Efficient Median Estimation for RFID Systems

Huda El Hag Mustafa*†, Xiaojun Zhu*‡, Qun Li*
*Department of Computer Science, College of William and Mary, Williamsburg, VA
†Department of Computer Science, Faculty of Mathematical Sciences, University Of Khartoum, Sudan
‡State Key Laboratory for Novel Software Technology, Nanjing University, P. R. China
Email: {huda,njuxjzhu,liqun}@cs.wm.edu

***Abstract*—Radio frequency identification (RFID) tags are used in a lot of applications such as production line and inventory management. Sensors are also a widely adopted wireless technology. This large scale deployment of both RFID technology and sensor technology has opened the door to innovative ways to integrate RFID and sensor technology. Sensor-tags are tags that can report sensed information to an RFID reader. The traditional way to obtain information from tags is to query each tag for its value. This works, however, only for a relatively small number of tags. When the number of tags becomes large it is prohibitive to query one by one due to the high delay. In this paper we present a probabilistic algorithm for estimating the median of a set of RFID tags using binary search. We then evaluate the accuracy and time efficiency of our algorithm.**

I. Introduction

Radio Frequency IDentification (RFID) technology is used in applications in which it is important to scan and monitor tagged objects. These applications are very diverse and can range from military applications to applications for elderly care. In a lot of application domains RFID tags are integrated with sensors, so that the tags not only report their IDs, but also sensed data. An example of tags integrated with sensors is WISP tags [1]. WISP tags can report quantities such as temperature, liquid level, and light. Inside the WISP tags, the energy taken from the reader operates a 16-bit programmable, low power micro-controller unit. The micro-controller can sample data from the sensors and report that data when probed by the readers.

We consider a sensor-RFID deployment in which RFID technology is used for identification and communication and sensors are used to periodically sense diverse physical quantities. We are interested in designing efficient protocols to estimate the aggregate of the sensed data in this type of systems. We argue that aggregating sensor readings is necessary in many applications because of the following reasons. First, it is more efficient to conduct data aggregation at the RFID reader in terms of system running time. Instead of collecting the sensor readings from all the sensor-tags, we can design efficient protocols to quickly aggregate the sensed data. Second, these sensor-tag deployments generate a large amount of data, which flows through the network. If the data is not aggregated it will result in an enormous amount of traffic flowing through the network [20]. Third, sensors are usually deployed in rugged areas, where there is no physical protection so the the sensors can be compromised. This leads to inaccurate sensor values [18][19][21]. Fourth, in most sensor deployments it is important to report the value of a variable in a specific area of the deployment, in this case individual sensor values are not required, rather some aggregate value is needed. This aggregate value can be MEDIAN, SUM, COUNT or AVERAGE. Median is an aggregate which is very robust because it is insensitive to extreme values or outliers.

We aim to develop an efficient algorithm to estimate the median of a large number of sensor-RFID tags, in-network, at the RFID reader. To the best of our knowledge, this paper presents the first efficient median estimation algorithm for sensor-RFID tags. Instead of collecting the sensed data from all the tags, we selectively query tags to quickly find the median. Our contributions can be summarized as follows: (1) We design a binary search algorithm to solve the median computation problem in RFID tags without probing the tags individually. We implement the binary search algorithm using a simple Threshold Checking Scheme (TCS) [6]. TCS is used to test whether the number of active tags is more than a given threshold. (2) We analyze our proposed scheme and show the performance via simulation. Our evaluation demonstrates that our algorithm is accurate and has significantly better execution time than the conventional methods.

II. Related Work

In sensor networks a lot of work has been done on data aggregation techniques [18][19][20][21]. In [18] a declarative query interface was proposed that allowed users to perform aggregate operations such as MIN, MAX, MEDIAN and similar operations. This approach showed improved performance over centralized techniques, since the sensors collaborated to obtain an accurate result. In [22] two algorithms were proposed to compute the median, a randomized protocol which computed an approximate median and a deterministic protocol which computed an exact median.

Numerous protocols to identify tags have been proposed for RFID systems [2][3][4][5][6][7][25]. Probabilistic estimation algorithms which efficiently estimate the cardinality of RFID tags have been proposed in numerous works [10][8][11]. To the best of our knowledge no one has proposed a median search algorithm for sensor-RFID tags. We believe the research in this paper on RFID sensor can be applied to other sensor network or security problems [13][14][15][16][23][24][26].

