

On the Accuracy of RFID Tag Estimation Functions

Dheeraj K. Klair, Kwan-Wu Chin and Raad Raad
 Telecommunications Information Technology Research Institute
 University of Wollongong
 Northfields Avenue, NSW, Australia
 {dkk282, kwanwu, raad}@uow.edu.au

Abstract—Dynamic Framed Slotted Aloha (DFSA) based tag reading protocols rely on a tag estimation function to calculate the best frame size to use for a given tag set. An inaccurate estimate results in high identification delays and unnecessary energy wastage. This is particularly serious when DFSA based tag reading protocols are used in RFID-enhanced wireless sensor networks (WSNs), where nodes are battery constrained. To this end, this paper presents qualitative and quantitative analysis of five tag estimation functions using Monte Carlo simulations. We iteratively estimate a given set of tags and evaluate the mean error, variability, skew and Kurtosis of each function’s error distribution. Lastly, we compare and identify the most efficient tag estimation function that is suitable for RFID-enhanced WSNs.

I. INTRODUCTION

In recent years, RFID has become a technology favored highly for object identification and supply chain management. Unlike conventional barcodes, RFID enables the identification of multiple tags wirelessly without requiring the line of sight. Some example applications include asset tracking, toll collection, and elder healthcare. Beside these applications, researchers are considering integrating an RFID reader to wireless sensor nodes to create an RFID-enhanced WSN.

An RFID-enhanced WSN has the ability to self-organize and form a multi-hop network capable of reading RFID tags. An example is tracking books in a library where RFID equipped wireless sensor nodes are placed on bookshelves. These nodes then form a connected network which users can submit queries to in order to obtain the whereabouts of a RFID tagged book. Other examples include [12] and [18].

A key constraint when deploying RFID-enhanced WSNs is their limited battery lifetime. In our previous work [14], we found that the energy expended to scan a single tag 96 bits in length is higher than the energy consumed to transmit and receive 96 bits of data by a sensor node. The energy consumption becomes even higher when there are multiple tags in a reader’s interrogation zone. It is therefore crucial that we analyze the underlying energy consumption issues of RFID anti-collision protocols.

In [15], we analyze Framed Slotted Aloha based RFID anti-collision protocols. Such protocols have the ability to adjust their frame size in accordance with varying tag population using a tag estimation function. Consider a reader with n tags in its interrogation zone. Initially, the reader starts the collision resolution process with an arbitrary frame size. Tags then choose a slot randomly to transmit their identification

(ID). The reader monitors the status of each slot and counts the number of slots filled with zero, one or multiple tag responses. This information is then manipulated by a tag estimation function to obtain a tag estimate, n' . Apart from monitoring slots’ status, a few tag estimation functions [21][5] also use the current frame size N in their calculations. Once an estimate is computed, the reader adjusts its frame size accordingly.

An inaccurate tag estimate results in non-optimal frame sizes, which increases identification delay and energy wastage. To date, very little works have conducted a comprehensive study on the accuracy of current tag estimation functions. Vogt [21][20] propose two tag estimation functions and analyzes their accuracy using weighted error and variance. On the other hand, Kodialam [17] et al. analyze the accuracy of their tag estimation functions using only the variance of tag estimates. Both works are limited to their respective tag estimation functions. Floerkemeier [10][9] compared his tag estimation function using two approaches. In the first approach, he compared his tag estimation functions with the Slot-count (Q) algorithm [1] and Schoute’s algorithm [19] by evaluating the difference between simulated and theoretical estimate. In the second approach, he compared one of Vogt’s [21] function to Schoute’s [19] tag estimation functions with his Bayesian approach using a test-bed comprising of a field programmable RFID reader and 64 HF Philips I Code RFID tags [9]. Floerkemeier, however, did not evaluate the functions proposed by Zhen et al. [22], and Cha et al. [5][6]. Zhen et al. [22], Cha et al. [5][6] and Khandelwal et al. [13] analyze the identification delays of their proposed estimation functions instead of accuracy. As a result, amongst the aforementioned tag estimation functions, it is unclear which is the best or most accurate. Moreover, the aforementioned works have not used a consistent set of metrics in their studies. Given these limitations and the impact of tag estimation functions on the performance of DFSA, it is critical that a comprehensive study is carried out to identify the best function to use.

