Journal of Academic and Applied Studies
(Special Issue on Applied Sciences)
Vol. 3(12) December 2013, pp. 23-38

Int. Association Academemians

ISSN 1925-931X

Available online@www.academians.org

Accepted December 2013

Research Article

Comparison Two Phase Anti Collision Algorithm in RFID Systems

Mohsen Chegin , Mehdi Hossienzadeh

Department of Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

Abstract
In this paper we focus on the anti-collision algorithms in RFID systems and discuss two methods for fast tag identification: Memory less Query Tree (MQT) and Intelligent Query Tree (IQT). These methods can be used for enhancing the identification speed of RFID systems. Compared with the normal query tree protocol, MQT and IQT protocols have lower communication overhead. MQT and IQT protocols have better performance with adopted two phases: first reading cycle and second reading cycle and they have minimal change in tag hardware complexity. In this work we describe two protocols and compare them with QT protocol in the number of query transmissions and number of bit transmissions. Mathematical results show that protocols with two phases can avoid of collisions better than QT protocol in RFID systems.

Keywords: IQT, MQT, QT, performance evaluation.

I. Introduction

The Radio frequency identification (RFID) system consists of radio tags, readers and the Data Processing Subsystem. Traditionally, RFID tags have been used as a replacement for barcodes in applications such as supply-chain monitoring, asset management, and building security (Kim et al., 2008). Although RFID technology is already widely applied in manufacture, supply chain, retail inventory control, transportation, healthcare, agriculture, construction, etc (Bhuptani et al., 2005, Raza et al., 1999) it still needs substantial improvements.
The RFID system contains number of tags with unique serial number (Eslamnezhad et al., 2011). The reader receives required information from the tags by sending and receiving wireless signals with the tag. The RFID system contains number of tags with unique serial number. The reader receives required information from the tags by sending and receiving wireless signals with the tag (Golsorkhtabaramiri et al., 2013).

Int. Association Academians
ISSN 1925-931X

Available online@www.academians.org

The Data Processing Subsystem initiates the reader and tags activities. RFID systems provide a quick, flexible, and reliable way to electronically detect, track and control a variety of items. RFID systems use radio transmissions to send energy to a RFID tag while the tag sends an identification code back to a reader linked to an information management Subsystem shown in Figure 1.

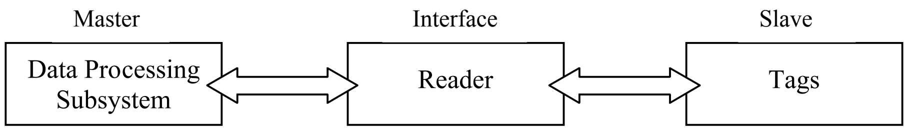


Figure 1. RFID System

One of the most important purposes in RFID systems is tags recognition in lessen possible time and with minimum searches in reader range wave. For fast tag identification, anti-collision protocols, which reduce collisions and identify tags irrespective of occurring collisions, are required (Myung et al., 2005, Myung et al., 2006a).

Tag collisions occur when multi tags try to respond to a reader simultaneously and cause the reader to identify no tag. This situation will lead to retransmission of tag IDs, which results in wastage of bandwidth and increases the total delay in identifying all the objects. Hence, anti-collision algorithms need to be divided between the tags and readers to minimize collisions. In general, the tag anti-collision techniques can be classified into two categories, ALOHA based algorithms and tree based algorithms.

ALOHA based algorithms, such as ALOHA, slotted ALOHA and frame slotted ALOHA, allow the tags to transmit their own serial numbers at a timeslot. In ALOHA, tags randomly choose their transmission time and, in slotted ALOHA, tags transmit only at the beginning of a timeslot which is a distinct time period. Frame slotted ALOHA configures a frame with many timeslots (Bio et al., 2006, Liu et al., 2006, Cho et al., 2007). As a tag sends its ID only at a single timeslot in every frame, the frame slotted ALOHA method decreases the collisions. The frame slotted ALOHA is the best choice of ALOHA based algorithms.

ALOHA based algorithms cannot completely avoid collisions. Moreover, these algorithms cannot guarantee that all the tags are recognized within a certain time period if the number of tags is not known in advance. Therefore, a specific tag may not be identified for a long time and causes tag starvation problem (Myung et al., 2007).

The methods based on tree anti-collision algorithms keep splitting the group of colliding tags into two subgroups until all tags are identified. The binary tree and the query tree (QT) algorithms are some of the tree based algorithms. In the binary tree algorithms, when a collision occurs, the reader splits a set of the colliding tags into two groups until a tag is identified without collision. In the query tree algorithms, a reader sends out a prefix in each communication round and tags respond with their IDs if the prefix matches parts of their IDs. When a collision occurs for a particular prefix, the reader ignores the response and expands the prefix by one bit-longer prefix later (Myung et al., 2005).

The QT algorithm minimizes the number of messages sent by a reader compared to the binary tree algorithm and reduces idle and collision cycles. Moreover its implementation has no more complexity. There are different schemes of the QT protocols (as noted in Jung-Sik et al., 2008, Okkyeong et al., 2009, Chen et al., 2009) for reducing the information exchanging overhead between the reader and tags, and also to have shorter identification time.

But there are few schemes of the QT protocols that use two read cycles. They aim at accelerating the identification phase by using the information of the previous phase and using the stored information belonging the previous processes. Two enhanced QT protocols that have been proposed in this paper are Memory less Query Tree (MQT) (Myung et al., 2006b) and Intelligent Query Tree (IQT) (Naval et al., 2006). In this work we describe two protocols and compare them with QT protocol in the number of query transmissions and number of bit transmissions.

