An Enhanced Query Tree Protocol for RFID Tag Collision Resolution with Progressive Population Estimation

Xinqing Yan and Guiliang Zhu
School of Information Engineering
North China University of Water Conservancy and Electric Power
Zhengzhou, 450011, China
E-mail: {yanxq, zgl500}@ncwu.edu.cn

***Abstract*—Rapid and reliable resolution of the collisions caused by multiple RFID tags during identification is a key factor for the proliferation application of RFID system. In this paper, an enhanced query tree protocol for memoryless passive RFID tags is proposed, in which the binary identifiers stored in the RFID tags and the binary query strings broadcasted by the RFID reader are mapped to real numbers in the interval [0, 1). During identification, according to the amount of RFID tags already identified and the value which the binary query string adopted by the protocol in the last frame maps to, the overall tag population is estimated, and an optimal binary query string for the protocol to adopt and the reader to broadcast in the next frame is calculated. Evaluation verifies that this protocol outperforms other query tree protocols regarding to the metrics of throughput and time delay.**

***Index Terms*—Collision resolution, progressive estimation, query tree protocol, RFID.**

I. INTRODUCTION

RFID (Radio Frequency Identification) enables the rapid tracking and tracing of multiple objects through contactless identification of the unique digital identifiers which are stored in tags attached to each item, bridges the gap between the physical item world with the virtual digital space, and is regarded as the main component of the upcoming pervasive computing[1] and backbone of the "Internet of things"[2]. Nowadays, RFID systems have been widely deployed in a diversity applications.

One of key issues that affect the proliferation adoption of RFID system is the collision caused by that multiple RFID tags try to transmit their data to the RFID reader simultaneously through the air interface, a shared wireless communication channel between the RFID reader and tags[3]. The modulated waveforms from multiple RFID tags will interfere in the air interface, and result in that what the reader can get is only a collision signal but no useful information. Because of collisions, RFID tag identification process suffers from low accuracy[4] and long time delay.

Proposed RFID tag collision resolution protocols can be categorized as the probabilistic frame slotted ALOHA based protocols, the deterministic splitting tree based protocols[5] and some hybrid protocols[6]. The frame slotted ALOHA based and hybrid protocols suffer from the "tag starvation" problem, in which case some RFID tags may not be recognized in quite a long time due to the continuous collisions and randomness introduced in the protocols, and will not be discussed any more in this paper.

The splitting tree based protocols recursively divide the set of RFID tags which cause collision into subsets until in each subset, there is at most one tag, so all confliction can be resolved and all RFID tags can be identified successfully[7], [8]. The query tree protocols, a category of the variant of the splitting tree based protocols, perform the subset division through asking the RFID reader to broadcast some binary query strings, and are especially suitable to resolve the collisions caused by memoryless passive RFID tags. Due to their simplicity and ease of implementation, some query tree RFID tags collision resolution protocols have been adopted as international or industrial standards.

Rapid and reliable tag collision resolution protocols have always been the pursuit of researchers. In this paper, an enhanced query tree protocol is suggested, in which a map function is devised to map the unique binary identifier of RFID tags and binary query string broadcasted by the RFID reader to real numbers in the

978-1-4244-5113-5/09/$25.00 ©2009 IEEE

interval [0, 1), and during identification, after some tags have been identified, progressively, based on the amount of tags already identified and the mapped value of the binary query string adopted, the overall population of tags within the vicinity of the reader is estimated, the optimal binary query string for the reader to broadcast in the next frame is calculated. This process is performed repeatedly until all collisions are resolved and all RFID tags are identified successfully. Theoretical analysis and numerical simulation verify that this enhanced protocol outperforms other query tree protocols regarding to the metrics of throughput and delay.

II. The Enhanced Query tree protocol

A. The Query Tree Protocols

In the query tree protocols, the collision caused by two or more RFID tags is resolved through a series of frames, and each frame is consisted of a command slot and a data slot. In frame *i*, the RFID reader broadcasts a message containing a binary query string q_i (with initial value $q_1 = \varepsilon$) in the command slot to all RFID tags in its vicinity. Each RFID tag, upon receiving this message and decoding the binary query string q_i, compares q_i with the binary identifier it stores. If its binary identifier starts with q_i, the RFID tag transmits its identifier back to the reader in the data slot, otherwise, it keeps silent and does not respond in the frame.