In this paper, we present an analysis of five tag estimation functions [21][5][22]. We quantify their accuracy according to their mean error and variance from the actual tag number. Moreover, we analyze their computational complexity qualitatively. We found Vogt’s tag estimation function that is based on Chebychey’s inequality to be the most accurate among the five estimation functions. However, it suffers from high computational costs. Zhen et al.’s function, on the other hand, is computational efficient, and has comparable accuracy to Vogt’s function when the number of tags is close to the

frame size. Therefore, we recommend Vogt's function if a sensor node has sufficient computational resources; otherwise, Zhen et al's function is a good alternative, assuming low tag numbers.

This paper is organized as follows. Section II gives an introduction to RFID systems, describes our motivation and presents a qualitative analysis of tag estimation functions. In Section III, we define our system model and the research methodology used to quantify the error distribution of each tag estimation function. In Section IV, we present our results before concluding in Section V.

II. BACKGROUND

As shown in Figure 1, an RFID system consists of a reading device called an RFID reader and a finite number of tags, which may be passive, active or semi-passive; depending on how they are activated by a RFID reader. Passive tags have no power source and on-tag transmitter. They use the power emitted from the reader to energize and transmit their respective 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 and has an on-board transmitter for sending data or ID.

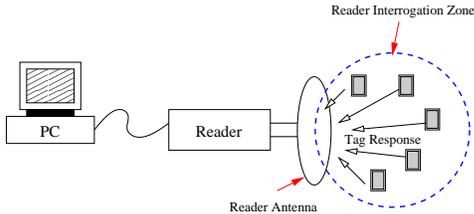


Fig. 1. Reader and tags interactions.

Collision is a key problem in RFID systems. It leads to bandwidth and energy wastage, and increases identification delay [8]. Numerous anti-collision protocols have been developed for RFID systems. Out of these, TDMA based protocols constitute the largest group of anti-collision methods [8], and they can be classified into Aloha and tree based algorithms. Tree based algorithms can be considered as an advanced polling method which uses the unique characteristics of each individual tag. On the other hand, Aloha based protocols operate asynchronously or in synchronized slots [3][4][2][7][11].

A. Motivation

In Frame Slotted Aloha (FSA) based protocols, a tag selects a slot randomly in a frame, and waits for a random time period before transmitting. Unlike conventional Pure and Slotted Aloha, a tag is only permitted to transmit its ID at most once per frame. FSA protocols have a fixed or varying frame size [15]. We refer to the former as basic FSA (BFSA) and the later as dynamic (DFSA) respectively.

FSA based protocols are often measured according to their system efficiency; defined as the ratio of slots filled with one tag over the current frame size. The system efficiency, $a_1^{N,n}$, of BFSA protocols is given as [16][19],

$$a_1^{N,n} = \frac{n}{N} \left(1 - \frac{1}{N}\right)^{n-1} \quad (1)$$

where n and N denote the number of tags and frame size respectively. The optimal throughput for BFSA can be evaluated from Equ. 1 and is equal to e^{-1} for $N = n$ [9].

Figure 2 plots Equ. 1 with varying number of tags and frame sizes; $N=64, 128, 256$ and 512 . We can clearly see that system efficiency starts to fall as the number of tags increases beyond the number of slots within a frame, i.e., $N > n$, and drops nearly to zero as the number of tags becomes very large. The system efficiency is at its highest when the number of tags is equal to the number of slots in a frame [16]. Therefore, if the frame size can be adjusted accurately, a reader will operate at the highest system efficiency. Thereby, avoiding any unnecessary energy wastage.