II. Related Works

A. QT protocol

Chen et al. (2009) described each tag having an unique ID string in $\{0, 1\}^k$ in which k is the length of the tag ID string. A string of k-bits uniquely identifies each tag. The current response of each tag only depends on the current query of the reader but not on the past history of the reader’s queries (memory less). The only required computation for each tag is to try to match its ID against the string in the query.
The QT algorithm consists of some queries and responses. The reader asks the tags whether any of their IDs contains a certain prefix. If more than one tag answers, then there is a collision and the reader knows there are at least two tags having same prefix. The reader then appends bit 0 or 1 to the prefix, and continues to query for longer prefixes. When a prefix matches a tag uniquely, the reader identifies a tag. So, by extending the prefixes until only one tag’s ID matches, the algorithm can discover all the tags. QT protocol (Abraham et al., 2002) on the assumption of three tags is shown in Figure 2 whose identifying IDs are 000, 001, and 100, respectively. Prefix 0 and 1 are the elements placed at the first of the queue, and the reader queries the tags by taking out prefixes in the queue. Initially, 0 is queried and tags 000 and 001 respond at the same time because the prefix is the same as their ID. Then the reader understands there are at least two tags that start with 0 and cause collision, then attends the bits 0 and 1 to the end of the previous prefix and enters 00 and 01 in the queue. Thereafter, prefix 1 is taken out from the queue and queried. Since there is only one tag, 100 is identified to the reader. The first round ends by the scheme

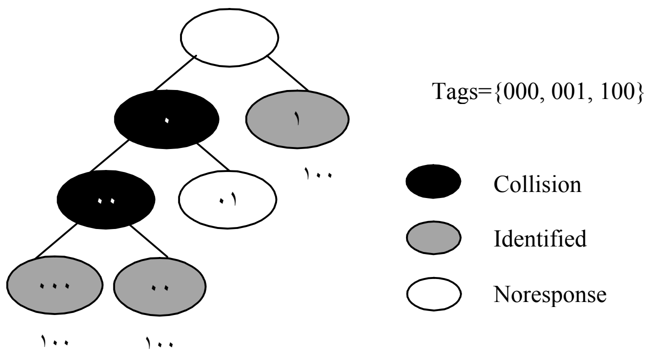
Journal of Academic and Applied Studies
(Special Issue on Applied Sciences)
Vol. 3(12) December 2013, pp. 23-38

Int. Association Academians

ISSN 1925-931X

Available online@www.academians.org

like this. The next round begins for 00 and 01 which has been previously placed in the queue. This algorithm ends when the queue is empty.



Query	0	1	00	01	000	001
Response	collision	identified	collision	no response	identified	identified

Figure 2. Identification process in QT protocol

B. MQT protocol

Two important procedure of the Memory less Query Tree protocol are the query insertion procedure and the query deletion procedure. MQT protocol uses information which is obtained during the previous identification process. The reader establishes the queue *Q* and also a candidate queue *CQ*. All the prefixes that cause collisions are maintained into *Q,* but prefixes that receive no or one reply are maintained into *CQ*. In start the reader initializes the queue *Q* with the candidate queue *CQ* which saves prefixes from the previous process of identification.

The candidate queue *CQ* saves prefixes of identified nodes for fast re-recognizing and also saves the no-response nodes for identifying the new tags. With the query insertion procedure, the queue *Q* is increasing and the size of prefixes in the queue *Q* are extending over time and disturb the fast identification. After the first non-initial process, unnecessary prefixes have to be deleted in query deletion procedure. This is to maintain the query tree and find the shortest and least prefixes for re-identifying current set of tags.

C. IQT protocol

Intelligent Query Tree Protocol (Naval et al., 2006) uses specific prefix in the tag identification to reduce the collision period between reader and tags. The common prefix may be due to the fact that items to which the tags are attached have the same manufacturer or product type.

IQT protocol use information in Electronic Product Code (EPCglobal Standard Specification, 2004). The EPC Numbering System uniquely identifies objects and facilitates tracking throughout the product's life cycle.
According to Table I EPC code essentially has four fields: *EPC Version*, *Manufacturer ID*, *Product Type ID*, *Item ID*.

TABLE I
FOUR FIELDS IN EPC CODE

Electronic Product Code (96 bits)			
EPC version	Manufacture ID	Product Type ID	Item ID
0-7 bits	8-35 bits	36-59 bits	60-95 bits

IQT identifies tags where the tag IDs have some common prefix (e.g. EPC Version, Manufacturer ID*)* because usually many of products have same specific pattern in their tag IDs.
It also uses history of read cycles to further improve the tag read efficiency. Following are its advantages of Intelligent Query Tree Protocol (Naval et al., 2006):

- Reduction in number of bits transmitted by tag.
- Reduction in number of collisions by maintaining the history of tag read patterns.
- Reduction in number of collisions for subsequent read cycles by using the information of previous read cycles.