If no RFID tag answer in the data slot, the RFID reader can detect the idle data slot and the protocol continues with other binary query string. Otherwise if one RFID tag answers in the data slot, no collision occurs, and the RFID reader can receive the data signals, decode the identifier successfully, and the RFID tag is identified. If two or more RFID tags answer in the data slot, collision will occur, the RFID reader can detect the collision and these tags will be divided into subsets in the following frames for further collision resolution.

When collision occurs, the query tree protocol appends 0 to the binary query string q_i, asks the RFID reader to broadcast q_i0 in the next frame and resolves the collision caused by RFID tags whose binary identifiers start with q_i0 in the following frames. After all collisions caused by these RFID tags are resolved and all these tags are identified, the query tree protocol appends 1 to the binary query string q_i, starts to identify these RFID tags whose binary identifiers start with q_i1. In such a way, the set of RFID tags which cause collision is divided into 2 subsets by the binary query strings broadcasted by the RFID reader. This process is performed repeatedly and recursively until no collision occurs in the final subsets.

Fig.1 shows the collision resolution process of RFID tags with binary identifiers *010, 011, 100* and *111*.

Slot ID	1	2	3	4	5	6	7	8	9
Query string	ϵ	0	00	01	010	011	1	10	11
Tags Answer	010 011 100 111	010 011		010 011	010	011	100 111	100	111
Slot State	Coll	Coll	Idle	Coll	Succ	Succ	Coll	Succ	Succ

Fig. 1. frames used for the identification of RFID tags with binary identifier 010, 011, 100 and 111

The working process of this protocol is like a binary search tree, hence the binary query tree protocol is named.

To divide t RFID tags into 2 subsets, the probability that a subset contains k ($k \geq 0$ and $k \leq t$) tags can be calculated according to the Bernoulli distribution as

$$B(k) = \frac{1}{2^t}\binom{t}{k} \tag{1}$$

For the standard binary query tree protocol to resolve the collision caused by t RFID tags, the number of frame L_t needed in an identification cycle can be calculated recursively with

$$L_t = \begin{cases} 1 & t \leq 1 \\ 1 + 2\sum_{i=0}^{t} B(i)L_i & t \geq 2 \end{cases} \tag{2}$$

Although L_t appears in both sides of the equation, through mathematical transformation, L_t can be shifted to the left side and calculated recursively.

The throughput of the RFID tag collision resolution protocol is calculated as $\frac{t}{L_t} \times 100\%$. It is reported that the stable throughput of this standard binary query tree protocol is 34.6%[10].

It can be observed that if the frame adopting binary query string q_i results in collision, and the following frame adopting the binary query string q_i0 results in idle, the broadcast of the binary query string q_i1 can be predetermined to result in collision and can be skipped to facilitate the identification, such as the fourth slot in Fig.1. Binary query strings q_i10 can be broadcasted directly following the frame with binary query string q_i0. The number of frames needed by this modified binary query tree protocol in resolving the collision caused by t RFID tags can be calculated as

$$L_t = \begin{cases} 1 & t \leq 1 \\ 1 + 2\sum_{i=0}^{t} B(i)L_i - B(0) & t \geq 2 \end{cases} \tag{3}$$

The stable throughput of this modified binary query tree protocol is 37.5% [10], near 3% improvement over the standard binary query tree protocol.

If the population of RFID tags is known or can be estimated in advance, these tags can be divided into multiple subsets initially using some deliberately calculated binary query strings, some intermediate frames which probably result in collision can be skipped, and the overall RFID tag collision resolution process can be facilitated. However, in typical RFID systems and applications, the population of RFID tags is usually unknown in advance.

B. Estimation of the RFID Tag Population

Here we propose a novel method for progressive estimation of RFID tag population. Firstly, we define a function which maps the binary identifiers stored in RFID tags and the binary query strings broadcasted by the RFID reader to real values in the interval [0, 1).

For any binary string $a_1 \ldots a_n$, the map function f is defined as

$$f(a_1 \ldots a_n) = \sum_{i=1}^{n} \frac{a_i}{2^i} \tag{4}$$

Function f maps every unique binary identifier of RFID tag to a unique value in the interval [0, 1). However, multiple binary query strings broadcasted by the RFID reader may be mapped to a same value, for example, the query strings 01, 010, 0100 are all mapped to 0.25.

To finish the definition of the map function, especially for the empty query string ε, we define $f(\varepsilon) = 0$.