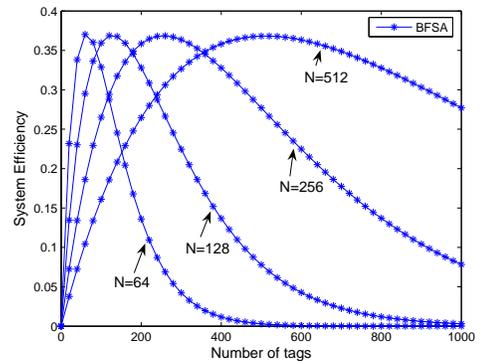


Fig. 2. System efficiency versus Number of Tags.

DFSA protocols have the ability to adjust their frame size according to tag estimates calculated by a tag estimation function. In order to achieve minimal delay and high system efficiency, the frame size used must match the number of tags [19][5]. Thus, DFSA protocols' performance is largely dependent upon the tag estimation function.

B. Tag Estimation Functions

A tag estimation function calculates the number of tags based on feedback obtained from a reader's frame. Here, feedback refers to slots with no reply (c_0), one tag reply (c_1), and multiple tag replies (c_k). Based on this feedback, researchers have developed several estimation functions.

1) *Vogt*: Vogt [20] present two tag estimation functions; referred to as DFSAV-I and DFSAV-II. For both functions, Vogt assumes tags reply in each read round. The muting feature, where tags are stopped from replying after they are identified, is not considered.

DFSAV-I is given by Equ. 2, and the number of estimated tags is denoted as ε_{lb} . DFSAV-I assumes all collisions are caused by two tags, hence the term $2c_k$. However, ε_{lb} quickly becomes erroneous as the number of tags increases since a collision in each slot may be caused by two or more tags.

$$\varepsilon_{lb}(N, c_0, c_1, c_k) = c_1 + 2c_k \quad (2)$$

DFSAC-II takes a different approach and is based on Chebyshev's inequality. The algorithm determines the distance between an actual read result vector $\langle c_0, c_1, c_k \rangle$ and the theoretical expected result vector $\langle a_0^{N,n}, a_1^{N,n}, a_k^{N,n} \rangle$ of a reading cycle, see Equ 3. The value of t at which ε_{vd} is minimum corresponds to the tag estimate.

$$\varepsilon_{vd}(N, c_0, c_1, c_k) = \min_t \left| \begin{pmatrix} a_0^{N,t} \\ a_1^{N,t} \\ a_k^{N,t} \end{pmatrix} - \begin{pmatrix} c_0 \\ c_1 \\ c_k \end{pmatrix} \right| \quad (3)$$

In Equ. 3, the elements of the vector $\langle a_0^{N,t}, a_1^{N,t}, a_k^{N,t} \rangle$ correspond to the expected number of empty slots, slots filled with one tag, and slots with collisions respectively. With a frame size of N , and the number of tags t , the expected number of slots filled with r responding tags is given by,

$$a_r^{N,t} = N \times \binom{t}{r} \left(\frac{1}{N}\right)^r \left(1 - \frac{1}{N}\right)^{t-r} \quad (4)$$

From Equ. 4, we calculate each element of the vector $\langle a_0^{N,t}, a_1^{N,t}, a_k^{N,t} \rangle$ as follows [20]:

$$a_0^{N,t} = N \times \left(1 - \frac{1}{N}\right)^t \quad (5)$$

$$a_1^{N,t} = t \times \left(1 - \frac{1}{N}\right)^{t-1} \quad (6)$$

$$a_k^{N,t} = N - a_0^{N,t} - a_1^{N,t} \quad (7)$$

According to [21], DFSAC-II is more applicable to scenarios where tag densities are high. However, DFSAC-II is computationally more complex than DFSAC-I.

2) *Cha et al.*: Cha et al. [5] present two tag estimation functions; referred to as DFSAC-I and DFSAC-II.

DFSAC-I computes a tag estimate from the collision rate that is derived using the "maximum throughput condition". The collision rate and maximum throughput condition are defined as follows.