III. Mathematical analysis of MQT, IQT, QT Protocols

For achieving at better performance, we used protocols with adopted two phases: first reading cycle and second reading cycle like MQT and IQT protocols that have better performance and have minimal changes in the tag hardware complexity.
The performance of Memory less Query Tree Protocol and Intelligent Query Tree Protocol is compared with normal Query Tree Protocol. Better efficiency can be obtained by decreasing the number of bits transmitted among reader and tag in one query, as well as by reducing the number of queries between reader and tag in one reading cycle. Some optimization techniques for the first reading cycle and the second reading cycles are applied in MQT and IQT.
The MQT protocol generates prefixes that are obtained from the past processes of tag identification. A tag just requires matching a prefix of a query with its ID. We focus our attention on decreasing the number of collisions. Because the collision leads to the communication overhead and the communication overhead consequently leads the transmission delay. IQT

Journal of Academic and Applied Studies
(Special Issue on Applied Sciences)
Vol. 3(12) December 2013, pp. 23-38

Int. Association Academians

ISSN 1925-931X

Available online@www.academians.org

protocol improvement is mainly due to reduction in the number of queries and reduction in bits transmitted between reader and tag.
We perform mathematical analysis of MQT and IQT Protocols over QT at the worst case of tag identification. The paper assumes there are *n* tags to be recognized and each tag has a unique ID of k=96 bits long. There are two performance parameters to identify all the tags: 1) the number of queries and 2) the number of bits per query. These two parameters mainly decide the communication overhead.

A. First Read Cycle

This paper splits the First Read Cycle into two sections: when a guess happens in and when no guess occurs in $\max_{guesses}$ tries. First read cycle in IQT protocol is guessing phase continues for a maximum of $\max_{guesses}$ number of times. But in MQT protocol the reader initializes the queue Q with the candidate queue CQ which stores prefixes from the last process of identification. Mathematical results discusses in below: generally when there are n tags, below Theorem 1 can be found in (Naval et al., 2006).

Theorem1: If n is the number of tags, then the maximum number of nodes in the query tree is given by:

$$2^{[\quad(\quad)]} - 1 + (k - [\log\ (n-2)]) * (n) \text{ If n is even} \tag{1}$$

$$2^{[\quad(\quad)]} - 1 + (k - [\log\ (n-2)]) * (n-1) \text{ If n is odd} \tag{2}$$

In follow the paper we are assuming that n is odd. So, Theorem1 shows that with n tags we need maximum of MAX_{QUERY} queries. Where MAX_{QUERY} is:

$$MAX \qquad = 2^{[\quad(\quad)]} - 1 + (k - [\log\ (n-2)]) * (n-1) \tag{3}$$

All the number of queries required in Query Tree Protocol in the first read cycle is given by:

$$2 * pre\ \ ix + 1 + (\,2^{[\quad(\quad)]} - 1 + (k - pre\ \ ix - [\log\ (n-2)]) * (n-1)\,) \tag{4}$$

In QT protocol, we are assuming that tags have same prefix, so, the remaining queries under common prefix will require k-prefix queries. if each tags response with their k bit tag IDs in the QT Protocol, the total number of bits transmitted between reader and tags as shown below:

$$(2 * pre\ \ ix + 1 + (\,2^{[\quad(\quad)]} - 1 + (k - pre\ \ ix - [\log\ (n-2)]) * (n-1)\,)) * k \text{ bits} \tag{5}$$

Journal of Academic and Applied Studies
(Special Issue on Applied Sciences)
Vol. 3(12) December 2013, pp. 23-38

Int. Association Academians

ISSN 1925-931X

Available online@www.academians.org

We show that IQT and MQT performs much better than QT. since, readers running IQT and MQT protocol can read more tags per unit time. When the number of the collisions is minimal the best case performance for MQT can be achieved. Theorem 2 can be used in MQT protocol:

Theorem 2: All non-leaf nodes show collisions and each collision needs two queries to check the collision for read its two child nodes.

The maximum number of collisions in MQT protocol can be acquired when the first K -1 bits of two or several tag are the same. This means that the maximum number of collisions can be found in [Haosong et al., 2012]:

$$n(k + 2 - \log\ n) - 1 \qquad (6)$$

Thus, in the worst case of MQT the maximum number of queries is:

$$2 * \text{pre}\ \ \text{ix} + 1 + \ 2^{[\ \ (\ \ )]} - 1 + (k - \text{pre}\ \ \text{ix} - [\log\ (n - 2)]) * (n - 1) \qquad (7)$$

This number of queries are for initial identification process (occurs just one time) and in another first read cycles the number of queries will be very decrease calculated by $(2 * n) - 1$. Since, tags are recognized with prefixes of identified nodes with no collision or only a few collisions. The total number of bits transmitted between reader and tags in initial status is given by:

$$2 * \text{pre}\ \ \text{ix} + 1 + \ 2^{[\ \ (\ \ )]} - 1 + (k - \text{pre}\ \ \text{ix} - [\log\ (n - 2)]) * (n - 1)\ \ * k\ \text{bits} \qquad (8)$$

Or in another first read cycles with no collision or a few collisions number of bits transmitted is as following:

$$(2 * n) - 1\ \ * k \qquad (9)$$

In the First Read Cycle of IQT protocol if all the tags have first prefix bits same, the reader is able to guess the prefix in $\text{num}_{\text{guesses}}$ tries, total number of queries required by IQT protocol is given by[Naval et al., 2006]:

$$\text{num}\qquad + 1\ + \ 2^{[\ \ (\ \ )]} - 1 + (k - \text{pre}\ \ \text{ix} - [\log\ (n - 2)]) * (n - 1) \qquad (10)$$

There for the total number of bits transmitted with $\text{num}_{\text{guesses}}$ guesses is given by:

