which they can be uniquely identified. 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. Total reader queries is 5 while total tag responses sent is 9. The corresponding frame and activity of each node in the tree is shown to the immediate left or right. Tags may be read multiple times because they are not put to sleep.

### 3.2 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. Thus tag responses are reduced, resulting 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, and our following analysis along with the numerical evaluation presented later tries to discover how worthwhile the usage of selective sleep commands are in reducing reader energy consumption.



**Figure 4. Example of Multi-Slotted Scheme with Assigned Slots**

The working example for the MSS scheme is shown in Figure 3. 3 reader queries are used along with 1 sleep command for tag B, thus requiring 4 reader messages in all. Total tag responses sent is 5.

### 3.3 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 further nodes 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 so that the reader can 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<sup>3</sup>  $v$  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  $qx\cdots x$  slot, where  $x$ 's are the  $\log_B F$  bits after prefix  $q$  in the tag IDs. Thus, all tags with

<sup>3</sup>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.)

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 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  $\log_B F$ ,  $2\log_B F$ ,  $3\log_B F$  and so on. A working example for the MAS scheme is shown in Figure 4. 2 reader messages and 5 tag responses are required.

## 4. Analysis and Numerical Evaluation

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 are then studied through numerical evaluation 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.

### 4.1 Energy Models

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<sup>4</sup> [4].

<sup>4</sup>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

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  $q(m) P_{Ttx} + u(m) P_{Trx}$ , where  $q(m)$  is the number of reader queries and  $u(m)$  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  $q(m) P_{Ttx} + u(m) P_{Trx} + z(m) P_{Trx}$ , where  $z(m)$  is the number of sleep commands issued by the reader.

### 4.2 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 which is easier to analyze than average energy per tag due to the collective nature of the tag arbitration process<sup>5</sup>. We will also derive expressions for the average number of time slots required to read all tags,  $\bar{t}(m)$ .

Due to space limitations, only the analysis for the MS scheme is presented here. The complete analysis for all three schemes can be found in [10].

#### 4.2.1 Analysis for MS Scheme

We begin by determining the average number of time slots required to read  $m$  tags,  $\bar{t}_{MS}(m)$ . The  $m$  tag IDs are uniformly distributed in the interval  $[0,1)$ . A probe at level  $L$  of the B-ary tree corresponds to a query over a subinterval of size  $B^{-L}$ . The probability that a ID falls within a particular subinterval of the above size is  $p = B^{-L}$  and the probability that it falls outside this subinterval is given by  $(1 - B^{-L})$ . Now the probability that  $k$  out of  $m$  tags fall within a particular interval of size  $B^{-L}$  (or analogously the probability of  $k$  of the  $m$  tags having the same prefix as the

when energy awareness is required, the mode of operation of RFID systems becomes important as well.

<sup>5</sup>The corresponding value per active tag can be found easily by dividing by the number of tags.

query of size  $L$  bits), is given by the binomial distribution

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

with  $p = B^{-L}$

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

Thus, the probability of no reply is

$$P(I) = P(0, m, L) = (1 - B^{-L})^m, \quad L > 0 \quad (2)$$

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

$$\begin{aligned} P(S) &= \sum_{k=1}^m P(k, m, L) P_{NC}(F, k) \\ &= \sum_{k=1}^{\min(m, F)} P(k, m, L) P_{NC}(F, k), \quad L > 0 \end{aligned} \quad (3)$$

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,

$$\begin{aligned} P(C) &= 1 - P(S) - P(I) = 1 - (1 - B^{-L})^m \\ &\quad - \sum_{k=1}^{\min(m, F)} P(k, m, L) P_{NC}(F, k), \quad L > 0 \end{aligned} \quad (5)$$

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

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 time slots used 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, 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 collision at node  $w$  at level  $L$  is the same as the probability of collision as given by Equation 5. i.e.

$$\begin{aligned} \chi_{w, L, m} &= \chi_{L, m} = 1 - (1 - B^{-L})^m \\ &\quad - \sum_{k=1}^{\min(m, F)} P(k, m, L) P_{NC}(F, k), \quad L > 0 \\ &= 1 - P_{NC}(F, m), \quad L = 0 \end{aligned} \quad (6)$$

Then the probability of a node being visited

$$V_{w, L, m} = V_{L, m} = \begin{cases} 1 & L = 0 \\ \chi_{L-1, m} & L > 0 \end{cases}$$

