

A Novel Anti-Collision Protocol for Energy Efficient Identification and Monitoring in RFID-Enhanced WSNs

Dheeraj K. Klair, Kwan-Wu Chin and Raad Raad

School of Electrical, Computer, and Telecommunications Engineering

University of Wollongong

Northfields Avenue, NSW, Australia

{dkk282, kwanwu, raad}@uow.edu.au

Abstract— This paper presents a dynamic framed slotted Aloha (DFSA) protocol that is energy efficient, and more importantly, is the first protocol capable of monitoring tags. Our protocol uses three separate frames: 1) reservation, 2) body, and 3) monitor. The reservation and body frame are used to identify tags, whereas the monitor frame is used to keep track of identified tags. We have performed extensive simulation studies on all three frames, and compared our protocol with existing framed Aloha protocols. From our results, we confirm that our protocol is suitable for use in RFID-enhanced Wireless Sensor Networks (WSNs).

I. INTRODUCTION

In recent years, radio frequency identification (RFID) has gained enormous publicity, especially when both Wal-Mart and the Department of Defense mandate their respective suppliers use RFID tags in an effort to cut logistical costs [22][23]. The key advantage of RFID is its ability to identify multiple objects wirelessly without direct line of sight. Moreover, the versatility of RFID tags in terms of their shapes, sizes, ranges, and types makes them far superior to conventional bar codes. In-Stat [12] predicts that more than 33 billion tags will be produced globally by 2010, which is 25 times the number of tags produced in the year 2005. This increasing interest in RFID technology has brought forward a number of promising research areas. One of which is to integrate RFID technology with a wireless sensor node to create RFID-enhanced WSNs.

To date, a number of organizations have started investigating RFID-enhanced WSNs. In [10], the authors are building an in-home elder healthcare system comprising of RFID-enhanced wireless sensor nodes that monitor patients' medication intake. In a similar work, Intel [13] has developed a system called *Caregiver's Assistant* to monitor an elder's activities. Specifically, their system collects events resulting from an elder touching RFID tagged items. In [8], NASA/JPL outlines a project called Sensor Webs. The goal is to develop a web of sensors for monitoring environment changes and take actions when events happen. Finally, BP Oil [24] is using RFID for location tracking and to sense a machine's working condition.

The aforementioned works have thus far only reported on system issues pertaining to the creation of RFID-enhanced WSNs. However, a key problem that remains open is energy efficient tag reading protocols. In our previous work [16], we showed that an RFID reader uses a significant amount of

energy when scanning RFID tags. We then analyzed twelve variants of Framed Slotted Aloha protocols to determine their energy consumption [18]. We found that putting a tag to sleep using a feature called muting after it is identified reduces collisions, and hence energy wastage. Even more energy can be saved if muting is used in conjunction with the early-end feature, where idle slots are closed early. Lastly, we found that protocols which vary their frame size in accordance with the number of tags have the lowest energy consumption compared to those that use a fixed frame size.

A key observation from our studies is that existing anti-collision protocols consume a significant amount of energy during identification. Critically, they cannot be used to monitor tags in an energy efficient manner. Thus, making them unsuitable for use in energy constrained RFID-enhanced nodes. In this respect, no work has specifically addressed energy efficient monitoring in RFID-enhanced WSNs. The closest work to ours is [28], where the authors use reservation slots to resolve contention. Their algorithm, called detection and jump (DJ), involves the use of a detection frame, where tags contend with each other in 16-bit slots to transmit a randomly generated bit string. The main limitation of their protocol is that it needs to re-identify all tags whenever there is a change in tag numbers, which leads to inefficient use of energy.

In this paper, we propose an energy efficient dynamic framed slotted Aloha (DFSA) protocol that is suitable for RFID-enhanced WSNs. More importantly, it allows a reader to monitor tags efficiently. Our protocol uses three frames to identify and monitor tags. Moreover, it uses an estimation function that has been specifically developed for muting based RFID systems and promises accurate estimates [5][17]. In addition, we use optimal frame sizes proposed by Vogt [27] to ensure the correct frame size is used to read a given tag population.

This paper is organized as follows. In Section II, we introduce RFID systems. Following that, in Section III, we describe the proposed protocol, including the frames that it uses and how they are used to read and monitor tags. After that, we present our research methodology and simulation results in Sections IV and V respectively. Section VI concludes the paper.

II. BACKGROUND

RFID, or Radio Frequency IDentification, is a system for tracking and identifying multiple objects in a reader's interrogation zone simultaneously via magnetic or electromagnetic response exchange. It consists of an RFID reader and a finite number of tags; see Figure 1. RFID tags contain tiny integrated circuits with a small antenna for communicating with a RFID reader as well as the ability to store identification information. RFID tags can be active passive or semi-passive. Passive tags have no power source and on-tag transmitter. They use the power emitted from the reader to energize and transmit their identification codes (ID) to the reader. On the other hand, semi-passive and active tags have an on-board power source, and are activated by a reader's field. Active tags, however, do not require the reader to be present to operate and has an on-board transmitter for sending data or identifier (ID) [9]. Passive and semi-passive tags are cheap compared to active tags [9], and are therefore suitable for large scale deployments.

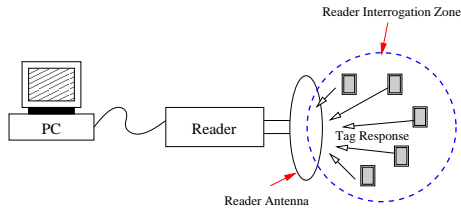


Fig. 1. Reader and tags interactions.