A RFID tag is said to be contained in an interval $[p,\ p+x)$ if its binary identifier *a* satisfies $p \leq f(a) < p+x$.

During collision resolution, if in a frame, the binary query string to be broadcasted is q_i, and $f(q_i) = p$, then the collisions caused by RFID tags contained in the interval [0, p) should have been resolved, and all these RFID tags should have already been identified successfully.

Suppose that there are *t* RFID tags in the vicinity of the RFID reader, according to the Bernoulli and Poisson distributions, the probability that k RFID tags are contained in the interval [0, *p*) can be calculated as

$$P(k,p) = \binom{t}{k} (p)^k (1-p)^{t-k} \tag{5}$$

If t is unknown and $k > 0$, according to the *Maximum Likelihood* theorem, the RFID tag population estimation $\hat{t}$ should be most likely estimated with

$$\hat{t} = MaxLikeli\ p(k) = \frac{k}{p} \tag{6}$$

For such estimation of the RFID tag population, we have the following important properties:

$$E\left|\frac{\hat{t}}{t}\right| = 1 \tag{7}$$

$$Var\left\|\frac{\hat{t}}{t}\right\| = \frac{t}{p} - t \tag{8}$$

and for any $\delta > 0$, we have

$$P\left(\left|\frac{\hat{t}}{t} - 1\right| \geq \delta\ |t\right) \leq \frac{1-p}{\delta^2 tp} \tag{9}$$

From (9), it can be found that as p increases and approaches to 1, the estimated RFID tag population $\hat{t}$ approaches the actual tag population t.

C. Calculation of the Next Subinterval to Resolve

Once the resolution of interval [0, *p*) results in k $(k > 0)$ RFID tags identified, and the overall RFID tag population is estimated as $\hat{t} = \frac{k}{p}$, the the next thing for the enhanced query tree protocol to do is to calculate and find out an appropriate real number x ($x > 0$ and $p + x < 1$) so that the next subinterval $[p,\ p+x)$ can be resolved with maximum throughput.

For any x ($x > 0$ and $p + x < 1$), the probabilities that one query to resolve the subinterval $[p,\ p+x)$ results in idle (the subinterval contains no RFID tag), successful (the subinterval contains exactly one RFID tag) and collision (the subinterval contains two or more RFID tags) can be calculated with

$$P_{idle}(\triangle) = e^{-\triangle} \tag{10}$$

$$P_{succ}(\triangle) = \triangle e^{-\triangle} \tag{11}$$

$$P_{coll}(\triangle) = 1 - e^{-\triangle} - \triangle e^{-\triangle} \tag{12}$$

where $\triangle = x\hat{t}$.

Fig.2 depicts the corresponding values of P_{idle}, P_{succ} and P_{coll} as $\triangle$ changes from 0 to 2.

To find the appropriate *x*, a natural way is to maximize P_{succ}, so that the probability that the subinterval $[p,\ p+x)$ contains exactly one tag and can be resolved with one query is also maximized. This leads to $\triangle = 1$, $P_{succ}(\triangle) = 0.368$, and $x = \frac{\triangle}{\hat{t}} = \frac{p}{k}$. Adopting this x, the resolution of the subinterval $[p,\ p+x)$ can achieve the throughput 36.8%.

However, it can also be observed in Fig.2 that with the same probability 36.8%, this subinterval may contain no

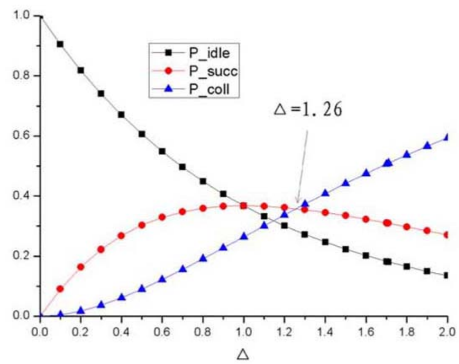

Fig. 2. The values of $P_{idle}, P_{succ}, P_{coll}$ as △ varies from 0 to 2

RFID tag and the corresponding query will result in idle. Idle query frame will reduce the overall throughput of the protocol significantly. So for the protocol to achieve maximum throughput, we wish to find an optimal value △, so that the function $g(\triangle)$ to calculate the frame per unit needed to resolve the corresponding subinterval is minimized. The function $g(\triangle)$ can be calculated according to the Poisson distribution with

$$g(\triangle) = \frac{\sum_{k=0}^{\infty} P(k,\triangle)L_k}{\triangle} \approx \sum_{k=0}^{\infty} \frac{\triangle^{k-1}}{k!} e^{-\triangle} L_k \quad (13)$$

Suppose that the modified binary query tree protocol is used to resolve the collision occurred in the subinterval $[p,\ p+x)$, and L_k can be calculated using (3), numerical evaluation shows that when $\triangle = 1.26$, $g(\triangle)$ is minimized to be about 2.16.