III. System Model and Problem Statement

A. System model

We consider a system in which there are n sensor-tags $t_1, t_2, ..., t_n$, each sensor-tag has a unique ID. We assume that associated with each sensor-tag is a data value y, which represents the quantity measured by the sensor. Our communication model is built upon the slotted ALOHA protocol [11]. In this protocol each frame is made up of a number of time slots. The reader broadcasts a frame size and a random seed S. An RFID tag uses a hash function $H(\cdot), f, S$ and its ID to pick a slot to communicate in. In addition to these parameters, the reader broadcasts a number y, and only the tags with value less than y will communicate in the following round.

B. Problem definition

Given an RFID reader and a large set of tags we want to accurately estimate the median value of the tags, in minimum time, without probing each tag individually. In our implementation the confidence level of the algorithm is a user specified value ϵ. Since the time window for the execution of our median search algorithm is very small we assume that no tags are removed while the median estimation algorithm is running.

In this paper we use two performance metrics which are: (1) The execution time of the median search algorithm which is defined as the time taken to estimate the median. (2) The accuracy of the median value.

IV. Median Estimation Algorithm

In this section we describe our median estimation algorithm, which is based on binary search using TCS. We give a formal description of the algorithm in Algorithm 1, the input consists of frame size f, maximum tag set cardinality n, and minimum data value E_{min} and maximum data value E_{max}. The reader then computes the load factor $\rho = n/(2f)$ (Line 3), and calculates τ_c which is the expected number of collision slots (n_c), if the number of responding tags is $n/2$ and the frame size is f. The basis of the algorithm is a slotted ALOHA scheme (Lines 7-9). The estimated value for the median x is then computed. At the beginning, the RFID reader probes the tags using the slotted ALOHA protocol with frame length f and the value x. The reader will then infer the number of tags based on the number of collision slots n_c. We chose the number of collision slots because the number of singleton slots is not monotonic with respect to the number of tags. If the number of collision slots is greater/less than the $\tau_c \pm \epsilon$ then the median is set to a new value and the reader broadcasts this new value, otherwise the value x is the required median (Lines 10-15).

Algorithm 1: Median Estimation Binary Search Algorithm

```
   Input: ϵ, n, f, E_max, E_min;
   (/* error bound, tag number, frame
   size, max. and min. values of the tags
   */)
   Output: x; the median
1  begin
2      u = E_max; l = E_min;
3      ρ ⟵ n/(2f);
4      τ_c ⟵ f(1 − (1 + ρ)e^{−ρ});
5      while true do
6          x ← (l + u)/2;
7          Reader broadcasts f and x ;
8          Each tag with value less than x randomly picks a
           time slot to reply;
9          Reader gets the number of collision slots n_c;
10         if n_c > (1 + ϵ)τ_c then
11             u ⟵ x;
12         else if n_c < (1 − ϵ)τ_c then
13             l ⟵ x;
14         else
15             return x;
```

A. Threshold Checking Scheme

In this section we elaborate on the the algorithm underlying our binary search implementation, the Threshold Checking Scheme (TCS) [6]. The basic idea of Threshold Checking Scheme is to test whether the number of concerned tags (say an unknown value n') is greater or less than a threshold (n). We first calculate the expected number of collision slots (τ_c) given the number of tags (n) and frame size (f). We will then count the number of collision slots (n'_c) for the concerned tags and frame size f. If $n'_c > (1+\epsilon)\tau_c$, then TCS asserts that the number of concerned tags is larger than n. If $n'_c < (1+\epsilon)\tau_c$, the TCS asserts that the number of concerned tags is less than n. The parameter ϵ determines the confidence level.

V. Theoretical Analysis

In this section we will provide the theoretical analysis. To facilitate the following analysis, we introduce some basic properties of the number of collision slots. For t tags and f time slots, the number of collision slots is a random variable, which we denote by n_c. Reference [8] shows that n_c approximately follows the normal distribution with

$$\begin{cases} E[n_c] = f\left(1-(1+\rho)e^{-\rho}\right) & (1) \\ \sigma^2 = fe^{-\rho}\left((1+\rho)-\left(1+2\rho+\rho^2+\rho^3\right)e^{-\rho}\right) & (2) \end{cases}$$

where $\rho = t/f$ is the load factor. If we keep frame length f fixed, then the variance σ^2 varies with respect to the number of tags. We denote by $F(x)$ the number of tags with value less than x. Denote the median value as m. We assume that no more than 1 tag has the value m. Therefore, we have $F(m) = n/2$ (when n is odd, we can simply take a floor operation). Consider an iteration of the algorithm, and suppose the current candidate median is x. In an ideal case, if $F(x) < n/2$, then the iteration should turn right (i.e., $l \longleftarrow x$); if $F(x) > n/2$, then the iteration should turn left (i.e., $u \longleftarrow x$); otherwise, the iteration should break the

loop. We will consider the undesirable cases in the following analysis.

