Using Hybrid Query Tree to Cope with Capture Effect in RFID Tag Identification

Milad HajMirzaei[1], Masoud Ghiasbeigi[2]
[1] CSE and IT Department, School of Electrical and Computer Engineering, Shiraz University International Division, Shiraz, Iran
[2] Department of Computer Engineering, Mahalat Branch, Islamic Azad University, Mahalat, Iran
hmirzaei.m@gmail.com

***Abstract*—Tag collision is one of the important issues in RFID systems. Many algorithms were proposed to address this issue. One of these algorithms is Query Tree (QT) which is an effective method. In addition, RFID suffers from Capture Effect (CE). CE occurs when a reader identifies one tag in the presence of a collision. We consider this as a bad phenomenon for QT, because under CE reader will not identify all of collided tags. Besides, CE is good phenomenon for some algorithms like Dynamic Framed Slotted Aloha (DFSA), because it can identify one tag even in collision slots. So we combine QT and DFSA to improve the QT performance, then we evaluate our proposed algorithm, called Hybrid QT, to show that it outperforms other similar algorithms.**

***Index Terms*— algorithm, anti collision, Collision, RFID**

I. Introduction

The RFID (Radio Frequency Identification) uses communicated radio frequency to retrieve data. One of the RFID issues is tag collision. When more than one tag transmits its ID to the reader simultaneously, the collision occurs. RFID suffers from incorrect received signals due to collision. It is reported that in typical RFID deployments, the tag`s read rate is usually about 60-70% [1]. To address this issue some algorithms have been proposed, called Tag anti-collision. We can classify them to 3 groups: Aloha-based algorithms, Tree-based algorithms and Counter-based algorithms.

Rest of this paper is organized as follows: Section 2 overviews QT and DFSA algorithms. In Section 3 we discuss capture effect problem in RFID systems and how this effects on QT and DFSA, then we propose a hybrid approach to cope with capture effect in Section 4. Evaluation and conclusion are in Section 5 and 6, respectively.

II. Tag Anti-collision Algorithms

To identify tags within the interrogation zone, a reader sends a request to ask tags to send back their IDs. When multiple tags respond to the request simultaneously, collision occurs and reader could not identify any tag properly. This is called the tag collision problem. Therefore, tag collisions are resolved by the reader utilizing techniques collectively known as tag anti-collisions schemes.

A. Query Tree (QT)

In this approach [2], a reader first broadcasts a request bit string S to tags. The tags, their prefix IDs match with S, respond to the reader by sending back the remaining bits of their IDs. If only one tag responds to reader, the tag is identified correctly otherwise if more than one tag respond simultaneously to reader, the collision occurs. In this case, reader sends a longer bit string that has one bit more than the last string. Usually reader appends 0 or 1 to S string that is S0 or S1 (almost first use 0). Tags are divided into two subgroups: tags started with S0 and S1. This will be repeated until only one tag matches with S string to identify correctly. For instance we use QT to identify three tags A, B, C in Figure1.

Authors in [2] define time of identification process of Query Tree as the number of queries needed to identify all n tags in interrogation zone of a reader. And then use $T_{QT}(n)$ to define worst case time complexity as follows:

$$T_{QT}(n) = n(k + 2 - \log n) \tag{1}$$

Where n is the number of tags and k denotes the tag ID`s length (bits).

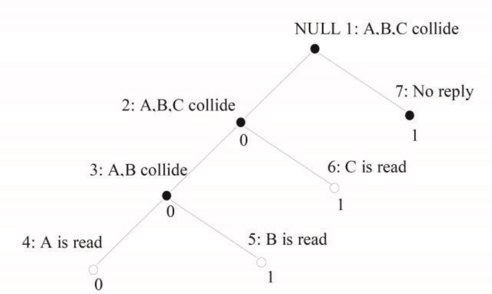

Figure 1: QT Process for three tags. Tags are read in white circles. A=00000, B=00101, C=01001

B. Dynamic Framed slotted Aloha (DFSA)

In Framed Slotted Aloha [3], reader sends a frame to all tags as a request to receive the tag IDs. Each frame is divided into several time slots. Every tag when receives the request, randomly choose a time slot and sends back its ID in that time slot as a reply. If just one tag responds in a slot, will be identified by reader, otherwise if more than one tag responds in one slot, collision occurs and collided tag will choose another slot in next frame same as before. The throughput of FSA decreases when the number of tags is unknown. So to address this issue Dynamic FSA was proposed in [4]. In this scheme frame size will be dynamically changed during the identification process. If collision occurs in frame slots, the reader increases the frame size to identify more tags. As long as tags are identified, reader will decrease frame size and so on. Let L be the variable representing the frame size previously used by the reader in previous read cycle. In [5] the DFSA throughput (S) is equal to probability of read and is given as:

$$s = \left(\frac{1}{L}\right)\left(1-\frac{1}{L}\right)^{n-1} \tag{2}$$

III. Capture Effect

In some algorithms like QT, when the reader identifies one tag x in a node, the reader assume there is not any tag with same ID prefix. In other words if QT identifies one tag in a node, then it never expands that node. This manner of reader in QT, neglects the capture effect at all. Recent studies[6,7] show that in real wireless network environment when collision occurs, the reader can identifies one tag if signal to interference (SINR) and noise ratio are high enough. Under capture effect, single tag can be identified through a reader in the midst of a collision and therefore other collided tags will be left unidentified.