According to Cha et al., the throughput S of FSA is defined as,

$$S = \frac{P_{succ}}{P_{succ} + P_{coll} + P_{idle}} \quad (8)$$

where $P_{coll} = 1 - P_{idle} - P_{succ}$ and $P_{idle} = (1 - p)^n$ and $P_{succ} = np(1 - p)^{n-1}$ respectively; $p = \frac{1}{N}$ is the probability that one tag transmits at a particular slot. The maximum throughput happens when,

$$\frac{dS}{dp} = 0 \quad (9)$$

Solving Equ. 9, we get,

$$p = \frac{1}{n} \quad (10)$$

Using Equ. 10, the collision rate is calculated as [5],

$$C_{rate} = \lim_{n \rightarrow \infty} \frac{P_{coll}}{1 - P_{succ}} \quad (11)$$

Inserting P_{coll} , P_{succ} and Equ. 10 into Equ. 11, we get $C_{rate} = 0.4180$, which Cha et al. refer to as the "maximum throughput condition". Using C_{rate} , the number of colliding tags in a slot is then calculated as,

$$C_{tags} = \frac{1}{C_{rate}} = 2.3992 \quad (12)$$

From Equ. 12, if the number of colliding slots is c_k , the number of tags is estimated as $2.3992c_k$

DFSAC-II estimates a tag set based on collision ratio. Cha et al. define collision ratio to be the ratio of the number of slots with collisions with respect to the frame size, and is computed as follows [5],

$$C_{ratio} = 1 - \left(1 - \frac{1}{N}\right)^n \left(1 + \frac{n}{N-1}\right) \quad (13)$$

where n is the estimated number of tags, and is computed after a read round from the number of colliding slots according to,

$$C_{ratio} = \frac{c_k}{N} \quad (14)$$

Using Equ. 14, Cha et al. solves for n iteratively by equating Equ. 13 to Equ. 14.

DFSAC-II is computationally more complex than DFSAC-I. Note, both these functions are developed for scenarios where tags are muted once identified.

3) *Zhen et al.*: Referred to as DFSAC, their function works as follows. For an observed slot, the *a posteriori* probability distribution of k tags choosing a slot is,

$$p_k^0(i) = \begin{cases} 0 & \text{if } k = 0, 1 \\ \frac{p_k(i)}{1 - p_0(i) - p_1(i)} & \text{if } k \geq 2 \end{cases} \quad (15)$$

From Equ. 15, the *a posteriori* expected value of a garbled slot is, $\lim_{N \rightarrow \infty} \sum_{k=2}^N k p_k^0(i) = 2.39 = K$. This indicates that on average 2.39 tags responded in a collided slot. Thus, according to Zhen et al. [22], the estimated number of tags is $c_1 + 2.39c_k$.

The estimation function is devised for passive tag environments and the muting feature is not taken into consideration.

4) *Summary*: The aforementioned functions utilize two main methodologies for tag estimation. The first, called *static estimation*, is based on computing the number of tags by observing the number of collided slots and multiplying that with a constant. DFSAC-I, DFSAC-II and DFSAC-III belong to this methodology. The second methodology, called *dynamic estimation*, compares the read results obtained after a read round with theoretically computed values to arrive at an estimate. DFSAC-II and DFSAC-III are examples of this methodology.

Static estimation algorithms suffer from erroneous estimates as the number of tags increases. This is due to the use of a constant multiplier which does not consider a large number of tags colliding in a single slot. On the other hand, dynamic estimation schemes do not have this assumption. Besides that, both static and dynamic algorithms require different parameters. The later incorporate the current frame size N whereas the former relies mainly on feedback obtained by observing the status of slots.

Another way to categorize tag estimation functions is based on whether they support the muting feature, where tags are muted after identification. Since the number of tags decreases with each successful read, muting affects how a tag estimation function operates. Tag estimation functions which consider muting use the number of collided slots in their computation. They do not consider slots with identified tags. On the other hand, tag estimation functions which do not consider muting need to take into account identified tags. Amongst the functions we reviewed, DFSAC-I and DFSAC-II consider muting whereas DFSAZ, DFSAV-II and DFSAV-I do not.