Journal of Academic and Applied Studies
(Special Issue on Applied Sciences)
Vol. 3(12) December 2013, pp. 23-38

Int. Association Academians

ISSN 1925-931X

Available online@www.academians.org

$$\mathrm{num} \quad +1 \ * k + \ 2^{[\quad (\quad)]} \ -1 + (k - \mathrm{pre} \ \ \mathrm{ix} - [\log \ (n-2)]) * (n-1) \ * (k - \mathrm{pre} \ \ \mathrm{ix}) \qquad (11)$$

When all the tags don't have first prefix bits same after $\max_{\text{guesses}}$ tries, hence, first prefix bits are learnt using the query tree protocol. Number of queries in this status is given by:

$$2 * \mathrm{pre} \ \ \mathrm{ix} + 1 + \ 2^{[\quad (\quad)]} \ -1 + (k - \mathrm{pre} \ \ \mathrm{ix} - [\log \ (n-2)]) * (n-1) \ + \max \qquad (12)$$

And the total number of bits transmitted is given by:

$$2 * \mathrm{pre} \ \ \mathrm{ix} + 1 + \max \quad * k + \ 2^{[\quad (\quad)]} \ -1 + (k - \mathrm{pre} \ \ \mathrm{ix} - [\log \ (n-2)]) * (n-1) \ * (k - \mathrm{pre} \ \ \mathrm{ix}) \qquad (13)$$

B. *Second Read Cycle*

In the MQT protocol, the 2th Read Cycle makes use of prefixes in the 1th process. All the prefixes in 1th process are either the ones which got one reply or idle prefixes in the previous process. In the best case for the number of queries each tag can be identified by one query. So, the average number of queries in Second Read Cycle in MQT is given by:

$$(n(k + 2 - \log \ \ n) \div 2) + n \ \ \div 2 \qquad (14)$$

Also, In IQT protocol, when set of tags is exactly same as that of first read cycle, we will save $2 * \mathrm{pre} \ \ \mathrm{ix}$ queries. Hence, number of queries in Second Read Cycle in IQT is given by:

$$2^{[\quad (\quad)]} \ + (k - \mathrm{pre} \ \ \mathrm{ix} - [\log \ (n-2)]) * (n-1) \ \div 2 \qquad (15)$$

IV. Performance Evaluation

The MATLAB software has been used to evaluate the performance of the proposed idea. The mathematical formulas have been entered to the MATLAB software and the results of the analysis have been extracted; then have been saved in some tables. Then the entries of the tables have been written to some tables.

In this study, the tag ID is 96 bits long according to EPC Class 1 Gen.2 standard (EPCglobal Standard Specification, 2004). The communication overhead is measured by the number of queries sent by the reader and the number of bits transmitted by tags. Performance of QT is compared with IQT, MQT. Performance was evaluated with various numbers of tags such as 5, 15, 25, 35, 45 and 55. Table II shows various Variables used in this Evaluation.

TABLE II
VARIABLES USED IN THIS EVALUATION

Variable	Value
N	5, 15, 25, 35, 45 and 55
Prefix	60 bits
K	96 bits
$\mathrm{Max}_{\mathrm{gusses}}$	4 times

A. *Performance Evaluation in First Read Cycle*

Figures 3 and 4 compare the number of query for identification and communication overhead against different numbers of tags in the first read cycle. In figure 3 guessing phase prefix succeeds, then IQT knows the first prefix bits that are same in all the tags present. But in figure 4 guessing phase for prefix not succeeds. IQT show the best performance in number of query.

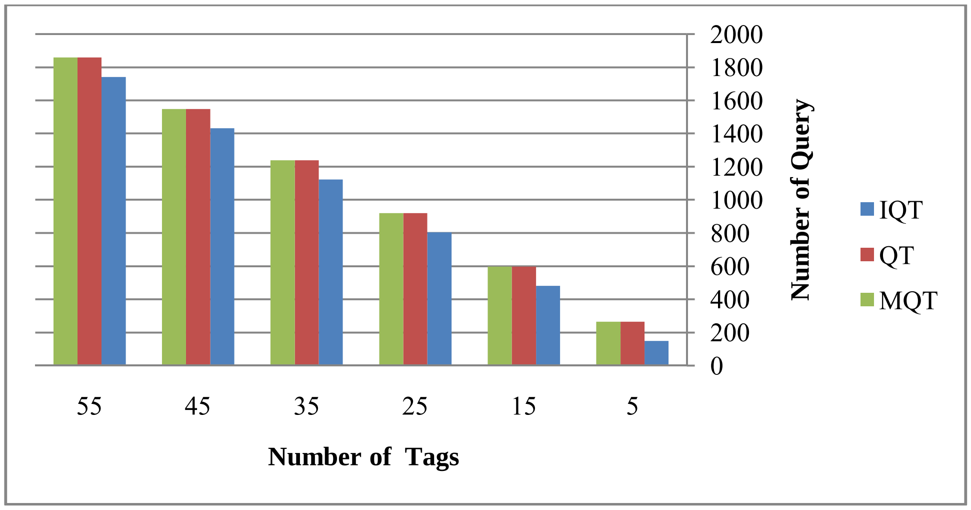


Figure 3. Number of queries in first read cycle with succeed guess

Journal of Academic and Applied Studies
(Special Issue on Applied Sciences)
Vol. 3(12) December 2013, pp. 23-38