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  $w$  at all levels  $L$ . We assume an infinite tree where  $L = \infty$  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,

$$\begin{aligned} \bar{t}_{MS}(m) &= (F + 1) \sum_{L=0}^{\infty} \sum_{w=0}^{B^L-1} V_{w, L, m} \\ &= (F + 1) \left[ 1 + \sum_{L=1}^{\infty} B^L V_{L, m} \right] \\ &= (F + 1) \left[ 1 + \sum_{L=1}^{\infty} B^L \chi_{L-1, m} \right] \\ &= (F + 1) \left[ 1 + \sum_{L=0}^{\infty} B^{L+1} \chi_{L, m} \right] \end{aligned} \quad (7)$$

Using 6 gives

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

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,

$$\begin{aligned} P_{NC}(F, k) &= \frac{F}{F} \times \frac{F-1}{F} \times \dots \times \frac{F-k+1}{F} \\ &= \frac{F!}{F^k (F-k)!} \end{aligned} \quad (9)$$

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 [5] that the average time to read all tags with a binary search tree can be computed within a tolerance  $\epsilon$ , if

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

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  $\epsilon = 0.01$ .

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  $F + 1$ ,

$$\begin{aligned} \bar{q}_{MS}(m) &= 1 + B(1 - P_{NC}(F, m)) \\ &+ B \sum_{L=1}^{\infty} B^L \{1 - (1 - B^{-L})^m \\ &- \sum_{k=1}^{\min(m, F)} P(k, m, L) P_{NC}(F, k)\} \end{aligned} \quad (10)$$

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 a node  $w$  in the tree, the expected number of tags resolved or read is

$$\begin{aligned} \rho_{w,n,m} = \rho_{n,m} &= \sum_{k=1}^{\min(m, F)} k P(k, m, n) P_{NC}(F, k), \quad n > 0 \\ &= m \cdot P_{NC}(F, m), \quad n = 0 \end{aligned} \quad (11)$$

which corresponds to the tags read in collision free frames. Thus,

$$\begin{aligned} \bar{s}(m, n) &= \sum_{w=0}^{B^n - 1} \rho_{w,n,m} = B^n \rho_{n,m} \\ &= B^n \sum_{k=1}^{\min(m, F)} k P(k, m, n) P_{NC}(F, k), \quad n > 0 \\ &= m \cdot P_{NC}(F, m), \quad n = 0 \end{aligned} \quad (12)$$

This is because, by counting all the tags read at level  $n$ , we ensure we have counted all tags that will be read above level  $n$  as well. Any tag read at levels less than  $n$  will be read at

**Table 1. 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

level  $n$  also.

At the top level of root, 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

$$\bar{u}_{MS}(m) = m + \sum_{L=1}^{\infty} (m - \bar{s}(m, L - 1)) \quad (13)$$

where  $\bar{s}(m, L - 1)$  is the number of tags resolved up to and including level  $L - 1$ . Using Equation 12 gives with some manipulation of level indices,

$$\begin{aligned} \bar{u}_{MS}(m) &= m + m - m \cdot P_{NC}(F, m) \\ &+ \sum_{L=1}^{L'-1} [m - B^L \sum_{k=1}^{\min(m, F)} k P(k, m, L) P_{NC}(F, k)] \end{aligned} \quad (14)$$

where  $L'$  is the upper limit to  $L$  as introduced previously.