Tag collisions arise when multiple tags respond simultaneously to a reader's request, see Figure 1. This results in collisions at the reader, leading to bandwidth and energy wastage, and prolonged tag identification time [1] [9]. To date, numerous anti-collision protocols have been devised to resolve and avoid collisions. In particular, those based on TDMA are the most popular [1]. They can be classified into tree based and Aloha based protocols [25].

Tree algorithms are mostly deterministic in nature and form a binary tree like structure during identification. There are several variants of tree algorithms. Namely, tree splitting, query tree, binary search, bitwise arbitration and polling [11][3][4][14][2][7]. On the other hand, Aloha based anti-collision protocols are probabilistic in nature. Example protocols include Pure, Slotted and Framed Slotted Aloha (FSA), and its variants [1][29][21][25][9]. In Pure and Slotted Aloha, a tag responds after a random delay until it is identified. In Slotted Aloha, a tag replies in synchronized slots. In FSA and its variants, a tag replies once in a frame of fixed or varying sizes. Tags select a slot in a frame randomly and replies. If there is a collision, tags will defer to the next frame. All Aloha based protocols can operate with or without muting; i.e., when muting is enabled, a tag is silenced after identification.

Tree protocols are complex, incur significant memory overheads, and require complex hardware. This is in contrast to Aloha protocols, which have simpler reader designs, lower protocol complexity and bandwidth requirements, smaller number of reader to tag commands, and are able to dynamically adapt to varying number of tags [9][25]. All these properties argue in favor of using Aloha based protocols in

RFID-enhanced WSNs. To this end, we propose an energy efficient FSA protocol that not only identifies tags quickly but also supports monitoring.

III. PROPOSED PROTOCOL

The proposed protocol has three frames: reservation (R_{Frame}), body (B_{Frame}), and monitor (M_{Frame}). We first define each frame before providing an example that illustrates how they are used to read and monitor tags.

A. Frames

1) *Reservation*: R_{Frame} is used to allocate each tag a unique slot in the forthcoming B_{Frame} , where tags transmit their ID to the reader. Figure 2 depicts the R_{Frame} structure. After energizing tags, the reader transmits a *reset and calibration* command. Following that, a *read* command is transmitted, which specifies the number of reservation slots N_i to follow. We restrict N_i to values that are powers of two, and a maximum value of 256 [27]. The reader then transmits a *Null* command, which signals the start and end of reservation slots. If a single response is received in a slot, the reader transmits an acknowledgement (ACK); otherwise a negative ACK (NACK) is transmitted. The start and end of a slot is controlled by ACK/NACK. This allows the reader to vary a slot's duration depending on whether a slot is collision free. Specifically, if a slot is collision free, then the slot includes the transmission of an A_{Slot} command followed by an ACK command. Otherwise, the reader ends a slot with a NACK command after detecting a collision or an idle slot.

Each tag selects a slot, $S_i \in \{1, N_i\}$, randomly, and transmits a randomly generated bit string that is R bits in length, where $length(ID) > length(R)$. If the reader successfully receives R , the reader allocates a B_{Frame} slot to the responding tag via an A_{Slot} command. A tag then stores the allocated slot in its unique frame slot counter (UFSC) before entering mute state.

The A_{Slot} command is 9 bits in length, thereby making available 512 slots in the B_{Frame} . After transmitting an A_{Slot} command, the reader increments its body frame slot counter (BFSC); a counter that tracks the last allocated slot in the upcoming B_{Frame} .

Tags' transmissions are Manchester encoded to facilitate collision detection. In Manchester encoding, each bit is transmitted in a fixed duration called a bit period. Each encoded bit contains a transition at the midpoint of the bit period. The direction of a transition determines whether the received bit is a 0 or 1. A '0' is expressed by a low to high transition, and a '1' by high to low transition. Figure 3 shows a Manchester coded bit string transmitted by tags A and B simultaneously. Also shown is the resulting signal from these two transmissions. As can be seen, bits 2 and 3 cause no change in the received signal level, meaning a collision has occurred.

The reader relies on a tag estimation function to adjust the R_{frame} size used in every round. Recall that our protocol uses muting, hence we require estimation functions that take muting into account. Fortunately, Cha et al. [5] has proposed such a function, where an estimate is obtained by computing

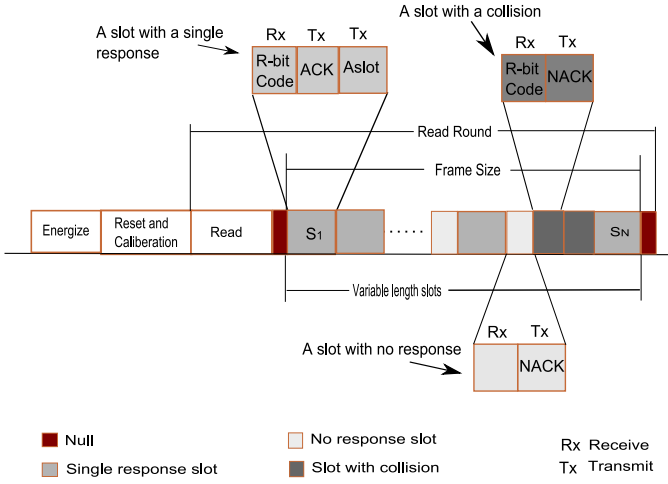


Fig. 2. Reservation frame structure.