Int. Association Academians

ISSN 1925-931X

Available online@www.academians.org

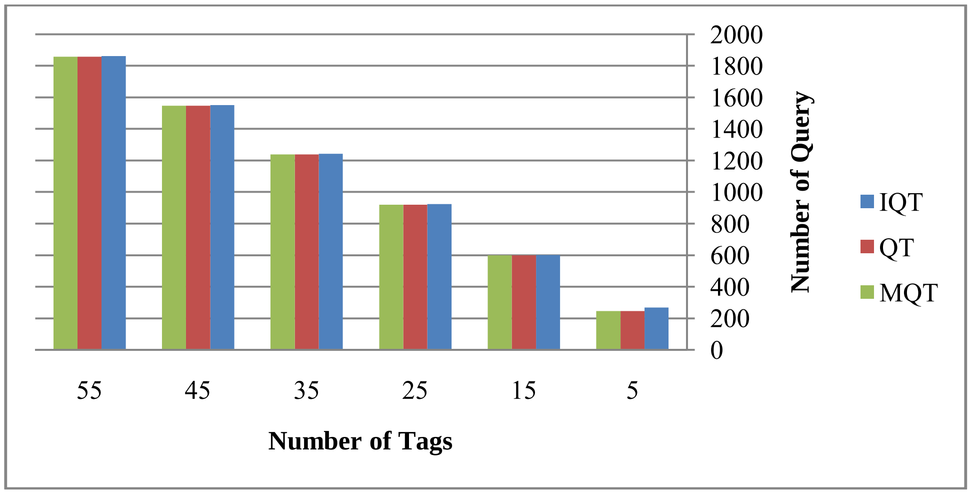


Figure 4. Number of queries in first read cycle with No-succeed guess

Figures 5 and 6 shows that IQT outperforms the QT and MQT protocols in respect of the total number of bits transmitted for identification, when guessing phase prefix succeeds and no guessing phase prefix succeeds, respectively.

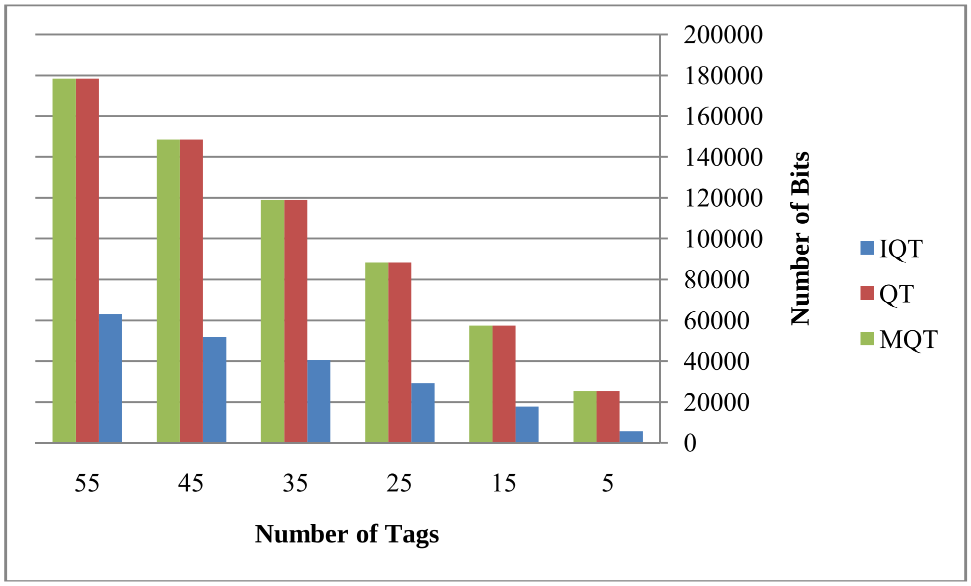


Figure 5. Number of bits in first read cycle with succeed guess

Journal of Academic and Applied Studies
(Special Issue on Applied Sciences)
Vol. 3(12) December 2013, pp. 23-38