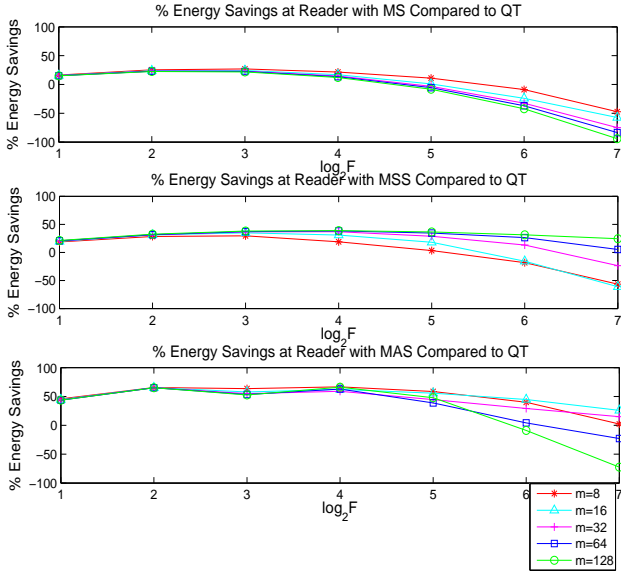
### 4.3 Numerical Evaluations

Here we numerically evaluate the analytical expressions derived before for average reader and total active tag energy consumption as well as the average 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. We look at values of  $m$  ranging from 8 to 128 and  $F$  from 1 to 128 as powers of  $B = 2$  to judge the effect of these two important parameters for our protocols<sup>6</sup>. The power consumption values used in the evaluation are shown in Table 1<sup>7</sup>. 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  $F$ .

<sup>6</sup>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.

<sup>7</sup>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.





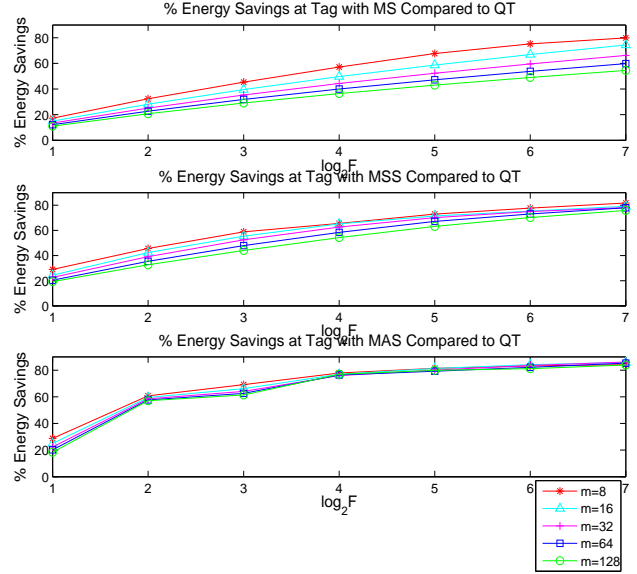
**Figure 5. Percentage energy savings at the reader for all three schemes compared to the QT protocol**

#### 4.3.1 Comparison of Energy Savings at the Reader

Figure 5 shows the plots for all three schemes. It is seen that for small values of  $F$  there is a reduction in energy consumption compared to the QT protocol. As the value of  $F$  increases, however, the energy due to receive slots begins to dominate resulting in increased energy consumption. The MSS scheme saves more energy than the MS scheme by reducing the number of collisions of tag responses. The fact that sleep commands are a drain on the reader’s energy is offset by the benefits of putting tags to sleep, primarily because each sleep commands uses up only one time slot. The MAS scheme achieves the highest energy savings even at small values of  $F$ . 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.

#### 4.3.2 Comparison of Total Energy Savings at Active Tags

The energy savings to read all active tags by the three schemes are shown in Figure 6. 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 great-



**Figure 6. Percentage energy savings at tags for all three schemes compared to the QT protocol**

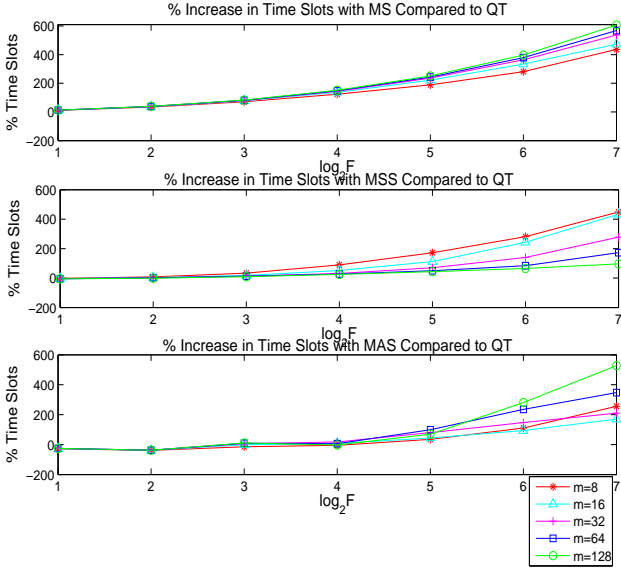
est energy savings, with a rapid increase in % savings with increasing  $F$ . The structured responses from tags to their assigned slots ensures that the reader knows which sub-tree to query based on collisions in slots, reducing the number of responses from tags. The MSS scheme does better than the MS scheme making use of sleep commands to reduce collisions.