It is worthy to be noticed that this △ also satisfy the following statement.

$$max(P_{succ}(\triangle) - P_{idle}(\triangle)) \text{ while } P_{succ}(\triangle) \geq P_{coll}(\triangle) \quad (14)$$

as depicted in Fig.2.

For such △, the corresponding x for next optimal subinterval $[p,\ p+x)$ which the protocol is to resolve in the next frame should be calculated with

$$x = \frac{1.26}{\hat{t}} = \frac{1.26p}{k} \quad (15)$$

D. Calculation of the Next Binary Query String

From the above discussion, we can draw the conclusion that if after interval $[0,\ p)$ has been resolved, and k $(k > 0)$ RFID tags have been identified, the optimal subinterval for the protocol to resolve in the next frame should be $[p,\ p+\frac{1.26p}{k})$.

However, due to the extreme constraint on computation put on the them, most RFID tags are unable to map their identifiers to real numbers or to make real number comparison. This subinterval $[p,\ p+\frac{1.26p}{k})$ must be converted to a binary query string in the protocol so that the RFID reader can broadcast in the command slot of next frame. But quite often for all binary strings mapped in the subinterval $[p,\ p+\frac{1.26p}{k})$,there may not exist a non-empty common binary prefix string. For example, for the subinterval [0.250, 0.725), there does no exist a non-empty common binary prefix string.

To overcome this issue, we suggest to divide the whole interval [0,1) into 2^m subintervals, where m is calculated using

$$m = round(log_2^{\frac{\hat{t}}{1.26}}) = round(log_2^{\frac{k}{1.26p}}) \quad (16)$$

With such division, we get a series of subintervals $[0, \frac{1}{2^m}), \ldots, [\frac{j}{2^m}, \frac{j+1}{2^m}), \ldots, [1-\frac{1}{2^m}, 1)$. For all binary strings mapped in any subinterval $[\frac{i}{2^m}, \frac{i+1}{2^m})$,there exists a non-empty common binary prefix string.

If interval [0, p) has been resolved with one or more RFID tags identified, and the corresponding series of subinterval have been divided, a subinterval $[\frac{i}{2^m}, \frac{i+1}{2^m})$ can be found, where $\frac{i}{2^m} \leq p < \frac{i+1}{2^m}$. The maximum-lengthed common binary prefix string for the subinterval $[p, \frac{i+1}{2^m})$ should be adopted by the protocol as the binary query string for the RFID reader to broadcast in the next frame.

E. The Enhanced Query Tree Protocol

Based on the discussion presented above, the enhanced query tree protocol with progressive tag population estimation is proposed, in which after one or more RFID tags have been identified using the modified binary query protocol, this protocols maps the query string broadcasted in the last frame to a real value using function f defined in (4), and progressively, when an interval $[0,\ p)$ has been resolved successfully, the overall RFID tag population $\hat{t}$ is estimated, the whole interval [0, 1) is divided into a corresponding series of subintervals, the next optimal subinterval $[p, \frac{i}{2^m})$ is located, and the maximum-lengthed common binary prefix string of this subinterval is adopted in the protocol for the RFID reader to broadcast in the next frame.

If the broadcast of this binary query string results in idle or successful, the same process is repeated. Otherwise, collision occurs in the subinterval $[p, \frac{i}{2^m})$, the protocol divides this subinterval into two halves, starts to copy with the first half $[p, \frac{i}{2^{m+1}})$, and asks the RFID reader to broadcast the maximum-lengthed common binary prefix string of this first half subinterval in the next frame.

This process is performed repeatedly and progressively until a frame whose binary query string is consisted of a series of binary digits “1” results in idle or successful. In such case, all collisions caused by RFID tags should have been resolved and all RFID tags should have been identified successfully.

III. Performance Evaluation

The performance of various RFID tag collision resolution protocol is often evaluated regarding to the metrics of throughput and time delay. Throughput reflects the efficient use of the wireless communication channel as well as the efficiency of the protocol.