Int. Association Academians

ISSN 1925-931X

Available online@www.academians.org

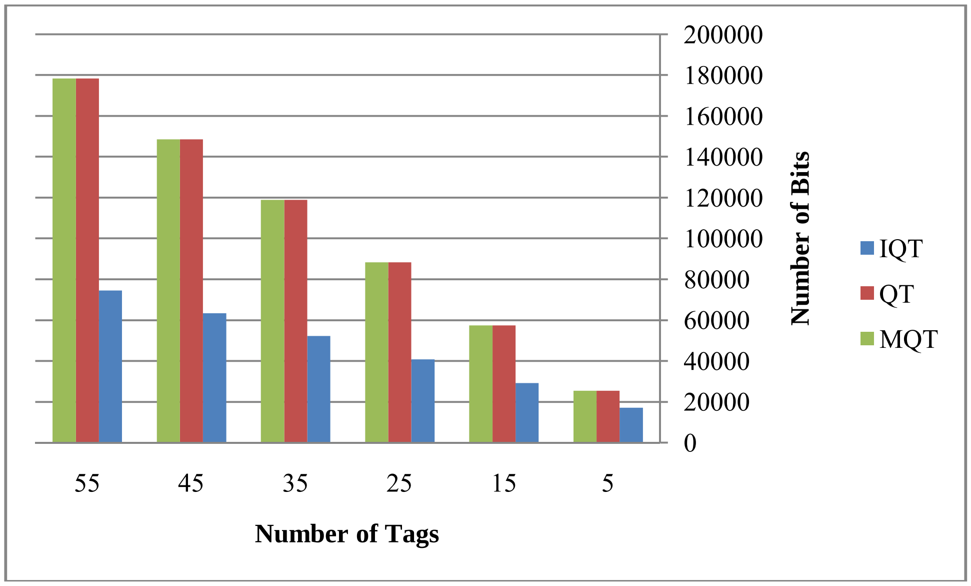


Figure 6. Number of bits in first read cycle with No-succeed guess

after initial identification process in MQT Protocol that occurs only one time (In this case normal Query Tree Protocol will be used for tag identification) in another first read process the number of queries will be very decrease calculated by (2*n)-1. Figure 7 shows that MQT outperforms the QT and IQT (even with guessing phase prefix succeeds) protocols in respect of the number of query for identification. This is because the MQT protocol obtained the information of previous initial identification process to reduce collisions. Thus MQT show the best performance if MQT protocol used the information of previous initial process (Figure 7).

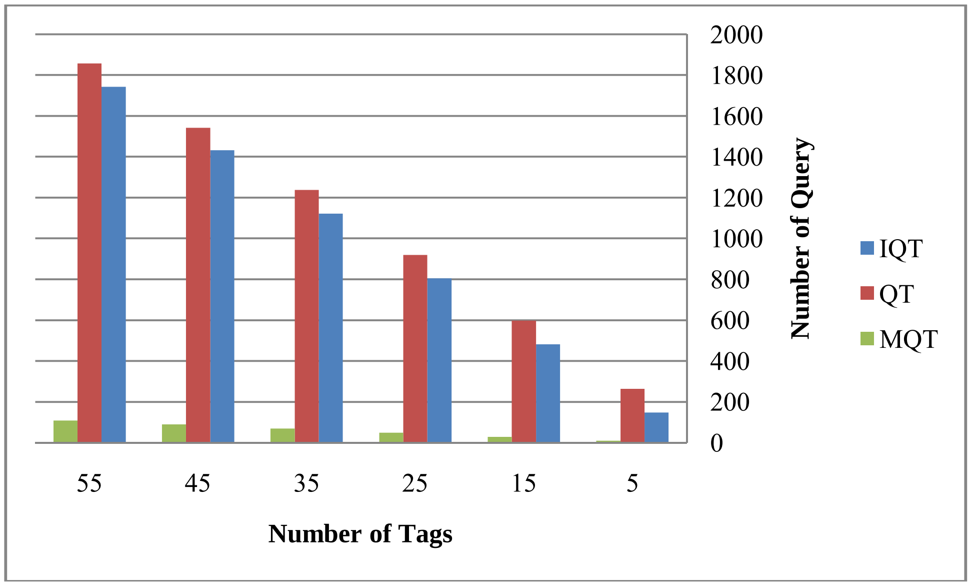


Figure 7. Number of queries with MQT protocol used the previous information

B. Performance Evaluation in Second Read Cycle

When guessing phase prefix succeeds, the IQT protocol outperforms QT and MQT in respect of the number of queries and number of bits transmitted respectively, as shown in Figure 8 and 9.

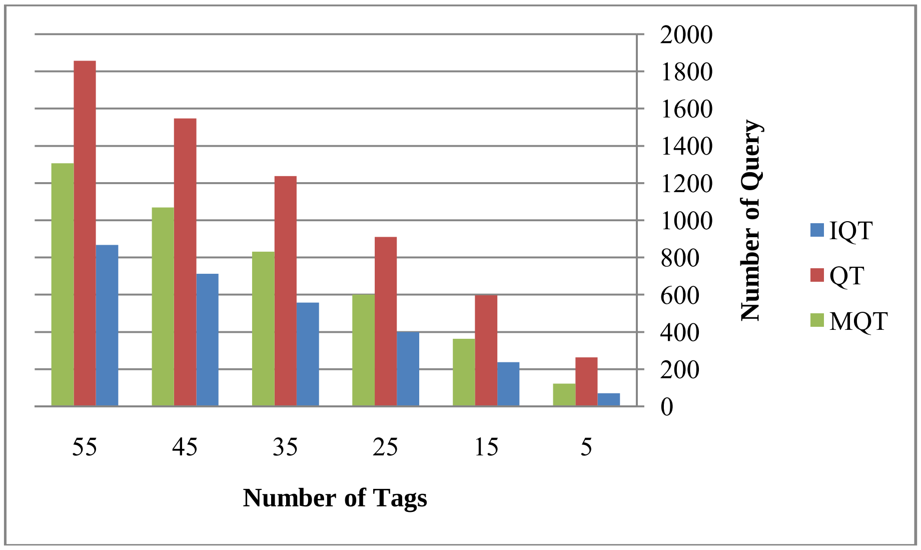


Figure 8. Number of queries in second read cycle with succeed guess

Journal of Academic and Applied Studies
(Special Issue on Applied Sciences)
Vol. 3(12) December 2013, pp. 23-38