Capture effect has two faces for two types of tag anti-collision algorithms. The good face of capture effect appears when we use Aloha based algorithms like DFSA because in collision cases reader can detect and identify one tag. And bad face of capture effect appears when we use Tree based algorithms like QT, because when reader sends query string S to the tags, although collision occurs, reader detects one tag and assumes there is not any tag whose ID prefix matches with S and does not send query S again.

Some schemes were proposed to modify Tree based algorithm to cope with capture effect in literature. In [8,9] the Multi-Round Collision Tree (MRCT) and General Binary Tree (GBT), both repeat identification cycle to detect unidentified tags until detection of all tags. In worst case when always CE happens, this approach will repeat $n+1$ times whole the identification cycle to detect all tags. In [10] Viktor proposes GQT1 and GQT2. GQT1 makes 2 changes to cope with CE. First, the prefix always is lengthened if there is a response. Second, the reader sends ACK signal to the tags when they are identified. Also GQT2 makes a change to GQT1. When one tag is decoded by a reader, reader rebroadcasts the same prefix to detect some unidentified tags, if exist. In this approach in every situation (worst or best), in addition to regular queries needed to detect all tags, the reader sends $n+1$ more queries to ensure that no unidentified tags are remained because of CE.

IV. Hybrid Query Tree (HQT)

As mentioned before, Query Tree algorithm suffers from Capture Effect (CE). So we use DFSA in Query Tree. This approach is called Hybrid Query Tree which comprises two phases, QT phase and DFSA phase. Here first the reader uses QT to identify all tags, every tag that is identified will receive an ACK from reader after identification. When QT is processing, CE may occurs and cause to leave some tags unidentified. Hence after QT phase has ended its process, DFSA phase begins to send a request to detect whether there is any tag unidentified. In this case all tags which have not received ACK (is not detected by QT phase because of CE) will respond to reader. Same as QT, in DFSA phase the reader sends ACK to tags which are identified. Here we should notice that we have added Sending Acknowledge message method to basic DFSA to use it in our scheme.

To illustrate performance of Hybrid QT we use 5 tags A=0011, B=1001, C=1100, D=0111 and E=1010. Figure 2 depicts two phases of Hybrid QT. In the first phase, QT tries to identify all tags but capture effect occurs in two points. First in step 2, tags A and D collide but reader identifies tag A and sends an ACK to A. Second in step 4, tag B is identified because of CE, and then B receives an ACK. Then QT identifies tag C in step 5 and finally ends its process with assumption that it has detected and identified all tags. But two tags D and E are remained. In the second phase, DFSA starts its process by sending a request to all tags which have not received ACK. In step 6 each tag receives the request and randomly chooses a slot number to respond in it. Tags E and D will be identified in step 6. In step 7, reader repeats its request and receives no response. If in DFSA phase reader receives no response, it assumes all tags were identified and ends the identification cycle.

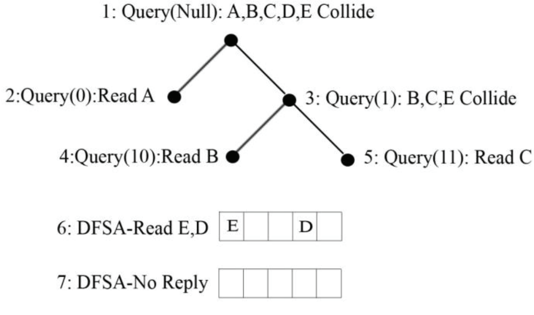

Figure 2: Hybrid QT identification process for 5 tags =0011, B=1001, C=1100, D=0111 and E=1010