#### 4.3.3 Comparison of Increase in Time Slots

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 7.

For the MS scheme, increasing  $F$  results in increased number of time slots required to read all tags, mainly because the reduction in collisions is not enough to offset the extra slots used. With the MSS scheme, increasing  $F$  does not necessarily imply that more time will be required to read all tags. The reduction in collisions in fact helps decrease the time slots (with % increase in time slots less than 0 in the figure) required for small values  $F$ . Increasing  $F$  above this value results in too many wasted slots. The MAS scheme also manages to decrease the amount of time (in fact greater than MSS as well) required to read all tags for small values of  $F$ .

The MS scheme takes the most number of time slots of



**Figure 7. Percentage increase in time slots for all three schemes compared to the QT protocol**

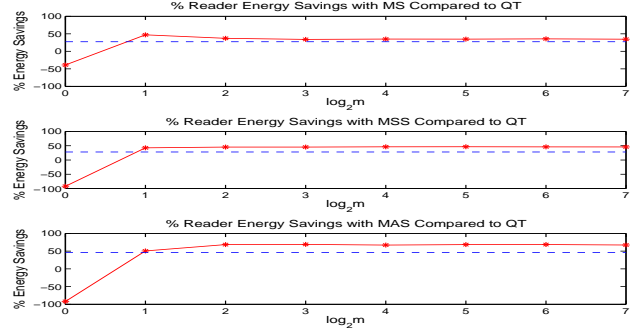
all three schemes and has the greatest rate of increase in time slots required. The MSS and MAS schemes have a slow rate of increase in time slots with increase of  $F$ . In fact, for  $F = 2, 4$  these protocols use fewer time slots than the QT protocol itself for all  $m \geq 2$  (values of  $m = 2, 4$  have not been shown in the plots in this section, but were considered as well). For large values of  $F$ , the MAS scheme uses more time slots than the MSS scheme.

#### 4.4 Discussion

Based on the analysis of the three schemes and comparisons based on their corresponding numerical evaluations, 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. The structured assignment of slots to specific sub-trees allows the reader to distinguish between the tags responsible for collisions in slots reducing the number of nodes visited in the tree leading to the reductions and slower increase in required time slots. Moreover, the MAS and the MSS schemes also reduce the time required to read all tags for small values of  $F$ .

### 5 Simulation Based Evaluation

This section aims to validate the analysis presented above by evaluating the schemes through simulation. Sim-



**Figure 8. Percentage energy savings achieved by all three schemes at the reader when reading different number of tags with tag upper bound  $M = 128$ . The dashed line shows the expected energy savings if encountering these tags randomly.**

ilar performance of the schemes by these experiments will reinforce the conclusions made above. The process of selecting an ideal value of  $F$  is then described with an evaluation of possible energy savings.

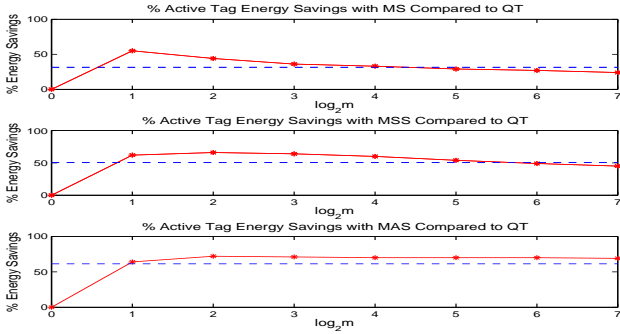
#### 5.1 Experiments and Results

We carried out Monte Carlo simulations of the four protocols; namely QT, MS, MSS and MAS and compared the latter three to the QT protocol. The size of a tag ID was set to be 64 bits and its value was picked randomly and uniformly from the total tag ID space thus created. Each data point shown in the plots is the average of 1000 runs. The power values used for the energy model are the ones previously used for the numerical evaluation as given in Table 1.

Due to space restrictions, we do not show the plots derived through simulation in this paper. The plots can be viewed by referring to [10]. The simulation plots are found to closely follow the patterns in the plots from the numerical evaluation. This validates our previous analysis and allows the use of the expressions obtained for predicting the performance of the protocol for any input parameters, and also selecting appropriate values of  $F$ .