The computational requirements of tag estimation functions vary according to the methodology employed for estimation. Static estimation techniques are simpler to implement and have low computational requirements. Their computation only involves simple additions and multiplications. On the other hand, dynamic estimation techniques have higher computational requirements since they need to evaluate theoretical values and compare them to read values in order to compute an estimate. Overall, DFSAV-I has the least computational requirement. This is followed by DFSAC-I, due to its use of floating point multiplications. DFSAZ has higher requirements than DFSAC-I as it involves both integer and floating point calculations. Finally, DFSAC-II and DFSAV-II have the highest computational requirements.

The foremost requirement of a tag estimation function is its accuracy. An inaccurate estimate leads to idle slots or collisions; both of which lead to unnecessary delays and energy wastage. Next, we present our system model before describing how we evaluate and quantify the accuracy of the aforementioned functions.

III. RESEARCH METHODOLOGY

The system consists of an RFID reader and n tags in its interrogation zone. We assume tags are passive, have no power source and they are used in read-only mode. Further, tags are static and can be read regardless of their orientation¹.

A reader starts collision resolution with an arbitrary frame size N . It then observes tag responses in each slot. We refer to these responses as slots with no reply (c_0), slots with one tag reply (c_1), and slots with multiple tag responses (c_k).

We use descriptive statistics to quantify the error distribution of tag estimation functions. For each corresponding error distribution, we evaluate its mean error, standard deviation, variance, skewness and Kurtosis.

The error in the u^{th} read round is evaluated as,

$$e(u) = n(u) - n' \quad (16)$$

where $e(u)$ is the error estimated in the u^{th} read round, $n(u)$ is the number of tags estimated using a tag estimation function in the u^{th} read round, and n' is the actual number of tags in a reader's interrogation zone, respectively. If $e(u)$ is negative, it indicates the estimated number of tags is less than the actual tag count, and vice versa.

¹The placement of tags with respect to the polarization of the reader's field can have a negative affect on the communication distance of RFID tags, in particular, reduced operating range and in the case of the tag being displaced by 90 degrees, a tag becomes unreadable.

In Equ. 16, $e(u)$ and $n(u)$ are random variables and n' is a constant. $e(u)$ and $n(u)$ are evaluated at each read cycle and $e(u)$ is dependent upon the estimated number of tags $n(u)$ in a particular read round.

We perform R read cycles on a tag set with n' tags and then calculate the mean of the distribution $e(u)$ as,

$$\mu(n') = \frac{1}{R} \sum_{u=1}^R (e(u)) \quad (17)$$

$$= \frac{1}{R} \sum_{u=1}^R (n(u) - n') \quad (18)$$

where $\mu(n')$ denotes the mean of the error computed in R read rounds.

The mean percentage of error of each tag estimate is evaluated as,

$$\mu_{-per}(n') = \frac{1}{R} \sum_{u=1}^R \frac{e(u)}{n'} \times 100 \quad (19)$$

In order to evaluate the variability or spread of the distribution $e(u)$, we evaluate its standard deviation as,

$$s = \sqrt{\frac{\sum_{u=1}^R (e(u) - \mu)^2}{R - 1}} \quad (20)$$

Squaring Equ. (20) gives us the variance $\sigma(n')$.

The skewness, i.e., lack of symmetry of $e(u)$, can be evaluated as

$$sk(n') = \frac{3(\mu(n') - m(n'))}{s(n')} \quad (21)$$

where $m(n')$ is the median of $e(u)$.

Finally, we measure the flatness or peakness of a distribution, i.e., Kurtosis of $e(u)$, as,

$$kurt = \frac{\sum_{u=1}^R (e(u) - \mu(n'))^4}{Rs^4} \quad (22)$$

If the distribution is relatively flat compared to the normal or bell-shaped distribution, we refer $e(u)$ to as platykurtic. Otherwise, it is called leptokurtic.