Consider the iteration that takes a wrong turn, that is, if that iteration gives a wrong range of the median. Let $\Phi(x)$ be the cumulative distribution function for standard normal distribution such that $\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}t^2}\, dt$. We then have the following lemma:

Lemma 1. *Suppose all previous iterations of the **while** loop give a correct range, then the current iteration takes a wrong turn with probability at most* $1 - \Phi\left(\frac{\epsilon\tau_c}{\sigma_{\max}}\right)$.

Proof: Let $[l, u]$ be the range of the median value at the previous iteration of the **while** loop. By assumption, $m \in [l, u]$. Let $z = (l+u)/2$ and n_z be the estimated number of collision slots (n_c in the algorithm). There are two undesired events where the current iteration takes a wrong turn.

(1) $F(z) < n/2$ but $n_z > (1+\epsilon)\tau_c$. Since function (1) is monotonically increasing with respect to the number of tags, we have $E[n_z] < \tau_c$. To see this, note that $F(z)$ is the number of tags in the expression $E[n_z]$, while $n/2$ is the number of tags corresponding to τ_c. Therefore,

$$\begin{aligned}
&\Pr\left[n_z > (1+\epsilon)\tau_c\right] \\
&= \Pr\left[\frac{n_z - E[n_z]}{\sigma_{n_z}} > \frac{(1+\epsilon)\tau_c - E[n_z]}{\sigma_{n_z}}\right] \\
&= 1 - \Phi\left(\frac{(1+\epsilon)\tau_c - E[n_z]}{\sigma_{n_z}}\right) \\
&< 1 - \Phi\left(\frac{\epsilon\tau_c}{\sigma_{\max}}\right)
\end{aligned}$$

(2) $F(z) > n/2$ but $n_z < (1-\epsilon)\tau_c$. Similarly, we have $E[n_z] > \tau_c$ and

$$\begin{aligned}
&\Pr\left[n_z < (1-\epsilon)\tau_c\right] \\
&= \Pr\left[\frac{n_z - E[n_z]}{\sigma_{n_z}} < \frac{(1-\epsilon)\tau_c - E[n_z]}{\sigma_{n_z}}\right] \\
&= \Phi\left(\frac{(1-\epsilon)\tau_c - E[n_z]}{\sigma_{n_z}}\right) \\
&< 1 - \Phi\left(\frac{\epsilon\tau_c}{\sigma_{\max}}\right)
\end{aligned}$$

Since the two events are mutually exclusive, the proof is complete. ■

Lemma 2. *Suppose the current iteration of the **while** loop has candidate median x such that $F(x) = n/2$, then the algorithm terminates at current iteration with probability* $2\Phi\left(\frac{\epsilon\tau_c}{\sigma_{\tau_c}}\right) - 1$.

Proof: Note that assumption $F(x) = n/2$ implies $E[n_c] = \tau_c$. Therefore,

$$\begin{aligned}
&\Pr\left[(1-\epsilon)\tau_c \le n_c \le (1+\epsilon)\tau_c\right] \\
&= \Pr\left[\frac{-\epsilon\tau_c}{\sigma_{n_c}} \le \frac{n_c - \tau_c}{\sigma_{n_c}} \le \frac{\epsilon\tau_c}{\sigma_{n_c}}\right] \\
&= 2\Phi\left(\frac{\epsilon\tau_c}{\sigma_{\tau_c}}\right) - 1
\end{aligned}$$

where the last equality comes from $\sigma_{n_c} = \sigma_{\tau_c}$. ■

Combining the above two lemmas, we have the following theorem.

Theorem 1. *Algorithm 1 finds the median with probability at least* $1 - \beta \log L$ *where* L *is the length of the tag value range and*

$$\beta = \max\left\{2 - 2\Phi\left(\frac{\epsilon\tau_c}{\sigma_{\tau_c}}\right), 1 - \Phi\left(\frac{\epsilon\tau_c}{\sigma_{\max}}\right)\right\}.$$

Proof: Each iteration involves three events, a correct turn, a wrong turn, and correct termination. No matter what the ground-truth situation is, the undesired event (making a wrong turn if it should make a turn, or making a turn if it should stop) of each iteration happens with probability at most β. Since there are at most $\log L$ iterations, the undesired event for the whole procedure happens with probability at most $\beta \log L$ by union bound. ■

For the consumed time, note that there are at most $\log L$ iterations and each iteration takes a time interval proportional to the frame length f. Thus the procedure finishes within at most $f \log L$ time slots.