#### 5.2 Selecting Appropriate Values of $F$

Based on the analysis presented above, we find that choosing an ideal value of  $F$  leads to greater energy savings. A larger value of  $F$  might increase energy savings but may not meet the time constraints imposed. A smaller value of  $F$  on the other hand might waste the opportunity



**Figure 9. Percentage energy savings achieved by all three schemes at active tags when reading different number of tags with tag upper bound  $M = 128$ . The dashed line shows the expected energy savings if encountering these tags randomly.**

to save energy. The analytical framework developed before can be used for determining this ideal value of  $F$ . Knowing the number of tags to read  $m$ , the reader could consult a table which lists the expected time it would take to read those tags for each value of  $F$ . Then it could select the value of  $F$  which meets the time constraint as well as provides the greatest energy savings at the reader. We will be concerned with minimizing the reader consumption in this work and will select  $F$  based on this goal since energy savings as a function of  $F$  differs for the reader and active tags. In different circumstances where active tag energy consumption is more important,  $F$  may be chosen accordingly (by choosing as large a value of  $F$  as possible based on our numerical evaluation of active tag energy savings).

This ideal value of  $F$  is, however, difficult to select in practice. A reader may be unsure of the number of tags it has to read. By selecting  $F$  assuming  $m$  to be small, it might exceed the time constraints which clearly is not acceptable. To ensure that regardless of the number of tags the time constraint is met, the reader could assume  $m$  to be a certain worst case upper bound on the number of tags that the reader will encounter. This upper bound value could be configured into the reader by its user based on his knowledge of the number of tags the reader could encounter. For example, in a supermarket, the maximum number of tags a reader could encounter on a shelf could be used. For predefined applications, this value could be set by the manufacturer itself. The closer the upper bound to the actual number of tags encountered, the better the energy savings.

To get a sense of energy savings by relying on only an upper bound, we show the results of an experiment with an upper bound of  $M = 128$  with the reader encountering

$m = 1, 2, 4, 8, 16, 32, 64$  and  $128$  tags. This should emulate the performance of these protocols when encountering a random number of tags, with a reasonably loose upper bound as well for some small values of  $m$ . The time constraint supplied was to read tags within an average time of 1 second. Each slot is assumed to be of duration 1 ms which corresponds to a data rate of 100 kbps with tag ID length up to 100 bits which are typical values. Thus, we consider the reader and tag energy savings with a 1000 time slot constraint.

Figure 8 shows the energy savings at the reader by selecting appropriate values of  $F$  as described above. The average savings for the MS, MSS and MAS schemes are 27%, 28% and 46% respectively for all values of  $m$  considered. This is a significant reduction considering that we just used an upper bound on  $m$ . One limitation that becomes apparent is that all three schemes use more energy than QT for  $m = 1$ , the case where no arbitration is actually required, which has resulted in decreasing the average reader energy savings. With  $m = 1$ , any value of  $F$  over 1 will waste energy with no benefits, and since we chose  $F$  based on an upper bound of  $m$  there is energy wastage. This gives us an insight into how we could overcome this by possibly querying the root node with  $F = 1$  and then letting the multi-slotted schemes take over by using  $F > 1$  for other nodes. Another approach would be to let the user switch to QT protocol mode when he is holding up an item to the reader or points the reader specifically at a tag (i.e. when the user knows  $m = 1$ ), and revert back to the energy-aware mode for all other times. This can be done with a simple switch or push-button on the reader. Another promising approach is to estimate the number of tags to be read first and then selecting  $F$  accordingly [7]. This approach, if reasonably accurate, would yield values of  $F$  that are close to optimal in which case the energy savings would increase significantly and follow the trends in Figure 5 instead. We leave the exploration of such approaches to increase energy savings as future work.

Figure 9 shows the energy savings for active tags, which is an important ‘byproduct’, while selecting a value of  $F$  that minimizes energy consumption at the reader using only an upper bound  $M$ . It is seen that the MS, MSS and MAS schemes provide energy savings of 32%, 50% and 61% respectively on average for all values of  $m$  considered. If the aim were to minimize the active tag energy consumption instead of that at the reader, the value of  $F$  could be chosen accordingly possibly leading to greater savings.

To summarize, even though the schemes just used an upper bound on the value of  $m$ , they were still able to select values of  $F$  that provided significant energy savings. If the value of  $m$  is known a priori or estimated reasonably accurately, the full potential of these protocols to reduce energy consumption can be realized. The MAS scheme turns out

to be the most efficient of the three protocols in trading off time for energy savings mainly due to its ability to save a lot of energy even with small values of  $F$  as noted before.