The average capture effect probability in collision cases is denoted by α (0<α<1). Here we have two extreme cases: upper bound time complexity is when α=0, which means capture effect never takes place. In this case, algorithm just uses QT phase to identify all tags. QT upper bound time complexity in worst case is mentioned in equation (1). Then DFSA phase just sends 1 request to ensure that no tag is remained. Assuming an ACK command takes half of time unit. Time needed to sending the ACK command to all n tags is $n/2$. So the upper bound for $n \geq 2$ is given by

$$T_{UP} = T_{QT}(n) + \frac{n}{2} + 1 \tag{3}$$

In the best case when α=1 which means capture effect always happen, the QT can just identify one tag in the root node. Then the QT assumes that just there is one tag in reader interrogation zone which is identified and then ends the process. After ending the QT phase, the DFSA phase start to send request to all *n-1* remained tags. In [11] the optimum frame size under CE is given by

$$L_{opt} = \alpha + (1-\alpha)n \tag{4}$$

In best case when *α=1* the optimum frame size (*L*) is 1. So in the second phase if we have *n-1* tags and frame size is 1 and ACKing needs half of time unit, the lower bound for *n≥2* is given by

$$T_{Low} = n + \frac{n}{2} \tag{5}$$

With combination of two bounds, we have:

$$T_{HQT}(n) \in \left[n + \frac{n}{2}, T_{QT}(n) + \frac{n}{2} + 1 \right] \tag{6}$$

Replacing T_{QT} in (1), we have,

$$T_{HQT}(n) \in \left[\frac{3n}{2}, n(k + \frac{5}{2} - \log n) + 1 \right] \tag{7}$$

Hybrid QT flow chart is depicted in Figure 3

V. EVALUATION

We used C# to study the performance of Hybrid QT in 3 steps. First in Figure 4 we compare Hybrid QT and basic Query Tree to show the flaw of QT in presence of CE. This evaluation based on the number of unidentified tags. Second in Figure 5 we compare performance of hybrid QT with different numbers of tags. Third we compare Hybrid QT, GQT1, GQT2, MRCT and GBT for different values of α in Figure 6. Second and third evaluations are based on time complexity which is equal to steps needed to identify all tags. In these simulations each tag has a 96-bit ID which is unique and is chosen randomly. We also assume a noise free channel and packet loss are due to collision only. Each simulation is repeated 100 times and the average value is used.

As a result, Figure 4 shows when *α>0* QT could not detect all tags successfully and by increasing α, the QT performance will be decreased.

Figure 5 shows when *α=0* , all tags will be identified by QT phase of Hybrid QT scheme but by increasing α, DFSA phase will be more active than before and time of identification will be decreased. The best time is achieved when *α=1* and approximately all tags (*n-1*) are identified by DFSA phase. Figure 5 also shows that the number of tags is not important when RFID faces capture effect.

Figure 6 shows that hybrid QT outperforms GQT1, GQT2, MRCT and GBT regardless of the value of α. In this evaluation when *α=0* , three algorithms Hybrid QT, MRC and GBT have same performance, but when α is increased we can see benefits of Hybrid QT obviously.