Algorithm 1 shows the pseudo-code of our implementation. The function *perform_read_cycle*(N, n) returns the vector $\langle c_0, c_1, c_k \rangle$ after being given the following parameters; frame size and actual number of tags in the current read cycle. The result is then used by *estimate_tags*(N, c), which returns the estimated tags obtained by the tag estimation function being investigated. The rest of the pseudo-code is an implementation of equations 16, 18, 20, 21, and 22.

IV. RESULTS

In this section, we present a detailed comparison analysis of tag estimation functions based on mean error, variability, skew and Kurtosis. The frame size in analysis is fixed to N and the performance of each tag estimation function is evaluated with varying number of tags.

```

N=32;
t=100;
R=10000;
for n = 1 : t do
  for u = 1 : R do
    c = perform_read_cycle(N,c);
    n(u)=estimate_tags(N,c);
    e(u)=n(u)-n';
  end
  mu(n') = mean (e(u),R);
  mu_per(n') = percent_error(e(u),n');
  s(n') = standard_deviation(e(u),R);
  sigma(n') = (s(n'))^2;
  sk(n') = skewness (e(u));
  kurt(n') = kurtosis (e(u));
end

```

Algorithm 1: Procedure for evaluating tag estimation functions.

5) *Mean error*: Figure 3 depicts the mean error in tag estimates. It can be seen that DFSAC-I has the highest mean error. When the number of tags is greater than 68, DFSAV-I becomes the worst performing function. DFSAC-II and DFSAV-II have the lowest mean error for a wide range of tag set. DFSAC-II, however, has a lower accuracy than DFSAV-I when the number of tags is less than 20. This is because DFSAC-II uses only collision slots when estimating tags. When the number of tags is less than the frame size used, the chances of collision becomes low. Thereby, we see the mean error for DFSAC-II is higher when the number of tags is lower than the frame size used. DFSAZ has a lower mean error than DFSAC-I and DFSAV, and similar error values to DFSAC-II and DFSAV-II for $n' < 40$, and becomes more erroneous when $n' > 40$. It can be observed that DFSAC-II outperforms DFSAV-II in accuracy when the numbers of tags starts to increase beyond the frame size. Overall, DFSAC-I, DFSAZ, DFSAV-I have better performance when the number of tags is similar to the frame size used. DFSAC-I has the lowest accuracy compared to the others.

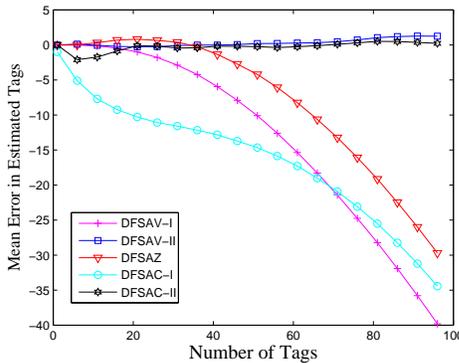


Fig. 3. Mean error versus number of tags.

6) *Variability*: Figure 4 compares the variability of each function's error distribution. DFSAV-I has the most stable error

estimates, followed by DFSAZ and DFSAC-I. DFSAV-II and DFSAC-II have higher variability in their error distribution, and therefore the error estimates are unstable. The variability of DFSAC-II is very unstable for tags ranging from 0 to 25. DFSAC-II has lower variability than DFSAV-II for $n' > 35$.

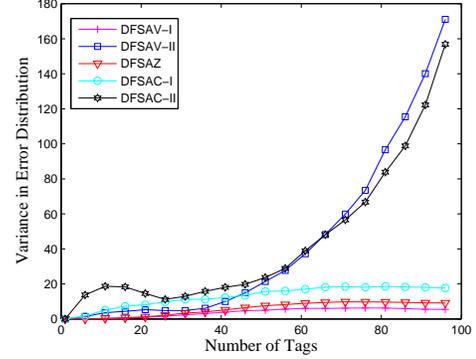


Fig. 4. Variability versus Number of Tags.

7) *Skew*: Figure 5 depicts the skew observed in the error distribution of each tag estimation function. The skew of the distribution does not show a generalized pattern since it spans positive and negative values, and varies with tag numbers.