Int. Association Academians

ISSN 1925-931X

Available online@www.academians.org

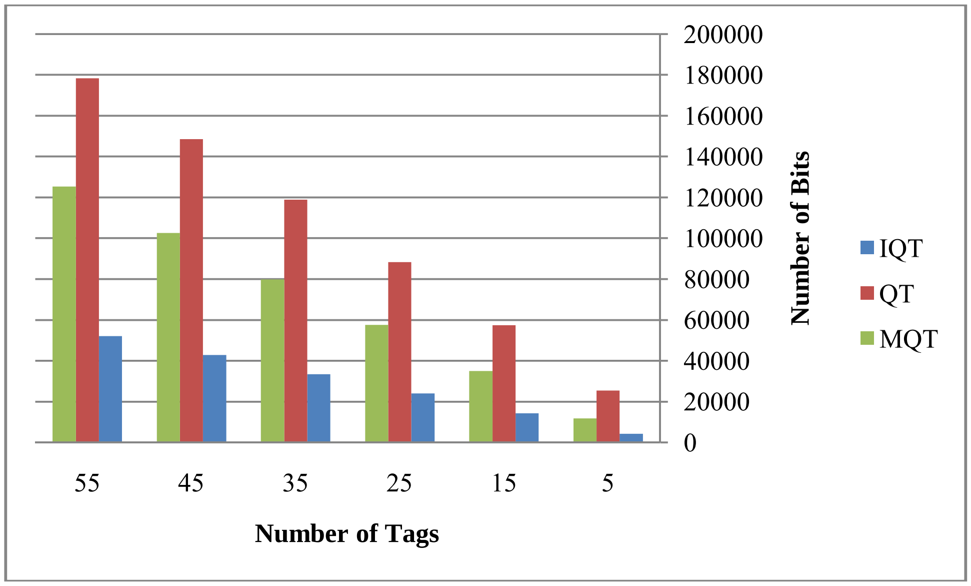


Figure 9. Number of bits in second read cycle with succeed guess

Due to the reduction of queries on tag identification, IQT achieve better performance than QT and MQT in respect of the number of bits transmitted, as shown in Figure 9. When no guessing phase prefix succeeds, MQT employs relevant information between two consecutive processes to reduce the number of queries and number of bits transmitted. Moreover, with the number of tags increasing, the advantage of MQT becomes more obvious. Thus as shown in Figure 10, MQT protocol outperforms QT and IQT in respect of the number of query. The decreased number of queries also reduces the number of bits transmitted as shown in Figure 11.

Journal of Academic and Applied Studies
(Special Issue on Applied Sciences)
Vol. 3(12) December 2013, pp. 23-38