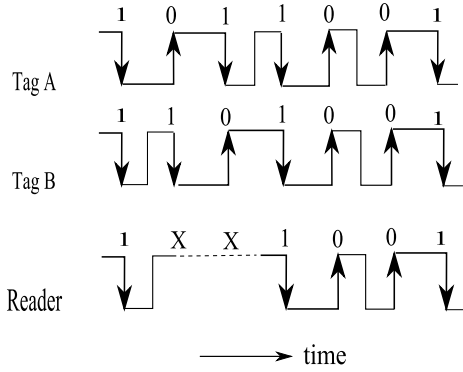


Fig. 3. Collision detection using Manchester encoding.

the ratio of the number of slots with collisions and the frame size, and is given by,

$$C_{ratio} = 1 - \left(1 - \frac{1}{N}\right)^n \left(1 + \frac{n}{N-1}\right) \quad (1)$$

where n is the tag to be estimated. C_{ratio} is computed after a read round as $C_{ratio} = \frac{c_k}{N}$ where c_k denotes the number of slots with a collision. Once a tag estimate is obtained, the reader updates its frame size according to Table I [27].

TABLE I
OPTIMAL FRAME SIZES FOR A GIVEN TAG RANGE.

FrameSize(N)	Tags range (n)
16	1 – 9
32	10 – 27
64	17 – 56
128	51 – 129
256	112 – ∞

2) *Body*: A B_{Frame} is transmitted after the R_{Frame} when $BFSC > 1$, and no collisions have been detected in the R_{Frame} , thereby indicating that all tags have been allocated a B_{Frame} slot and are muted.

Figure 4 depicts the structure of a B_{Frame} . The reader transmits an *unmute* command to activate all tags. After that, the reader transmits a synch pulse to synchronize tag responses. Following that, the reader transmits a body frame

(BF) command to inform tags to respond with their ID. The reader then transmits a slot offset (SO) command, which is used to skip tags identified in previous B_{Frame} rounds.

SO is initially set to zero. On reception of a SO command, tags with $UFSC \leq SO$ are muted, which corresponds to those tags identified in the last B_{Frame} . On the other hand, tags with $UFSC > SO$ remain active, and respond according to Algorithm 1. Once B_{Frame} ends, the reader sets $SO = BFSC - 1$.

```

1 if ( $SO == 0$ ) then
2   Tag transmits if the current reader slot is equal to tag's UFSC ;
3 else
4    $UFSC_{tmp} = UFSC$ ;
5   Tag transmits if the current reader slot is equal to
   ( $UFSC_{tmp} - SO$ ) ;
6 end

```

Algorithm 1: Pseudo code to avoid re-identification of tags.

The size of B_{Frame} equals $BFSC - 1$ when $SO = 0$, and $BFSC - 1 - SO > 0$. We limit the number of slots to 512, which is twice the maximum frame size available in R_{Frame} . However, in practice, a higher number of slots can be used if the reader has sufficient resources to manage more than 512 tags.

The reader transmits an *ACK* when an ID is received successfully and a *NACK* when it experiences collisions. Collisions occur when two or more tags transmit the same random bit string R simultaneously in a R_{Frame} , thereby causing the reader to assign them the same UFSC. Therefore, the reader must update their UFSC to resolve future collisions. To do this, each tag maintains a next frame slot counter (NFSC) that operates as follows.

On reception of a BF command, each tag initializes its NFSC to UFSC. Each tag then transmits in the $UFSC^{th}$ slot. For every slot with a single tag response, the reader transmits an *ACK*. On reception of an *ACK*, a tag first sets its UFSC to NFSC before going to sleep. On the other hand, for every slot with a collision, the reader transmits a *NACK*, which causes each unread tag to decrement its NFSC by one, and collided tags to set their UFSC and NFSC to zero.

Table II illustrates how slots with collisions are removed from a B_{Frame} . In round 1, tags C and D's transmission resulted in a collision, causing the reader to transmit a *NACK*. On reception of a *NACK*, the tags operate as per Algorithm 2. That is, tags C and D set their UFSC and NFSC to zero, whereas tag E decrements its NFSC by 1. Setting C and D's UFSC to zero effectively barred them from participating in future B_{Frame} and M_{Frame} because slots always begin from one. Hence, tags C and D will have to contend again in the next R_{Frame} to gain a unique slot in order to transmit in the following B_{Frame} .

3) *Monitor*: The reader uses M_{Frame} to monitor tags. Note that M_{Frame} is collision free, given that every tag has a unique transmission slot.

Figure 5 depicts the M_{Frame} structure. The reader first energizes tags before transmitting a *Sync* pulse to synchronize tag responses. After that, the reader sends a monitor frame

TABLE II
REMOVAL OF COLLIDED SLOTS FROM B_{Frame} .

	Tags	Response slot (UFSC)	Reader's slot	Reader's response	NFSC before ACK/NACK	Update NFSC using Algorithm 2	Update UFSC using Algorithm 2
Body Frame Round 1	Tag A	1	1	ACK	1	1 (Unchanged)*	1
	Tag B	2	2	ACK	2	2 (Unchanged)*	2
	Tag C & D	3	3	NACK	3	0 (Changed)**(***)	0
	Tag E	4	4	ACK	4	3 (Changed)	3
<p>* Tag A or B updates its UFSC to NFSC because its UFSC equals the current reader slot, and it received an ACK (See Algorithm 2). ** Tag C and D set their UFSC and NFSC to zero because their UFSC matches the current reader slot, and they received a NACK (See Algorithm 2). *** Tag E decrements its NFSC (4-1=3) because it is not muted, its UFSC is not equal to the current reader's slot, and it received a NACK (See Algorithm 2).</p>							

```

1 if  $UFSC == Current_{ReaderSlot} \& NACK$  then
2    $NFSC = 0$ ;
3    $UFSC = 0$ ;
4 else if  $UFSC == Current_{ReaderSlot} \& ACK$  then
5    $UFSC = NFSC$ ;
6 else if  $((UFSC \neq Current_{ReaderSlot}) \& (NACK) \& (NotMuted))$  then
7    $NFSC = NFSC - 1$ ;
8 else
9   No Change;
10 end

```

Algorithm 2: Pseudo code used by a tag to remove collided slots in a B_{Frame} .