VI. Performance Analysis

We evaluate the performance of the median estimation algorithm by simulation and the algorithm is compared to a tag identification algorithm. The metric used is the time taken to correctly evaluate the median and the accuracy. We we vary the number of tags n from 1000 to 50,000 tags. We evaluated our algorithm for different statistical distributions of the tag value data y.

In this section we first evaluate the time efficiency of the median binary search algorithm. The required threshold for the TCS scheme is set using the knowledge about the total number of tags. In this paper we use the fact that a long slot is equal to 5 short slots $L = 5S$ [6]. For the median estimation algorithm we set $\epsilon = 0.15$.

We carried out simulations on the uniform and normal tag set distributions. We observed that the uniform distribution exhibited the best time performance. This is due to the fact that the tags are evenly distributed, as in Figures 1 and 2. In the figures the solid line represents the time needed for our algorithm to converge while the dotted line represents the time need by an identification algorithm to solve the median selection problem.

In all the simulations for the median search, accuracy requirement was the second factor used to determine performance. The value range of the error is in the range $0 - 1$. We calculated the error as Absolute error = $|(M - M_{est})/(E_{max} - E_{min})|$, where M is the real median and M_{est} is the estimated median. The closer the error is to one the larger the error is. The binary median search algorithm was tested with different tag set distributions via simulation, and we observed that all the tests showed a high level of accuracy with high probability. In Figures 3 and 4 we observe that increasing the number of tags does not increase the absolute error, which shows that our algorithm is scalable with respect to tag set cardinality.

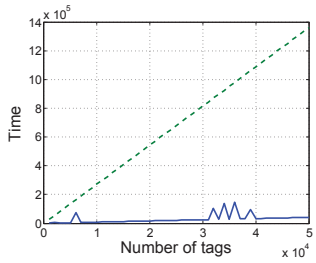

Fig. 1. Execution time for the uniform distribution in time units, the error was set to $\epsilon = 0.15$ and the maximum and minimum tag values (E_{max}, E_{min}) were generated according to the parameters of the uniform distribution.

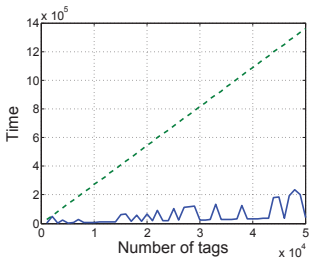

Fig. 2. Execution time for the Normal distribution in time units, the error was set to $\epsilon = 0.15$ and the maximum and minimum tag values (E_{max}, E_{min}) were generated according to the parameters of the normal distribution.

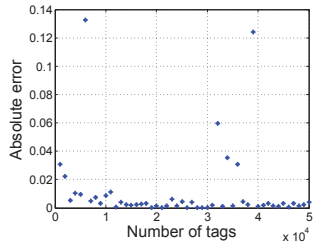

Fig. 3. Error in median value for the uniform distribution

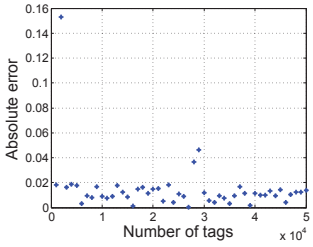

Fig. 4. Error in median value for the normal distribution

VII. Conclusion

In this paper we study the the median estimation problem in a large scale sensor-RFID system. This algorithm can be used with real time data collected from tags. We showed that the median can be estimated efficiently by probing the RFID tags using binary search. Since in our algorithm the accuracy requirement can be set by the user, this adds flexibility to the implementation. The frame length was minimized by minimizing the number of slots, and by using short slots. Since our implementation did not require tag identification, short slots achieved the required objective. In the implementation the frame length was set at $n/2.44$. We believe our algorithm can be implemented efficiently in massive deployments of RFID tags and with great accuracy.

Acknowledgment

This project was supported in part by US National Science Foundation grants CNS-1117412 and CAREER Award CNS-0747108.

References

[1] S. Roy, V. Jandhyala, J. Smith, D. Wetherall, B. Otis, R. Chakraborty, M. Buettner, D. Yeager, Y. Ko, A. Sample, "RFID: from supply chains to sensor nets," Proc. of the IEEE 2010, vol.98, no.9, pp.1583-1592.

[2] C. Law, K. Lee, and K.-Y. Siu, "Efficient memoryless protocol for tag identification," Proc. of the 1st International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications 2000, ACM, pp. 7584.

[3] A. Micic, A. Nayak, D. Simplot-Ryl, and I. Stojmenovic, "A hybrid randomized protocol for RFID tag identification," Proc. IEEE International Workshop on Next Generation Wireless Networks, 2005.