Int. Association Academemians

ISSN 1925-931X

Available online@www.academians.org

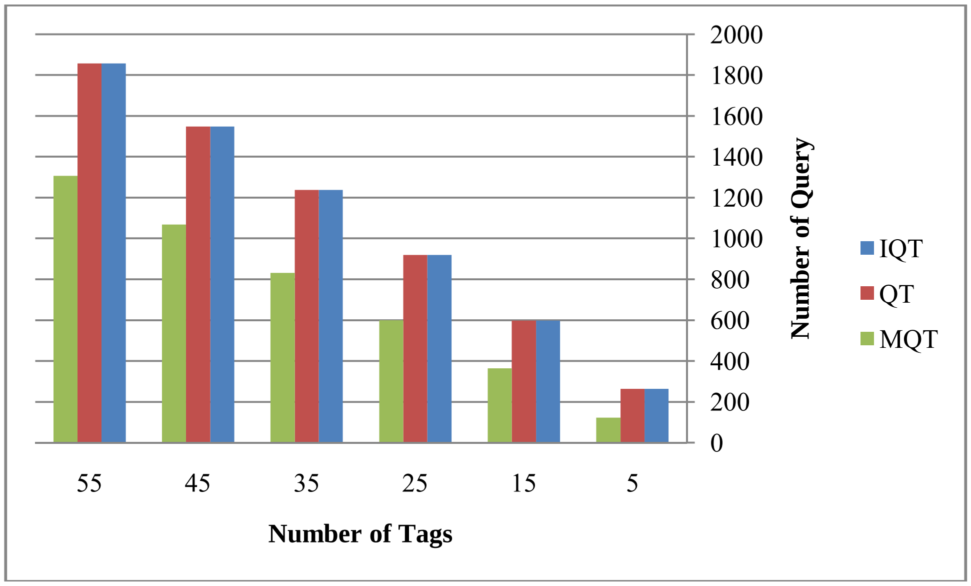


Figure 10. Number of queries in second read cycle with No-succeed guess

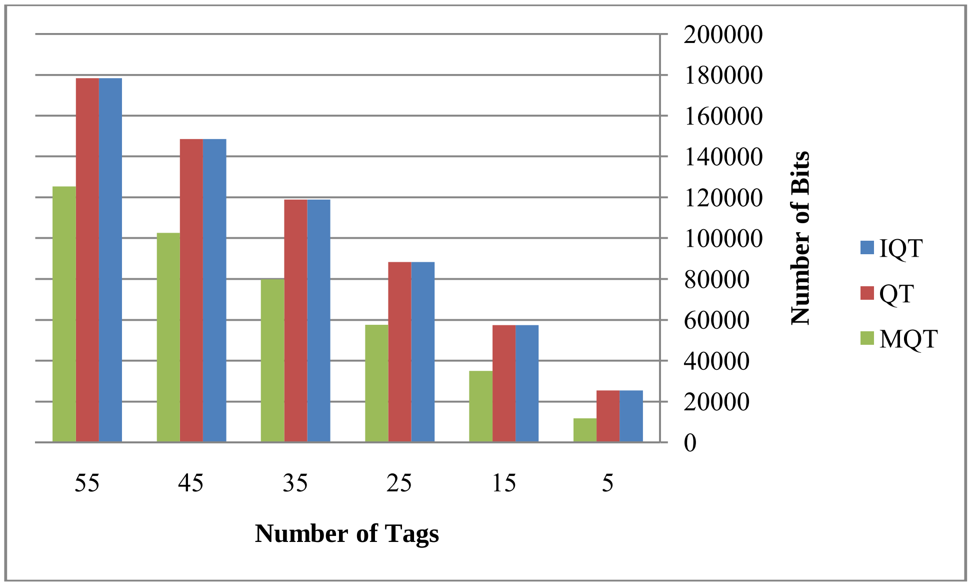


Figure 11. Number of bits in second read cycle with No-succeed guess

Int. Association Academians

ISSN 1925-931X

Available online@www.academians.org

V. Conclusion

In this paper, we compared MQT and IQT protocols with normal QT protocol in respect of number of queries transmitted by reader and number of bits transmitted by tags. Mathematical analysis shows that decreased number of queries also reduces the number of bits transmitted in RFID systems and vice versa. Compared with the normal query tree protocol, MQT and IQT protocols have lower communication overhead. If the number of tags between two consecutive processes is similar, the performance keeps stable.

Acknowledgment

The author wishes to acknowledge the Islamic Azad University, Dezful branch, for supporting the work.

References

Abraham, V. Ahuja, A.K. Ghosh, P. Pakanati (2002), Inventory Management using Passive RFID Tags: A Survey, Department of Computer Science, The University of Texas at Dallas, Richardson, Texas, pp. 1–16.

Bhuptani and S. Moradpour (2005), RFID field guide:deploying radio frequency identification systems. Upper Saddle River, NJ: *Sun Microsystems/Prentice Hall.*

Biao, H. Ai qun, and Q. Zhong yuan (2006),Trends and Brief Comments on Anti-collision Techniques in Radio Frequency Identification System, *6th International Conference on ITS Telecommunications Proceedings*, pp241-245

Chen-Cung Liu, Yin-Tsung ChanN(2009), An anti-collision protocol of RFID based on divide and conquer algorithm, *26th Workshop on Combinatorial Mathematics and Computation Theory*.

Cho, W. Lee, and Y. Baek (2007), LDFSA: A Learning-Based Dynamic Framed Slotted ALOHA for Collision Arbitration in Active RFID Systems, *Advances in Grid and Pervasive Computing*, pp. 655-665.
EPCglobal Standard Specification(2004), EPC radio-frequency identity protocols class-1 generation-2 UHF RFID protocol for communications at 860 MHz - 960 MHz, version 1.0.9.

Eslamnezhad, F. B. Aghdam, M. Hosseinzadeh(2011), A Secure and Efficient RFID Mutual Authentication Protocol, *International Journal on Communications Antenna and Propagation (IRECAP)*,Vol(1), No. 5, pp.429-433

Golsorkhtabaramiri, Mehdi Hosseinzadeh(2013), A novel stable cluster-based protocol for heterogeneous RFID enhanced wireless sensor networks, *QScience Connect*.

Journal of Academic and Applied Studies
(Special Issue on Applied Sciences)
Vol. 3(12) December 2013, pp. 23-38

Int. Association Academians

ISSN 1925-931X

Available online@www.academians.org

Haosong G and Younghwan Y(2012), Bit Collision Detection Based Query Tree Protocol for Anti-Collision in RFID System, *Journal of Innovative Computing, Information and Control*, Vol(7), pp 3081-3102.

Jung-Sik Cho, Jea-Dong Shin, Sung K. Kim (2008), RFID Tag anti-collision protocol: Query tree with reversed IDs. *10th International Conference on Advanced Communication Technology (ICACT)*, Vol(1),pp 225–230.

Kim, S. J. Lee, and K. S. Ahn (2008), An Efficient Anti-Collision Protocol Using Bit Change Sensing Unit in RFID System, *The 14th IEEE International Conference on RTCSA 08*, pp. 81-88.

Liu and S. Lai (2006), ALOHA-Based Anti-Collision Algorithms Used in RFID System, *Wireless Communications, Networking and Mobile Computing*.

Myung, and W. Lee (2005), An adaptive memoryless tag anticollision protocol for RFID networks, *IEEE INFOCOM'05*.

Myung, W. Lee, and J. Srivastava, Adaptive binary splitting for efficient RFID tag anti-collision, *IEEE Communication Letter*, Vol(10), pp. 144–146.

Myung, W. Lee, J. Srivastava, and T.K. Shih (2007), Tag-Splitting: adaptive collision arbitration protocols for RFID tag identification, *In IEEE Trans. on Parallel and Distributed Systems*, Vol(18), pp. 763-775.

Myung, W. J. Lee and T. K. Shih (2006), An adaptive memoryless protocol for RFID tag collision arbitration, *IEEE Transactions on Multimedia*, Vol(8).

Naval Bhandari, Anirudha Sahoo, and Sridhar Iyer (2006), Intelligent Query Tree (IQT) Protocol to Improve RFID Tag Read Efficiency , *IEEE Computer Society*, pp 46-51.

Okkyeong Bang, Choi, J. H., Lee, D. W., Lee, H. J.(2009), Efficient novel anti-collision protocols for passive RFID tags. *Auto-ID Labs White Paper www.autoidlabs.org*.

Raza, V. Bradshaw, and M. Hague, Applications of RFID technology, *presented at RFID Technology (Ref. No. 1999/123), IEE Colloquium*.