To evaluate the performance of different query tree based protocols, we compare the throughput and time delay of the standard and modified binary query tree protocol as well as the enhanced query tree protocol proposed in this paper.

The throughput and time delay of the standard and modified binary query tree protocols can be calculated directly, but for the enhanced query protocol with progressive tag population estimation, no direct calculation method has been found, so instead, we perform numerical simulation to measure its performance.

In the numerical simulation, 100 different data sets are used, and in each data set there are 1000 binary string groups representing 1 to 1000 RFID tag identifers. The enhanced query tree protocol is implemented and the conflict resolution process is simulated for each data set. The number of frames, amounts of bits broadcasted by the RFID reader and transmitted by the RFID rags in resolving the collision caused by 1 to 1000 RFID tags in each data set are recorded and averaged finally for the overall 100 data sets.

A. Throughput of Protocols

Fig.3 shows the throughput of these query tree protocols achieved through theoretical analysis and numerical simulation. It can be seen that the throughput of the enhanced query tree is the highest, and as the RFID tag population exceeds 50, the throughput of the standard binary query tree protocol is stabilized to 34.6%, and that of the modified binary query tree protocol is stabilized to 37.5%, as we have expected. But the throughput of the enhanced query tree protocol varies around 44.9%, with minimum 43.1% and maximum 46.5%. This variance may be due to that the number of subintervals to for the whole interval $[0,\ 1)$ to divide must be a mean of 2 in order to find the optimal binary query string for RFID reader to broadcast in the next frame.

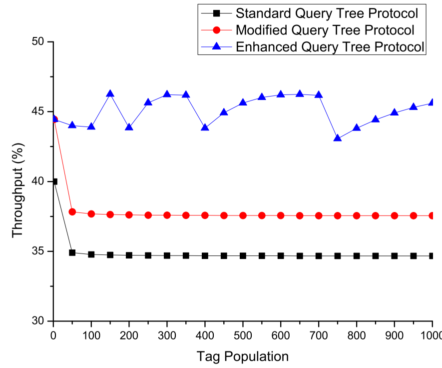

Fig. 3. Throughput of the different query tree protocols.

B. Time Delay

The data transferred in an identification cycle can be divided into two parts: the bits contained in the binary query string broadcasted by the RFID reader and the bits contained in the answers transmitted by the RFID tags.

For the standard binary query tree protocol, to resolve the collision caused by t RFID tags, the amount of bits contained in the query strings broadcasted by the RFID reader in the command slots with in an identification can be calculated recursively as

$$DC_t = \begin{cases} 0 & t \leq 1 \\ 2 + 2\sum_{i=0}^{t} B(i) DC_i & t \geq 2 \end{cases} \tag{17}$$

and the amount of bits contained in the answers transmitted by all RFID tags in the data slots within an identification cycle can be calculated as

$$DD_t = \begin{cases} 0 & t = 0 \\ t_d & t = 1 \\ t_d t + 2\sum_{i=0}^{t} B(i) DD_i & t \geq 2 \end{cases} \tag{18}$$

where t_d is the amount of bits transmitted in an answering by a RFID tag, including the lengths of its binary identifier and the corresponding CRC code.

For the modified binary query tree protocol, the corresponding DC_t and DD_t can be calculated using

$$DC_t = \begin{cases} 0 & t \leq 1 \\ 2 + 2\sum_{i=0}^{t} B(i) DC_i - B(0) & t \geq 2 \end{cases} \tag{19}$$

and

$$DD_t = \begin{cases} 0 & t = 0 \\ t_d & t = 1 \\ t_d t + 2\sum_{i=0}^{t} B(i) DD_i - B(0) t_d t & t \geq 2 \end{cases} \tag{20}$$

For the enhanced query tree protocol proposed in this paper, the number of bits broadcasted by the RFID reader DC_t and that transmitted by all RFID tags DD_t in an identification cycle are achieved through numerical simulation because that no direct calculation method has been found, and numeric simulation.

