This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2009 proceedings.

Smart Trend-Traversal: A Low Delay and Energy Tag Arbitration Protocol for Large RFID Systems

Lei Pan and Hongyi Wu

***Abstract*—We propose a Smart Trend-Traversal (STT) protocol for RFID tag arbitration, which effectively reduces the collision overhead occurred in the arbitration process. STT, a Query Tree-based scheme, dynamically issues queries according to the online learned tag density and distribution; and therefore, it significantly reduces delay and energy consumption comparing with the existing Tree-based and Aloha-based protocols. Our analytic studies further show that the optimality of STT does not rely on any presumed network conditions, which is in sharp contrast to other available schemes and renders it a highly desirable and practical solution.**

I. Introduction

The promising potentials of RFID technology such as inventory management and real-time monitoring have brought tremendous attentions to large-scale RFID systems. As a result, tag arbitration is facing unprecedent challenges to achieve optimal or close-to-optimal efficiency, which otherwise may lead to significant delay and energy consumption and hinder RFID's wide application in large industrial systems.

However, fast arbitration of massive tags is nontrivial to achieve in RFID systems, especially due to the limited computation and communication capability of tags. The existing tag arbitration protocols can be grouped into two categories: the Aloha-based and the Tree-based. Both approaches, although taking completely different strategies to address the problem, are facing the common dilemma: performance degradation in large-scale RFID systems due to the excessive collisions incurred. Not only does the collision increase the process delay, it also consumes extra energy, which is critical for the battery powered readers and active RFID tags. Several algorithms are proposed recently aiming to reduce the tag collision, but an optimal and practical solution for tag arbitration is hardly found in the literature.

To provide the background to tag arbitration protocols, we briefly discuss the two categories of tag arbitration schemes. The Aloha algorithm is a probabilistic approach where the tags respond to the reader's query at random time [1]–[3]. The frame slotted Aloha is further proposed by restricting tag responses within a frame [4]–[6]. A proper frame size is essential to the frame slotted Aloha algorithm. To choose the frame size, the number of tags present must be known or estimated in prior. In general, the Aloha-based schemes are simple and with low implementation cost. They work well under sparse networks. But the random reply strategy does not guarantee a delay bound for recognizing all tags, and its performance degrades with the increase of tag density.

Alternatively, the tree-based algorithms utilize the binary or n-ary tree structure to split tags into smaller subsets until the collision is resolved, which provides a bounded delay for tag arbitration [7]. For example, the Query Tree (QT) protocol adopts a memoryless scheme for tag identification, by splitting the tags based on the binary structure of their tag ID's [8], [9]. The idea of the QT protocol can be described as follows. At the beginning of each time slot, the reader issues a query that contains a h-bit binary string called prefix. All tags will respond if the first h bits in their ID's match the prefix. When multiple tags respond in the same slot, collision happens and the algorithm moves down to the next level of the tree by appending bit 0 or 1 to the prefix, and continues to query until the collision is resolved. This is essentially a depth-first pre-order traversal. It guarantees all the leaf nodes will respond at least once in the process. However, this scheme inevitably introduces a large number of collisions when the tag distribution is dense. In the worst scenario that the tree is full (i.e., all tags are present), the QT protocol has to visit every node, resulting 100% overhead.

The above observations naturally lead us to the question: how to reduce the collision in the arbitration? A straightforward answer will be to redirect the way a reader issues queries so that less collision nodes are visited. There are several protocols proposed with this attempt. For example, an optimal algorithm is introduced in [9] to start the query from Level h ($h > 0$) in order to avoid visiting nodes at the upper part of the tree. Essentially, it forms 2^h sub-trees and identifies tags in each sub-tree. Separately, the Multi-Slotted Scheme with Assigned Slots protocol (MAS) is proposed in [10]. MAS divides the current set into F ($F = 2^j$) subsets when collision occurs. For example, if $F = 4$ and a collision occurs at the root, then the query will directly go to Level 2 and skip the nodes at Level 1. Clearly, bigger F works better in dense networks and smaller F fits for sparse ones.

These enhancements intend to improve the query process and reduce collisions. Their performance, however, largely depends on the optimality of a few key parameters (e.g., h in [9] and F in [10]). Although such parameters can be optimized under a given network scenario, it is very difficult to dynamically choose their optimal values in general applications. In particular, they make two strong assumptions

This work is supported in part by the National Science Foundation under grant CNS-0831823.

The authors are with the Center for Advanced Computer Studies, University of Louisiana at Lafayette, Lafayette, LA 70504. Email:{lxp1250, wu}@cacs.louisiana.edu.

978-1-4244-3513-5/09/$25.00 ©2009 IEEE

Authorized licensed use limited to: University of Louisiana at Lafayette. Downloaded on August 26, 2009 at 08:24 from IEEE Xplore. Restrictions apply.

that are rarely true in practice: (a) the number of tags (M) is known, and (b) the tag distribution is uniform.