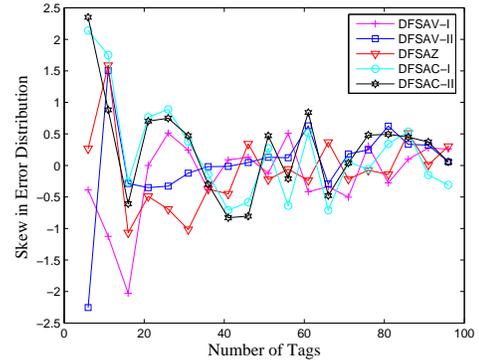


Fig. 5. Skew versus Number of Tags.

8) *Kurtosis*: Figure 6 plots the Kurtosis for all tag estimation functions. All estimation functions are leptokurtic. The Kurtosis values for DFSAC-II vary widely compared to other estimation functions. When there are less than 10 tags, its Kurtosis is very high. This is because when the number of tags is very low, the functions can estimate the number of tags accurately; yielding a small standard deviation value. As Kurtosis is inversely proportional to standard deviation, it reaches a very high value when the number of tags is low.

9) *Summary*: Table I presents a summary of the aforementioned statistical analysis for low tag density $n' = 32$ and high tag density $n' = 100$. From Table I, when $n' = 32$, DFSAZ has the lowest percentage of error. On the other hand, when $n' = 100$, DFSAC-II has the lowest error percentage. DFSAV-I has the lowest variance for $n' = 100$ as well as $n' = 32$. However, DFSAC-II and DFSAV-II have very high variability in their error distribution. The Kurtosis for DFSAV-II is significantly higher than other functions for $n' = 100$.

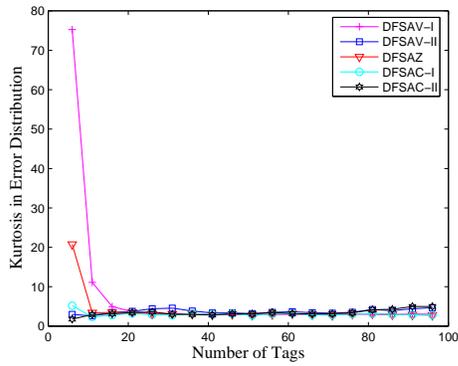


Fig. 6. Kurtosis versus Number of Tags.

DFSAV-I is the only entry in Table I which is negatively skewed for $n' = 32$. All others are positively skewed and leptokurtic.

From the comparison analysis, DFSAV-II has the best performance; both for low and high n' values. However, this is achieved at a higher computational cost. DFSAC-II has a higher accuracy than DFSAV-II, especially when the number of tags is higher than the frame size used. DFSAC-I is the least accurate amongst all functions. It underestimates the tag set as it does not consider slots with a single tag response.

TABLE I
SUMMARY OF STATISTICS FOR ERROR DISTRIBUTION PLOTS

Function	Mean error	Standard Deviation	Variance	Skew	Kurtosis
$R = 10000, n' = 100, N = 32$					
DFSAV-I	-42.9608	2.2547	5.0838	0.0522	2.9431
DFSAV-II	1.7528	14.6118	213.5056	0.7705	26.0821
DFSAZ	-32.6555	2.8684	8.2274	0.3603	2.9028
DFSAC-I	-36.9455	4.0686	16.5534	0.7775	2.8955
DFSAC-II	1.0827	14.0181	196.5071	1.0396	8.0436
$R = 10000, n' = 32, N = 32$					
DFSAV-I	-3.1032	1.5908	2.5308	-0.1946	3.0082
DFSAV-II	-0.0682	2.2055	4.8644	-0.0928	3.7691
DFSAZ	0.2843	1.8487	3.4178	0.4613	3.0047
DFSAC-I	-11.6616	3.3170	11.0024	1.2105	2.8183
DFSAC-II	-0.4589	3.6836	13.5687	1.2551	2.8448

V. CONCLUSION

We have analyzed the accuracy of five tag estimation functions by evaluating the mean error and statistics of their error distribution. DFSAV-II has the best accuracy. Therefore, we find it suitable for RFID-enhanced WSNs. One caveat however is its high computational requirements.