[4] L. Xie, B. Sheng, C. C. Tan, Q. Li, D. Chen, "Efficient tag identification in mobile RFID systems," Proc. Infocom 2010, San Diego, CA, Mar., pp. 15-19.

[5] C. C. Tan, B. Sheng, Q. Li, "How to monitor for missing RFID tags," Proc. IEEE ICDCS 2008, Beijing, China, June 17-20, pp. 295-302.

[6] B. Sheng, C. C. Tan, Q. Li, W. Mao, "Finding popular categories for RFID tags," Proc. ACM Mobihoc 2008, Hong Kong, China, pp. 159-168.

[7] M. A. Bonuccelli, F. Lonetti, F. Martelli, "Tree slotted ALOHA: a new protocol for tag identification in RFID networks," Proc, WOWMOM 06, 2006.

[8] M. S. Kodialam and T. Nandagopal, "Fast and reliable estimation schemes in RFID systems," Proc. ACM MOBICOM'06, pp. 322-333.

[9] H. Han, B. Sheng, C. C. Tan, Q. Li, W. Mao, and S. Lu, "Counting RFID tags efficiently and anonymously," Proc. INFOCOM, 2010, pp.1028-1036.

[10] C. Qian, H. Ngan, and Y. Liu, "Cardinality estimation for large-scale RFID systems," Proc. PerCom, 2008, pp.30-39.

[11] S.-R. Lee, S.-D. Joo, and C.-W. Lee. "An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification," Proc. MOBIQUITOUS 05, 2005.

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

[13] S. Ren, Q. Li, H. Wang, X. Chen, and X. Zhang, "Analyzing object detection quality under probabilistic coverage in sensor networks," Proc. IWQoS'05, Hermann Meer and Nina Bhatti (Eds.). Springer-Verlag, Berlin, Heidelberg, pp. 107-122.

[14] H. Wang, C. C. Tan and Q. Li. "Snoogle: a search engine for the physical world," Proc. INFOCOM 2008. Phoenix, AZ, USA, pp. 1382-1390.

[15] D. Xuan, R. Bettati and W. Zhao, "A gateway-based defense system for distributed DOS attacks in high-speed networks," IEEE Transactions on Systems, Man, and Cybernetics (2002).

[16] K. Xing, X. Cheng and M. Ding. "Safety warning based on highway sensor networks," Proc. IEEE WCNC 2005. New Orleans, LA, March, pp. 2355-2361.

[17] R. Clauberg. "RFID and sensor networks," Proc. of RFID Workshop, University of St. Gallen 2004. Switzerland, September

[18] S. Madden, M. Franklin, J. Hellerstein and W. Hong, "TAG: A tiny aggregation service for ad-hoc sensor networks," SIGOPS Oper. Syst. Rev. 36, SI (December 2002), pp. 131146.

[19] N. Shrivastava, C. Buragohain, D. Agrawal, and S. Suri. "Medians and beyond: new aggregation techniques for sensor networks," Proc. of SenSys 2004. ACM, New York, NY, USA, pp. 239-249.

[20] S. Nath, P. B. Gibbons, S. Seshan, and Z. Anderson. "Synopsis diffusion for robust aggregation in sensor networks," ACM Trans. Sen. Netw. 4, 2, Article 7.

[21] S. Roy, S. Setia, and S. Jajodia. "Attack-resilient hierarchical data aggregation in sensor networks," Proc. of ACM SASN '06. ACM, New York, NY, USA, pp. 71-82.

[22] B. Patt-Shamir, "A Note on efficient aggregate queries in sensor networks," Proc. PODC, 2004, pp.283-289.

[23] X. Bai, C. Zhang, D. Xuan, J. Teng, and W. Jia. "Low-connectivity and full-coverage three dimensional wireless sensor networks," Proc. of ACM MobiHoc '09. ACM, New York, NY, USA, pp. 145-154.

[24] M. Ding, F. Liu, A. Thaeler, D. Chen, and X. Cheng. "Fault-tolerant target localization in sensor networks," EURASIP J. Wirel. Commun. Netw. 2007, 1 (January 2007). Hindawi Publishing Corp. New York, NY, United States, pp. 19-19.

[25] B. Sheng, Q. Li, and W. Mao. "Efficient continuous scanning in RFID systems," Proc. of INFOCOM 2010. San Diego, CA, USA, pp. 1010-1018.

[26] H. Wang, B. Sheng, C. C. Tan and Q. Li. "WM-ECC: an Elliptic Curve Cryptography suite on sensor motes," Computer Science, College of William and Mary, Williamsburg, VA, Tech. Rep. WM-CS-2007-11, 2007.