Suppose that the time needed for the RFID reader to broadcast a bit is τ_c, and that for a RFID tag to transmit a bit is τ_d, the time needed for the protocols to resolve the collision caused by t tags can be calculated with

$$T = DD_t\tau_c + L_t t_d \tau_d \tag{21}$$

Without loss of generality, the 915MHz EPCglobal class 1 RFID system is taken as an example in this paper for performance evaluation and comparison regarding to different query tree protocols, in which $t_d = 120$ bits, including 96bit binary identifier stored in RFID tag identifier and 24bit CRC code, $\tau_c = 12.5\mu s$, and $\tau_d = 4\mu s$. Similar evaluation can also be performed to other kind of RFID systems using changed parameters.

Fig.4 shows the time needed in an identification cycle for these query tree protocols.

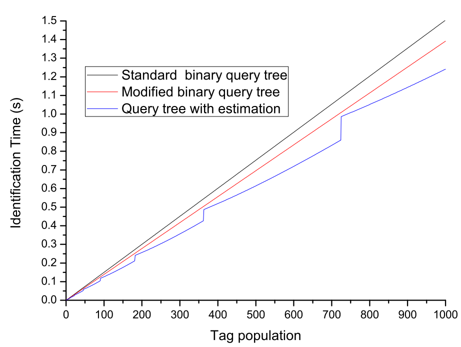

Fig. 4. Time delay of the query tree protocols.

From Fig.4, the conclusion can be drawn that the time needed by these protocols is almost linearly related with the RFID tag population, and the enhanced query tree protocol also outperforms other two protocols.

IV. Conclusion and Future Works

RFID technology holds the promise to enable human beings to monitor the world with fine granularity, however efficient and effective RFID tag collision resolution protocol is a necessity to make this promise a reality.

In this paper, an enhanced query tree protocol is presented, in which through the definition of a map function, during collision resolution, based on the number of RFID tags already identified and the binary query string broadcasted by the RFID reader in the previous frame, the overall population of RFID tags is estimated, and an optimal binary query string for the RFID reader to broadcast in the next frame is calculated. This process is performed repeatedly and progressively until all RFID tags are identified successfully. Theoretical analysis and numeric evaluation verify that this enhanced query protocol can achieve better performance than the standard and modified binary query tree protocols regarding to the metrics of throughput and time delay.

By combining the method introduced in this paper and other recent works, such as that proposed in [11], we think better result may be achieved in future research.

Acknowledgement

The research of this paper is partly sponsored by the High Talent Starting Research Project of North China University of Water Conservancy and Electric Power.

References

[1] V. Stanford, *Pervasive Computing Goes the Last Hundred Feet with RFID Systems*, IEEE Pervasive Computing, Vol. 2(2). pp.1-14,2003

[2] D.W Engels,J. Foley, and J. Waldrop, et al., The networked physical world: an automated identication architecture. In Proceedings of the Second IEEE Workshop on Internet Applications. pp.76-77. 2001.

[3] N. C. Wu, M. A. Nystrom and T. R. Lin, et al., *Challenges to global RFID adoption*, Technovation, Vol. 26(12), pp. 1317-1323, 2006.

[4] S.R. Jeffery, M.J. Franklin, and M. Gaorfalakis, *An adaptive RFID middleware for supporting metaphysical data independence*. The VLDB Journal, 2008. 17(2): pp. 265-289.

[5] D. H. Shih, P. L. Sun and D. C. Yen, et al., *Taxonomy and survey of RFID anti-collision protocols*, Computer Communications, Vol. 29(1), pp.2150-2166, 2006.

[6] M. A. Bonuccelli, F. Lonetti, and Martelli. P. F. Tree Slotted Aloha : a New Protocol for Tag Identification in RFID Networks. *Proceedings of the 2006 International Symposium on World of Wireless, Mobile and Multimedia Networks.* 2006. pp.603-608

[7] S. H. Kim and P. Park, *An Efficient Tree-Based Tag Anti-Collision Protocol for RFID Systems*, IEEE Communications Letters, Vol. 11(5), pp.449-451, 2007

[8] J. Myung, W. Lee, and J. Srivastava, *Adaptive binary splitting for efficient RFID tag anti-collision*, IEEE Communications Letters, Vol. 10(3), pp.144-146, 2006.

[9] M. A. Kaplan and E. Gulko, *Analytic Properties of Multiple-Access Trees*, IEEE Transactions on Information Theory, Vol. IT-31(2), pp.255-263, 1985.

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

[11] B. Sheng, C. C. Tan, Q. Li, and W. Mao. Finding Popular Categories for RFID Tags. *Proceedings of the 9th ACM International Symposium on Mobile Ad hoc Networking and Computing*, pp. 159-168, 2008