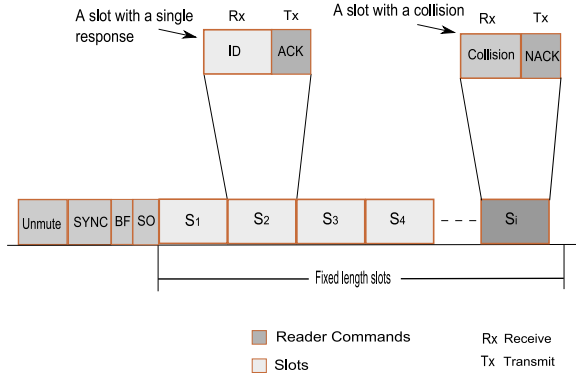


Fig. 4. Body frame structure. $S_i = BFSC - 1$ when $SO = 0$. $S_i = BFSC - 1 - SO$ when $SO > 0$.

(MF) command to mark the start of the monitoring phase, with the number of slots set to $BFSC - 1$.

Each tag responds with the same predefined 4-bit Manchester coded bit string. If the bit string is received successfully, the reader transmits an ACK . In addition to acknowledging the tag, the ACK also has the effect of muting the tag. If a tag has departed from the reader's interrogation zone, the reader will receive no response, and hence experiences an idle slot. In this case, the reader transmits a $NACK$. Both ACK and $NACK$ assist the reader in removing idle slots, as elaborated in Section III-D.

B. Protocol Operation

We now show how the aforementioned frames are used to identify tags. Let's say there are four tags A, B, C, and D. During initialization, the reader initializes $BFSC$ to one, and

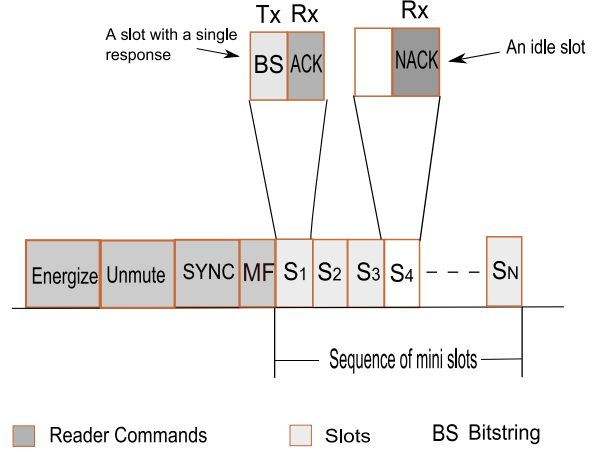


Fig. 5. Monitor frame structure.

tags set their $UFSC$ to zero. The reader begins by transmitting a R_{Frame} . Assume that the reader identifies these tags in the following order: A, B, C, and D. Hence, tag A will be assigned a $UFSC$ value of one via the A_{Slot} command, tag B will have a value of two, and so forth. After each tag is identified, the reader increases its $BFSC$ by one. Thus, at the end of the R_{Frame} , $BFSC$ will have a value of five.

The reader then starts a B_{Frame} , given that tags were detected in R_{Frame} with a SO value of zero. The number of slots in the B_{Frame} is $BFSC - 1$, which equals four. Tags then respond according to Algorithm 1, and upon conclusion, the reader sets the value of SO to $BFSC - 1$. If there are collisions, tags use Algorithm 2 to update their $UFSC$. The reader then keeps track of these collided slots and subtracts the number of slots with collision from the $BFSC$ at the end of the frame.

Lastly, the reader transmits M_{Frame} with $BFSC - 1$ slots. As specified in Section III-A.3, all tags respond with the same 4-bit bit string in their allocated slots. Upon receiving a reply, the reader notes that a tag is still present. Otherwise, if an idle slot is encountered, the protocol initiates the idle slot removal process outlined in Section III-D.

C. New Tags

Assume there are four new tags in the reader's interrogation zone: D, E, F, and G. After some time, the reader transmits a R_{Frame} and the slot allocation begins from the current $BFSC$ value, i.e., 5. Let's assume the tags are identified in the following sequence: D, E, F, and G. Thus, the $UFSC$ of

these tags is 5, 6, 7, and 8 respectively, and the reader would have a BFSC value of 9.

To read these new tags, the reader transmits a B_{Frame} . Recall that tags A, B, C, and D have already been identified. Hence, we will need to omit them. To do this, the reader transmits a SO command with a value of 4. Then, based on Algorithm 1, tags D, E, F, and G, transmit in the $(UFSC_{tmp} - SO)^{th}$ slot, i.e., slots 1, 2, 3, and 4 respectively. Note, the UFSC value for tags D, E, F, and G remains unchanged. Hence, at the end of this process, tags A to G have a UFSC value of 1 to 9 respectively.

D. Departing Tags

Tags may leave a reader's interrogation zone, thereby causing idle slots in M_{Frame} , and unnecessarily prolonging the monitoring process. To remove these idle slots, a tag uses its next frame slot counter (NFSC).

On reception of the MF command, each tag initializes its NFSC to UFSC. Each tag then transmits in the $UFSC^{th}$ slot. For every idle slot, as indicated by a $NACK$, each unread tag decrements its NFSC by one. On the other hand, if a single response is received, the reader transmits an ACK . Those tags with UFSC that matches the current reader slot and received an ACK set their UFSC to NFSC before going into mute state.