In reality, obtaining M is nontrivial in large RFID systems. When there are over thousands of tags present, it takes long time just to get an estimation of the tag population [11]. A Lottery Frame scheme has been proposed to utilize the geometric nature of the binary ID to provide quick estimation of M [12]. However, the tag distribution must be uniform, otherwise the scheme may incur major errors. In practice, it is unrealistic to assume the tag distribution is always uniform. Consider the goods in the warehouse that belong to several specific categories. The distribution of the tag ID's in one category may be uniform, but it is not so overall. Such scenario is common for inventory management.

In a nutshell, the existing enhancements of the tree-based protocol are all static approaches, without the ability to adapt to a wide range of application scenarios. This motivates us to rethink the design philosophy for a more flexible and intelligent solution. In this paper, we propose a Smart Trend-Traversal (STT) protocol for tag arbitration, which eliminates the drawbacks of the tree-based and the Aloha-based algorithms. With the ability to dynamically issue queries according to the online self-learned tag density and distribution, STT significantly reduces the collisions that commonly occur in the existing algorithms. The key contributions of our work are highlighted as follows:

a) Trend recognition: In STT, the reader follows the trend of tag density and distribution to issue queries, and therefore, it keeps the number of empty slots and collision slots to the minimum.

b) Practicality: There is no need to estimate M or assume the tag distribution is uniform. STT works without any prior knowledge of the network and outperforms the existing protocols. Moreover, the computation is done by the reader. The algorithm adds no computation overhead to the tags that have very limited processing power.

c) Delay guarantee: STT is a deterministic and memoryless protocol. All tags can be identified with a bounded delay, and no state information is needed.

II. The Proposed STT Protocol

The Smart Trend-Traversal (STT) protocol is proposed in this section. In this research, we focus on memoryless Query Tree-based protocols, aiming to achieve efficient tag arbitration with low delay and energy consumption under a wide range of tag density and ID distributions. First, we introduce a few important terms and definitions frequently used in this paper. Define K the number of bits in the tag ID. Each tag has a unique K-bit ID, which can be denoted as $b = b_1 b_2 ... b_K$ with b_i as the i^{th} bit in the ID. Define M the total number of tags to be identified, and we have $0 \leq M \leq 2^K$. A *query tree* is the collection of all possible query prefixes formed in the binary tree structure. Each vertex in the tree represents a *node* or a prefix. A node is *visited* if the reader issues a query using its prefix. An *empty, singleton and collision* node indicates the query represented by the node matches no tag, exactly one tag and multiple tags, respectively. Define the *query traversal path (QTP)* a sequence of all queries used to identify total tags, denoted by $Q = \{q_1, q_2, q_3, ...\}$, where q_i is the i^{th} query used in the arbitration process. Let n_q be the number of queries used, i.e., $n_q = ||Q||$. A *time slot* is the transmission time needed for a reader to issue a query plus the time it takes for the tags to backscatter replies. The total number of time slots consumed in the arbitration equals to the total number of nodes in the query tree visited by the reader, or the length of QTP (i.e., n_q). And *Tag distribution* is the distribution of tag ID's in the query tree. Dense distribution means tag ID's are closely located in the tree. Closer tags usually have longer common prefixes in their ID's.

Ideally, the minimum overhead can be achieved by querying the singleton nodes only. We call this query traversal path the ideal QTP. In practice, achieving truly minimum overhead is hardly possible, since the singleton nodes are dynamically located depending on the tag distribution. The goal of this research is to identify the possible optimal QTP that is close to the ideal one. To reach this goal, the algorithm should be able to predict which level most likely contains the next singleton node, and then quickly traverse there if the QTP is not at such a level currently.

The basic idea of STT is to online learn the tag distribution from the query responses and dynamically adjust the QTP to reach the singleton nodes, and thus significantly reduce the number of empty and collision slots during tag arbitration. To be more specific, if a singleton node is visited, indicating a match with the pattern of the tag distribution, the QTP moves right horizontally to follow this pattern. If a collision (or empty) node is visited, the QTP moves down (or up).

The STT protocol is depicted in Fig. 1. The algorithm starts querying at the root. At the beginning of each time slot, the reader issues a query with an h-bit prefix. For the query at the root, $h = 0$ or a null prefix is used. Depending on the result of tag response, the algorithm takes different movements:

(1) A Collision implies the current QTP is at a level too high and should traverse down by using a longer prefix to reduce the tag responses. In the other words, STT recursively splits the tag IDs in pre-order depth-first way until there is no more collision, similar to QT.

(2) No response indicates the current QTP is at a lower level than the next singleton node and needs to traverse up. Instead of using the pre-order traverse adopted by the existing protocols, here the algorithm moves just one level up to avoid QTP traversing too high, which may lead to unnecessary collisions. However, this rule only applies for the right child. If the empty response comes from a left child, QTP must move horizontally to the right, i.e., visit the right sibling of the empty node. This is to guarantee all the leaf nodes to respond at least once in the arbitration.