## 6 Conclusions and Future Work

The approach of using multiple slots per node of a B-ary search tree to reduce collisions among tag responses was presented to provide for energy-aware RFID tag arbitration by the reader. Three different variants of this approach were explored with the aim of finding the one which was most efficient in trading off time in exchange for reduced energy consumption. These protocols, like the existing Query Tree protocol, are memoryless requiring the tags to store no state of the arbitration process and offer guarantees on the time required to read all tags. An analytical framework was developed to predict the average case performance of these protocols for different input parameters. The numerical evaluation of this framework was further validated with the help of simulation. All three protocols were shown to reduce energy consumption at the reader as well as active tags.

In this work we explored the benefits of using a frame with multiple slots per node of the binary tree. The frame size  $F$ , however, was kept fixed at all nodes. As we query more levels, the number of tags responding to a prefix keeps decreasing. This creates an opening for designing a scheme that uses a decreasing frame size  $F$  as we descend the tree. If done carefully, this should preserve the energy savings at the reader and active tags, while at the same time eliminating the number of wasted time slots resulting in increased tag reading throughput and even more energy savings at the reader. Another area of work is studying the effect of tag IDs which are not uniformly distributed in the tag ID space. It has been shown previously that the QT protocol performs poorly in such scenarios [11]. The three protocols proposed above could be evaluated for this scenario as well. It is expected that the MAS scheme will suffer like the QT protocol because its steps are fully deterministic without any randomness in choosing slots by tags. The tags in MS and MAS schemes on the other hand choose slots among a frame randomly allowing tags with similar prefixes to be still read at higher levels of the binary tree. Finally, the effect of ensuring reliability of the tag identification process needs to be considered in terms of energy. The reader may need to issue more queries to achieve this goal, and how this would affect the performance of the three proposed protocols compared to the QT protocol needs to be studied.

## Acknowledgements

This work is supported in part by NSF grant CCF-0329794. We also thank the anonymous reviewers for giv-

ing valuable suggestions to further improve on this work.

## References

- [1] EPC Generation 1 Tag Data Standards Version 1.1 Rev. 1.27, EPC Global Inc. Standard Specification, May 2005.
- [2] MIT Auto-ID Center. Draft Protocol Specification for a 900 MHz Class of Radio Frequency Identification Tag, February 2003.
- [3] Dimitri Bertsekas and Robert Gallager. Data networks, 2nd edition, Prentice-Hall, Inc., 1992.
- [4] K. Finkenzeller. RFID Handbook, *John Wiley & Sons*, 1999.
- [5] D.R. Hush. A Comparison of Arbitration Algorithms for RFID. *Technical report, Micron Communications, Inc.*, Boise, ID, 1997.
- [6] D.R. Hush, C. Wood, Analysis of Tree Algorithms for RFID Arbitration, In *Proceedings of IEEE International Symposium on Information Theory* pg. 107, 1998.
- [7] M. Kodialam and T. Nandagopal. Fast and Reliable Estimation Schemes in RFID Systems, In *Proceedings of ACM MOBICOM*, Los Angeles, CA, Sep. 2006, pp. 322-333.
- [8] C. Law, K. Lee, and K. Y. Siu. Efficient memoryless protocol for tag identification, In *Proceedings of DIALM*, Boston, MA, Aug. 2000, pp. 75-84.
- [9] J. Myung and W. Lee. Adaptive Binary Splitting: A RFID Tag Collision Arbitration Protocol for Tag Identification, accepted for publication in *ACM MONET*, 2006.
- [10] V. Namboodiri and L. Gao. Energy-Aware Tag Anti-Collision Protocols for RFID Systems, *ECE Tech. Report*, 2006. Available at [www.people.umass.edu/vnambood/publications.html](http://www.people.umass.edu/vnambood/publications.html).
- [11] A. Sahoo, S. Iyer and N. Bhandari. Improving RFID System to Read Tags Efficiently, *KRSIT Technical Report*, IIT Bombay, June 2006.
- [12] F. Zhou, C. Chen, D. Jing, C. Huang and H. Min. Evaluating and optimizing power-consumption of anti-collision protocols for applications in RFID systems, *Auto-ID Labs Tech. Report AUTOIDLABS-WP-SWNET-014*.