Table III illustrates how idle slots are removed. In monitoring round 1, no tag has departed. Thus, the NFSC and UFSC value of tags remains unchanged. In monitoring round 2, Tag-C departs. This causes slot-3 to be idle, thereby, causing the reader to transmit a $NACK$. Since tags A and B have been muted, the $NACK$ does not affect their UFSC. Tag-D, however, will decrement its UFSC by one. As a result, in monitoring round 3, tag D responds in slot 3 instead of slot 4.

The reader uses an idle slot counter (IDSC) to record the number of idle slots appearing in its M_{Frame} . After removing all idle slots, the number of slots in a M_{Frame} is computed as $BFSC = BFSC_{Current} - 1 - IDSC$, where $BFSC_{Current}$ is the value of BFSC at the beginning of a M_{Frame} .

IV. SIMULATION METHODOLOGY

To study the proposed protocol, we wrote a simulator using Matlab 7.0.4. The system consists of an RFID-enhanced sensor node with n tags in its interrogation zone. We assume each node is equipped with a SkyTek M1-Mini RFID reader [26]. The node operates from a Lithium rechargeable battery (B) that has 480 joules of energy. The tag to reader data rate is 26 kbps, as per ISO 15693 [26]. The power consumed by a RFID reader for scanning and sleeping is 180 milli-watts and 150 micro-watts respectively.

In our simulation, we omit propagation and processing delay. Further, our simulation considers a noise free channel, i.e., packet losses are due to collisions only. Finally, tag ID is 96 bits in size.

We assume tags are passive, have no power source, static, and are used in read-only mode. Further, we assume tags' antenna is never at 90 degrees with respect to the reader. Otherwise, tags become unreadable, and hence they do not contribute to the offered load. In other words, a reader is

unaware of tags that are displaced by 90 degrees since they are not energized to participate in any communications [20][9].

The proposed protocol involves eleven different commands, see Table IV. The duration of the *Reset and Energize* command is hardware dependent. In our simulation, we set it to 1 milliseconds according to [15]. Table IV also defines the duration of slots that appear in R_{frame} , B_{frame} and M_{frame} .

TABLE IV
SIMULATION PARAMETERS.

Parameter	Length (Bits)	Duration(milliseconds) (Length/26kbps)
Reset and Energize	-	1.00
Sync	9	0.35
Read	13	0.50
BF	4	0.15
MF	4	0.15
A_{slot}	9	0.35
ACK	6	0.23
NACK	6	0.23
Null	3	0.12
Unmute	6	0.23
SO	9	0.35
Reservation slot	10	0.38
Body frame slot	96	3.69
Monitor frame slot	4	0.15

We record the average delay experienced by the reader during the identification and monitoring phase. The identification phase comprises of a reservation followed by a body frame, whereas the monitoring phase only involves periodic transmissions of the monitor frame.

In the identification phase, we compute the 1) average delay incurred to read a given number of tags, 2) the average delay due to collisions, and 3) the average delay due to idle slots. In the monitoring phase, we compute the time it takes to monitor a given tag set when a fixed number of tags depart from the reader's interrogation zone.

Once we have the delays, the energy consumed can be computed by multiplying the delay with the power consumed during scanning. The total energy consumed by the reader to identify and monitor a given number of tags is then obtained by adding the energy consumption in the identification and monitoring phases along with the energy consumed due to sleeping.

To analyze the overall protocol, we vary the frequency in which the reservation and monitor frame are transmitted. We tested three different settings for both frames, see Section V-C for details.

In our simulation studies, we compare the proposed protocol with three framed slotted Aloha (FSA) protocols, namely Basic FSA (BFSA)[18][29], Dynamic FSA (DFSA) [18][5][6], and Enhanced DFSA (EDFSA) [18][19]. Moreover, for all protocols, we consider the following two features:

- 1) Muting, where a tag is muted after it is identified.
- 2) Muting and early end, where a protocol combines both muting and early end; the later feature allows a reader to close an idle slot early

V. RESULTS

In this section, we compare the energy consumption of our protocol with six FSA variants using the simulation settings

TABLE III
DEPARTING TAGS EXAMPLE.

	Tags	Response slot UFSC	NFSC before ACK/NACK	Reader's Resposne	NFSC after ACK/NACK	Assign NFSC to UFSC after NACK/ACK
Monitoring Round 1	Tag A	1	1	ACK	1 (Unchanged)	1 (Unchanged)
	Tag B	2	2	ACK	2 (Unchanged)	2 (Unchanged)
	Tag C	3	3	ACK	3 (Unchanged)	3 (Unchanged)
	Tag D	4	4	ACK	4 (Unchanged)	4 (Unchanged)
Monitoring Round 2 (Tag C departs)	Tag A	1	1	ACK	1 (Unchanged)	1 (Unchanged)
	Tag B	2	2	ACK	2 (Unchanged)	2 (Unchanged)
	-	-	-	NACK	-	-
	Tag D	4	4	ACK	3 (Changed)	3 (Changed)
Monitoring Round 3 (Idle slot removed)	Tag A	1	1	ACK	1 (Unchanged)	1 (Unchanged)
	Tag B	2	2	ACK	2 (Unchanged)	2 (Unchanged)
	Tag D	3	3	ACK	3 (Unchanged)	3 (Unchanged)

presented in Section IV. Firstly, we present results concerning the identification phase, followed by the energy consumption incurred in the monitoring phase when a fixed number of tags depart from the reader's interrogation zone. Lastly, we present results when the number of arriving and departing tags is varied.