VI. ACKNOWLEDGMENT

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

REFERENCES

[1] Alien Technology. EPCglobal Class 1 Gen 2 RFID Specifications. Standard. http://www.alientechnology.com/docs/AT_wp_EPCGlobal_WEB.pdf.

[2] F. Bo, L. Jin-Tao, G. Jun-Bo, and D. Zhen-Hua. ID-binary tree stack anticollision algorithm for RFID. In *The 11th IEEE Symposium on Computers and Communications (ISCC '06)*, pages 26–29, 2006.

[3] L. Bolotny and G. Robins. Randomized pseudo-random function tree walking algorithm for secure radio-frequency identification. In *The 4th IEEE Workshop on in Automatic Identification Advanced Technologies*, pages 43–48, 2005.

[4] M. A. Bonuccelli, F. Lonetti, and F. Martelli. Tree slotted Aloha: a new protocol for tag identification in RFID networks. In *World of Wireless, Mobile and Multimedia Networks*, pages 28–34. IEEE, 2006.

[5] J.-R. Cha and J.-H. Kim. Novel anti-collision algorithms for fast object identification in RFID system. In *The 11th Intl. Conference on Parallel and Distributed Systems*, pages 63–67, 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, 2006.

[7] H.-S. Choi, J.-R. Cha, and J.-H. Kim. Fast wireless anti-collision algorithm in ubiquitous ID system. In *The 60th IEEE Vehicular Technology Conference*, pages 4589–4592, 2004.

[8] K. Finkenzerler. *RFID Handbook, Fundamentals and Applications in Contactless Smart Cards and Identification*. John Wiley and Sons Ltd., 2003.

[9] C. Floerkemeier. Transmission control scheme for fast RFID object identification. In *The 4th Annual Intl. Conference on Pervasive Computing and Communications Workshops*, 2006.

[10] C. Floerkemeier and M. Wille. Comparison of transmission schemes for framed ALOHA based RFID protocols. In *International Symposium on Applications and the Internet Workshops*, 2006.

[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, 1998.

[12] Intel. Intel sensor nets/RFID research areas. Webpage. http://www.intel.com/research/exploratory/wireless_sensors.htm.

[13] G. Khandelwal, A. Yener, K. Lee, and S. Serbetli. ASAP: a MAC protocol for dense and time constrained RFID systems. In *IEEE International Conference on Communications (ICC'06)*, 2006.

[14] 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 International Symposium on a World of Wireless, Mobile and Multimedia Networks (IEEE WoWMoM'07), June 18-21, Helsinki, Finland, 2007*.

[15] D. K. Klair, K.-W. Chin, and R. Raad. On the suitability of framed aloha based RFID anti-collision protocols for RFID-Enhanced WSNs. To Appear. In *The 16th IEEE International Conference on Computer Communications and Networks (IEEE ICCCN'2007)*. Honolulu, Hawaii, USA, 2007.

[16] 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, 2005.

[17] M. Kodialam and T. Nandagopal. Fast and reliable estimation schemes in RFID systems. In *SIGMOBILE: ACM Special Interest Group on Mobility of Systems, Users, Data and Computing ACM: Association for Computing Machinery*, pages 322–333, 2006.

[18] NASA/JPL. Sensor webs project. <http://sensorwebs.jpl.nasa.gov/>.

[19] F. C. Schoute. Dynamic frame length Aloha. *IEEE Transactions on Communications*, 31(4):565–568, 1983.

[20] H. Vogt. Efficient object identification with passive RFID tags. In *The International Conference on Pervasive Computing (PerCom)*, 2002.

[21] H. Vogt. Multiple object identification with passive RFID tags. In *The IEEE Intl. Conf. on Man and Cybernetics*, pages 6–13, 2002.

[22] B. Zhen, M. Kobayashi, and M. Shimizui. Framed Aloha for multiple RFID objects identification. *IEICE-Transactions on Communications*, E88-B:991–999, 2005.