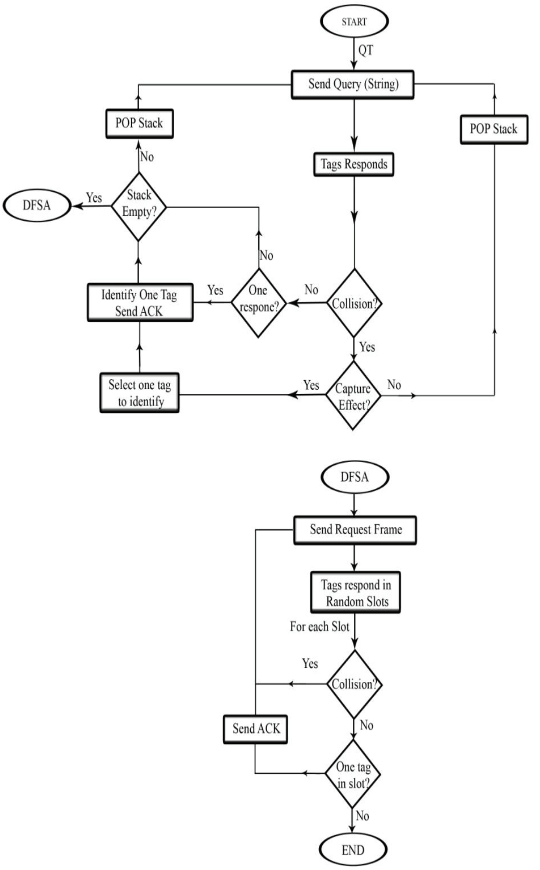

Figure 2: Flow chart of Hybrid QT algorithm

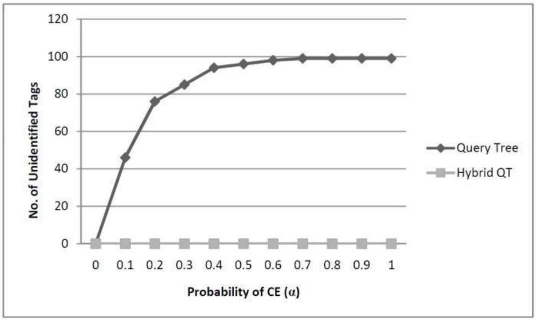

Figure 3: Simulation result for comparison of Hybrid QT and Query Tree

VI. CONCLUSION

Some anti-collision algorithm like QT is proposed to cope with collision in RFID systems. Although QT is efficient algorithm, it suffers from Capture Effect. Capture effect happens when one tag is identified in a collision node. In this situation QT supposes that there is just one tag and now is identified. So QT does not expand that node. So in this paper we present a new algorithm which is combination of Query Tree and DFSA. This uses 2 sequential phases, QT and DFSA. We can understand from Figure 5 and 6 that Hybrid QT is more efficient in RFID environment via high probability of CE and is independent of tag density.

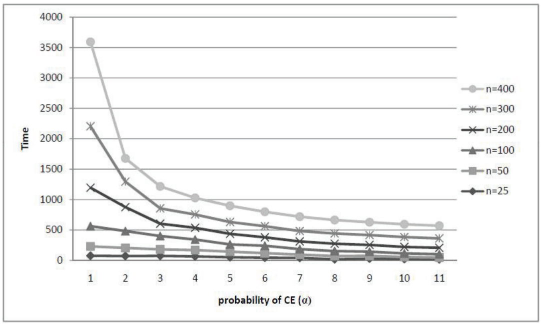

Figure 4: Simulation result of evaluating Hybrid QT algorithm for different values of CE

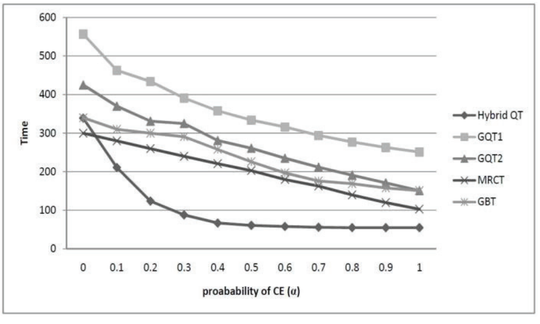

Figure 5: Simulation result for comparison of Hybrid QT, GQT1, GQT2, MRCT and GBT

REFERENCES

[1] S. R. Jeffery, M. J. Franklin and M. Gaorfalakis, “An adaptive RFID middleware for supporting metaphysical data independence” *The VLDB Journal*, 17(3), 265–289, 2008

[2] C. Law, K. Lee, and K.Y. Siu, efficient Memoryless Protocol for Tag Identification, Discrete Algorithms and Methods for MOBILE Computing and Comm, 75-84, 2000

[3] H. Vogt. Efficient object identification with passive RFID tags. In *Proc. of Pervasive Computing*, pp. 98–113, Berlin, 2002.

[4] F. C. Schoute, “Dynamic frame length ALOHA,” *IEEE Trans. Commun*., vol. COM-31, pp. 565-568, Apr. 1983.exc

[5] Weilian Su; Alchazidis, N.; Ha, T.T., "Multiple RFID Tags Access Algorithm," *Mobile Computing, IEEE Transactions on* , vol.9, no.2, pp.174,187, Feb. 2010

[6] Y. Maguire and R. Pappu, “An optimal Q-Algorithm for the ISO 18000-6C RFID Protocol,” IEEE Trans. Automation Science and Engineering, vol. 6, no. 1, pp. 16-24, Jan. 2009.

[7] C. Floerkemeier and M. Lampe, “Issues with RFID usage in ubiquitous computing applications,” Lecture Notes in Computer Science, vol. 3001, pp. 183-193, 2004.

[8] S. Choi, J. Choi, J. Yoo, "An efficient anti-collision protocol for tag identification in RFID systems with capture effect," *Ubiquitous and Future Networks (ICUFN), 2012 Fourth International Conference on* , vol., no., pp.482,483, 4-6 July 2012

[9] Y. Lai, L. Hsiao, "General binary tree protocol for coping with the capture effect in RFID tag identification," *Communications Letters, IEEE* , vol.14, no.3, pp.208,210, March 2010

[10] V.K.Y Wu, R. H. Campbell, "Using Generalized Query Tree to Cope with the Capture Effect in RFID Singulation," *Consumer Communications and Networking Conference, 2009. CCNC 2009. 6th IEEE* , vol., no., pp.1,5, 10-13 Jan. 2009

[11] B. Li; J. Wang, "Efficient Anti-Collision Algorithm Utilizing the Capture Effect for ISO 18000-6C RFID Protocol," *Communications Letters, IEEE* , vol.15, no.3, pp.352,354, March 2011.