A. Identification Phase

Figure 6 presents the energy wasted from collisions. Our protocol, labeled as 'ReserveMonitor', consumes the lowest energy compared to FSA variants. The key reason for this is the use of 0.38 ms or 10-bit reservation slots, compared to FSA variants that have a slot of duration 4.3 ms [18]. This means, every collision in FSA variants is approximately 10 times longer, thereby consumes ten times more energy. As the number of tags increases, the energy consumption of BFSa variants increases exponentially. This is because they use a fixed frame size, which has a high collision probability when the number of tags exceeds the frame size. Lastly, DFSA and EDFSA have lower energy wastage from collisions compared to BFSa variants because they use varying frame sizes to reduce the probability of collisions. Nevertheless, they consume more energy than ReserveMonitor, due to their longer slot duration.

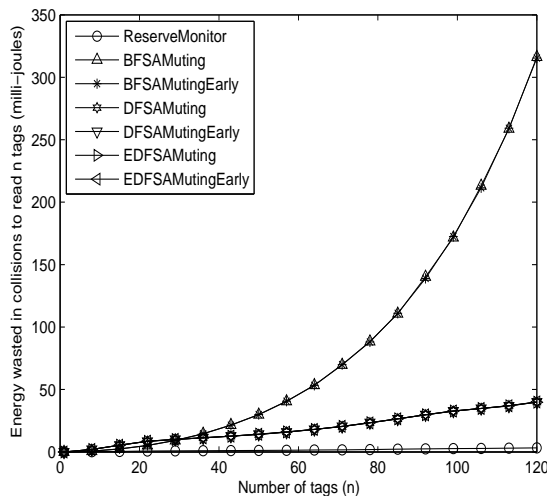


Fig. 6. Energy wasted in collisions during the identification phase.

Figure 7 presents the energy consumed by idle slots. Here, BFSAMutingEarly wastes the least amount of energy. This is because of two reasons. Firstly, the early-end feature results in a significant reduction in energy wastage associated with idle listening. Secondly, because of the use of a fixed frame, the probability of having an idle slot reduces as the number of tags increases. ReserveMonitor has a slightly higher energy consumption than BFSAMonitorEarly because it is difficult to set a frame size that reduces both collision and idle slots simultaneously. Lastly, DFSAMuting and EDFSAMuting have the highest energy wastage due to idle listening because each idle slot is 4.3 ms in length [18].

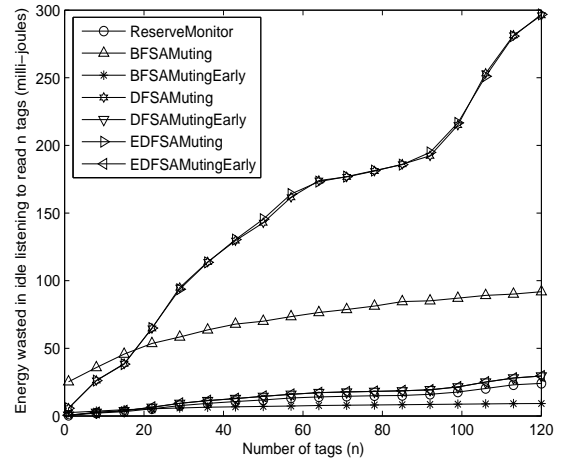


Fig. 7. Energy consumed to read n tags during the identification phase.

Figure 8 shows the energy consumption to read a given number of tags successfully. ReserveMonitor consumes the lowest energy compared to FSA variants. This is because of the use of small reservation slots, which reduces the energy wastage due to idle listening as well as collisions.

B. Monitoring Phase

Figure 9 compares the energy consumption when FSA variants are used for monitoring. The energy consumed by ReserveMonitor is significantly less than FSA protocols. This is because each identified tag has a monitor frame with a unique 1.5 ms response slot, where they transmit a predefined

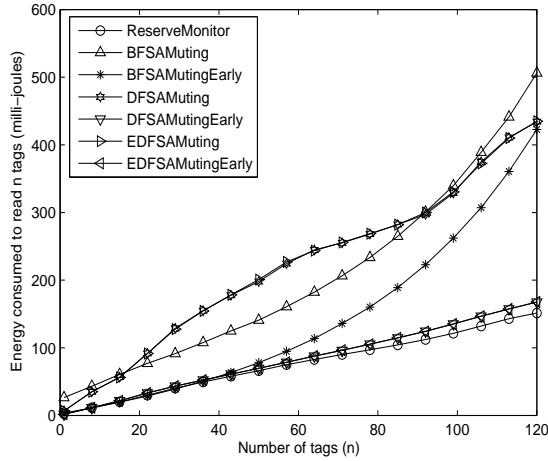


Fig. 8. Energy consumed to read n tags during the identification phase.

4-bit bit-string. On the other hand, FSA variants have to re-identify all tags to determine whether certain tags have been removed from the reader's interrogation zone. The energy consumption of each protocol reduces as tags are removed after every monitor round because there are less contention, hence tags can be identified quicker.

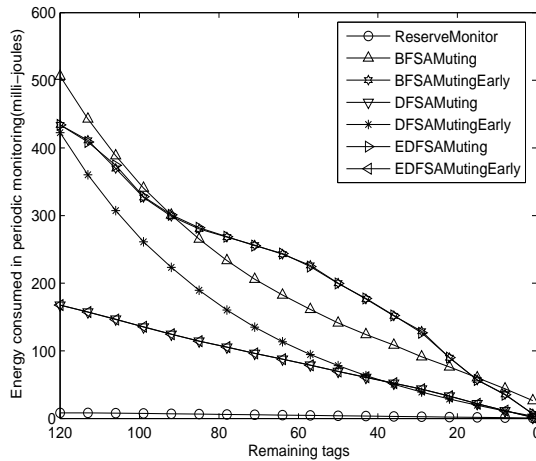


Fig. 9. Energy consumed while monitoring a given tag set.

C. Realistic Scenarios

We now compute the energy consumption and battery lifetime of the reader when it is used to monitor a given region where tags arrive and depart randomly. Three experiments are performed where we vary the frequency of identification and monitoring rounds. In the first scenario, an identification round and a monitor round is performed daily, with half a day sleep time between them. In the second scenario, both identification and monitoring rounds are transmitted hourly with a half an hour gap between them. Lastly, both frames are transmitted half hourly with a 15 minutes gap. Initially, there are 10 tags in the reader's interrogation zone. A random number of tags

depart from the reader's interrogation zone before the arrival of the monitor frame. Once a monitor frame finishes, more random number of tags depart from the reader's interrogation zone. The simulation ends when the reader finishes its battery.

From Figure 10(a), if the reservation frame and monitor frame are transmitted daily, the reader has a lifetime of 5.1 years. On the other hand, if the reservation frame and monitor frame are transmitted with an hourly sleep period, then the reader finishes its battery in 8.4 months, as shown in Figure 10(b). Lastly, as shown in Figure 10(c), if the respective frames are transmitted every half hour, the reader is only able to operate for 138 days or 4.6 months. Table V summarizes these results; on average, for each scenario, the number of tags is approximately 200.

TABLE V
SUMMARY OF RESULTS INVOLVING REALISTIC SCENARIOS.

Figure	Reservation Frame Frequency	Monitor Frame Frequency	Mean number of tags	Battery Lifetime
10(a)	1 Day	1 Day	199.35	5.1 Years
10(b)	1 Hour	1 Hour	202.75	8.4 Months
10(c)	30 Minutes	30 Minutes	201.68	138 Days

From above results, it is clear that the frequency of reservation and monitor frames significantly affect the reader's energy consumption. Hence, applications need to ensure the correct frequency is used to track tagged items; i.e., one that balances energy usage and application requirements.

VI. CONCLUSION

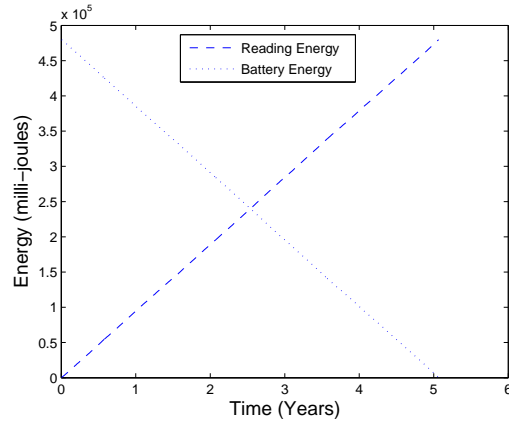
In this paper, we have proposed a DFSA based protocol that uses a reservation frame for slot allocation, a body frame for tag identification, and a monitor frame to monitor RFID tags. Our simulation results show that our protocol is significantly more energy efficient than existing FSA protocols, primarily because it uses small reservation and monitor slots, and it removes idle slots quickly. Moreover, it resolves collisions promptly. Thereby, making our protocol suitable for use in RFID-enhanced WSNs.

VII. ACKNOWLEDGMENT

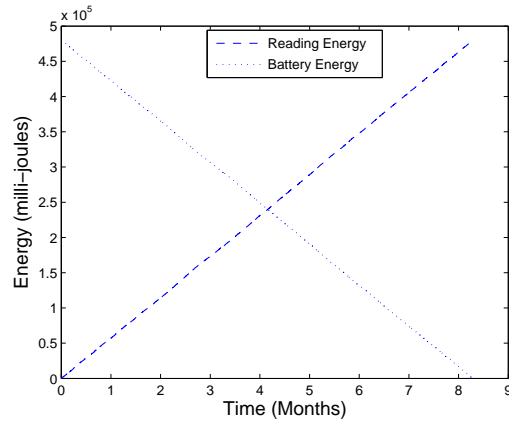
We acknowledge and thank the support of the Australia Research Council, grant number DP0559769.

REFERENCES

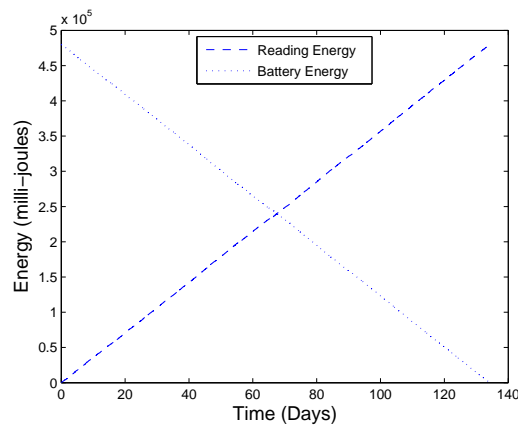
- [1] C. Abraham, V. Ahuja, A. Ghosh, and P. Pakanati. Inventory management using passive RFID tags: A survey. <http://www.cs.rutgers.edu/badri/553dir/papers/MCPaperFinal-coldetect.pdf>.
- [2] F. Bo, L. Jin-Tao, G. Jun-Bo, and D. Zhen-Hua. ID-binary tree stack anticollision algorithm for RFID. In *Proceedings of the 11th IEEE Symposium on Computers and Communications*, pages 207–212, Pula-Cagliari, Sardinia, Italy, June 2006.
- [3] L. Bolotnyy and G. Robins. Randomized pseudo-random function tree walking algorithm for secure radio-frequency identification. In *Fourth IEEE Workshop on Automatic Identification Advanced Technologies*, pages 43–48, Buffalo, NY, USA, Oct 2005.
- [4] M. A. Bonuccelli, F. Lonetti, and F. Martelli. Randomized hashing for tag identification in RFID networks. In *ERCIM Technical Report: Computer Communications and Networks*, Oct 2005.



(a) Frequency of identification and monitoring round = One Day



(b) Frequency of identification and monitoring round = One Hour



(c) Frequency of identification and monitoring round = Half Hour

Fig. 10. Energy consumption with with varying transmission frequency of identification and monitoring round.

- [5] J.-R. Cha and J.-H. Kim. Novel Anti-collision Algorithms for Fast Object Identification in RFID System. In *The 11th Intl. Conf. on Parallel and Distributed Systems*, pages 63–67, Fukuoka, Japan, July 2005.
- [6] J.-R. Cha and J.-H. Kim. Dynamic framed slotted aloha algorithms using fast tag estimation method for RFID system. In *The 3rd IEEE Consumer Communications and Networking Conference*, pages 768–772, Las Vegas, Nevada, USA, Jan 2006.
- [7] H.-S. Choi, J.-R. Cha, and J.-H. Kim. Fast wireless anti-collision algorithm in ubiquitous ID system. In *IEEE Vehicular Technology Conference*, pages 4589–4592, Los Angeles, CA, 2004.
- [8] J. Collins. NASA creates thinking RF sensors. http://www.sensorwaresystems.com/historical/press/RFIDJrnl_NASA_Oct04.pdf.
- [9] K. Finkenzeller. *RFID Handbook, Fundamentals and Apps. in Contactless Smart Cards and Identification*. John Wiley and Sons Ltd., 2003.
- [10] L. Ho, M. Moh, Z. Walker, T. Hamada, and C.-F. Su. A prototype on RFID and sensor networks for elder healthcare: Progress report. In *Proceeding of the 2005 ACM SIGCOMM Workshop on Experimental Approaches to Wireless Network Design and Analysis*, pages 70–75, Pennsylvania, USA, 2005.
- [11] D. R. Hush and C. Wood. Analysis of tree algorithms for RFID arbitration. In *The IEEE Intl. Symposium on Information Theory*, pages 107–107, Cambridge, MA, USA, Aug 1998.
- [12] In-Stat. Explosive growth projected in next five years for RFID tags. <http://www.instat.com>.
- [13] Intel. Intel sensor nets/RFID research areas. Webpage. http://www.intel.com/research/exploratory/wireless_sensors.htm.
- [14] C. J. Tree algorithms for packet broadcast channels. *IEEE Transactions on Information Theory*, 25(5):505–515, 1979.
- [15] G. Khandelwal, K. Lee, A. Yener, and S. Serbetli. ASAP: a MAC protocol for dense and time-constrained RFID systems. *EURASIP Journal on Wireless Communications and Networking*, 2:13, 2007.
- [16] 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 *IEEE WoWMoM*, Helsinki, Finland, June 2007.
- [17] D. K. Klair, K.-W. Chin, and R. Raad. On the Accuracy of Tag Estimation Functions. In *The 6th IEEE International Symposium on Communications and Information Technologies (IEEE ISCIT07)*, Sydney, Australia, Oct 2007.
- [18] D. K. Klair, K.-W. Chin, and R. Raad. On the Suitability of Framed Aloha Based RFID Anti-Collision Protocols for RFID-Enhanced WSNs. In *The 16th IEEE International Conference on Computer Communications and Networks (IEEE ICCCN'2007)*, Honolulu, Hawaii, Aug 2007.
- [19] S.-R. Lee, S.-D. Joo, and C.-W. Lee. An enhanced dynamic framed slotted aloha algorithm for RFID tag identification. In *The 2nd Intl. Annual Conference on Mobile and Ubiquitous Systems: Networking and Services*, pages 166–172, San Diego, CA, USA, July 2005.
- [20] S. Lewis. Think RFID Technology. <http://hosteddocs.ittoolbox.com/laran032604.pdf>, 2004.
- [21] E. Microelectronics. Supertag category protocols. Datasheet. <http://www.gaw.ru/doc/EM-Marin/P4022.PDF>.
- [22] RFID Journal. DOD releases final RFID policy. <http://www.rfidjournal.com/article/articleview/1080/1/1>.
- [23] RFID Journal. Wal-mart begins RFID process changes. <http://www.rfidjournal.com/article/articleview/1385>.
- [24] M. Roberti. BP leads the way on sensors. Webpage. <http://www.rfidjournal.com/article/articleview/1216/1/2/>.
- [25] D. H. Shih, P. L. Sun, D. C. Yen, and S. M. Huang. Taxonomy and Survey of RFID Anti-collision Protocols: Short Survey. *Elsevier Computer Communications*, 29(11):2150–2166, 2006.
- [26] Skyetek. Skyemodule M1-Mini. Datasheet. http://www.skyetek.com/Portals/0/SkyeModule_M1Mini_060426.pdf.
- [27] H. Vogt. Multiple object identification with passive RFID tags. In *IEEE Intl. Conf. on Man and Cybernetics*, pages 6–13, Tunisia, Oct 2002.
- [28] J. Wang, Y. Zhao, and D. Wang. A novel fast anti-collision algorithm for RFID systems. In *International Conference on in Wireless Communications, Networking and Mobile Computing (WiCom 2007)*, pages 2044–2047, Shanghai, China, Sept 2007.
- [29] B. Zhen, M. Kobayashi, and M. Shimizu. Framed aloha for multiple RFID objects identification. *IEICE-Transactions on Communications*, E88-B:991–999, 2005.