An example is provided in Fig. 1 to illustrate the rules: Node U, a right child of Node E, is an empty node. Therefore, after visiting Node U, QTP moves one level up to visit Node F. On the other hand, although an empty tag response is observed at Node M, the QTP must go to Node N first to make sure all

Authorized licensed use limited to: University of Louisiana at Lafayette. Downloaded on August 26, 2009 at 08:24 from IEEE Xplore. Restrictions apply.

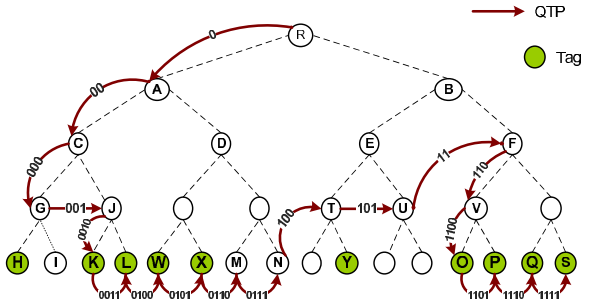


Fig. 1. An example of the STT protocol.

the leaf nodes are covered. When Node N, the right child, is empty, the QTP then moves up.

To select the next query prefix (denote by q_n) after an empty response, the reader first checks the pattern of bit string used in the current query prefix, denoted by $q_c = b_1 b_2 ... b_h$. If $b_h = 0$, indicating the empty node is a left child, the reader simply flips bit b_h and makes $q_n = b_1 b_2 ... b_{h-1} 1$.

If $b_h = 1$, the empty node is a right child and QTP should move up one level. To do that, the reader must first get the number of consecutive 1's from the least significant bit in q_c, denoted by m. That is, $q_c = b_1 b_2 ... b_{h-m-1} \mathbf{0} 1 ... 1$, where$b_{h-m}$ is the right-most 0 in q_c. Then the reader reverses the bit pattern of $b_{h-m} ... b_{h-1}$ and removes the last bit b_h. The reconstructed string, $b_1 b_2 ... b_{h-m-1} 10..0$ is the next query prefix q_n. The details are shown in Lines 5 - 8 in Alg. 1.

If q_c does not have bit 0, the arbitration is finished.

(3) A single response means the QTP has just visited a singleton node. Since the trend of tag distribution has certain continuity, the algorithm predicts the next singleton node will be at the same level. Therefore, the QTP moves in the breath-first order to the immediate right node. As depicted in Fig. 1, after the QTP visits a singleton node, Node O, it moves right in a horizontal way. As a result, Nodes P, Q and S are directly identified without going through the collisions at the upper levels. Once a non-singleton node is visited, the *current trend* is discontinued and the algorithm will adjust the QTP according to the result of tag replies.

The algorithm of STT is sketched in Alg. 1.

The unique feature of STT is how it traverses after the collision is resolved in a sub-tree. Instead of traveling all the way up to the unvisited head node of the right subtree, our design allows STT to adjust its QTP in a local level. As a result, STT does not visit all the collision nodes at the upper levels and effectively eliminates large amount of collision overheads. Such movements are not defined by the conventional concepts of binary tree traversal, because the QTP does not follow any pre-defined traversal order. However, they offer great flexibilities and are extremely efficient in the nodal search/identification process.

III. Performance Evaluation

Now we evaluate and compare the performance of the STT protocol with other protocols in terms of arbitration delay and energy consumptions under various tag densities and

Algorithm 1 The STT protocol.

Require: $q_c = b_1 b_2 ... b_h$ is the current query prefix used. m is the number of consecutive 1's from the least significant bit in q_c.
Ensure: q_n is the prefix in the next query to be issued by the reader.
1: **if** The reader detects a collision node **then**
2: $q_n \leftarrow q_c 0$;
3: **else**
4: $q_n \leftarrow b_1 b_2 ... b_{h-m-1} 1$;
5: **if** The reader detects an empty node **then**
6: **for** $i = 1$ to $m - 1$ **do**
7: $q_n \leftarrow q_n 0$;
8: **end for**
9: **else**
10: //The reader detects a singleton node
11: **for** $i = h - m + 1$ to h **do**
12: $q_n \leftarrow q_n 0$;
13: **end for**
14: **end if**
15: **end if**

distributions. To our best knowledge, no existing papers have considered the types of tag distribution other than uniform. However, different tag distributions can greatly influence the protocol performance. In reality, tags are not always uniformly distributed. One example is the healthcare system that issues each individual a unique RFID tag with the first 15 bits of the ID storing the person's birth date. Assume the patients with heart disorders are increased exponentially by the ages. Then at the cardiology & blood vessels clinic of a given hospital, the patients present approximately follow the geometric distribution. As RFID technology is a promising field, more applications are to be introduced, possibly with different tag distributions.

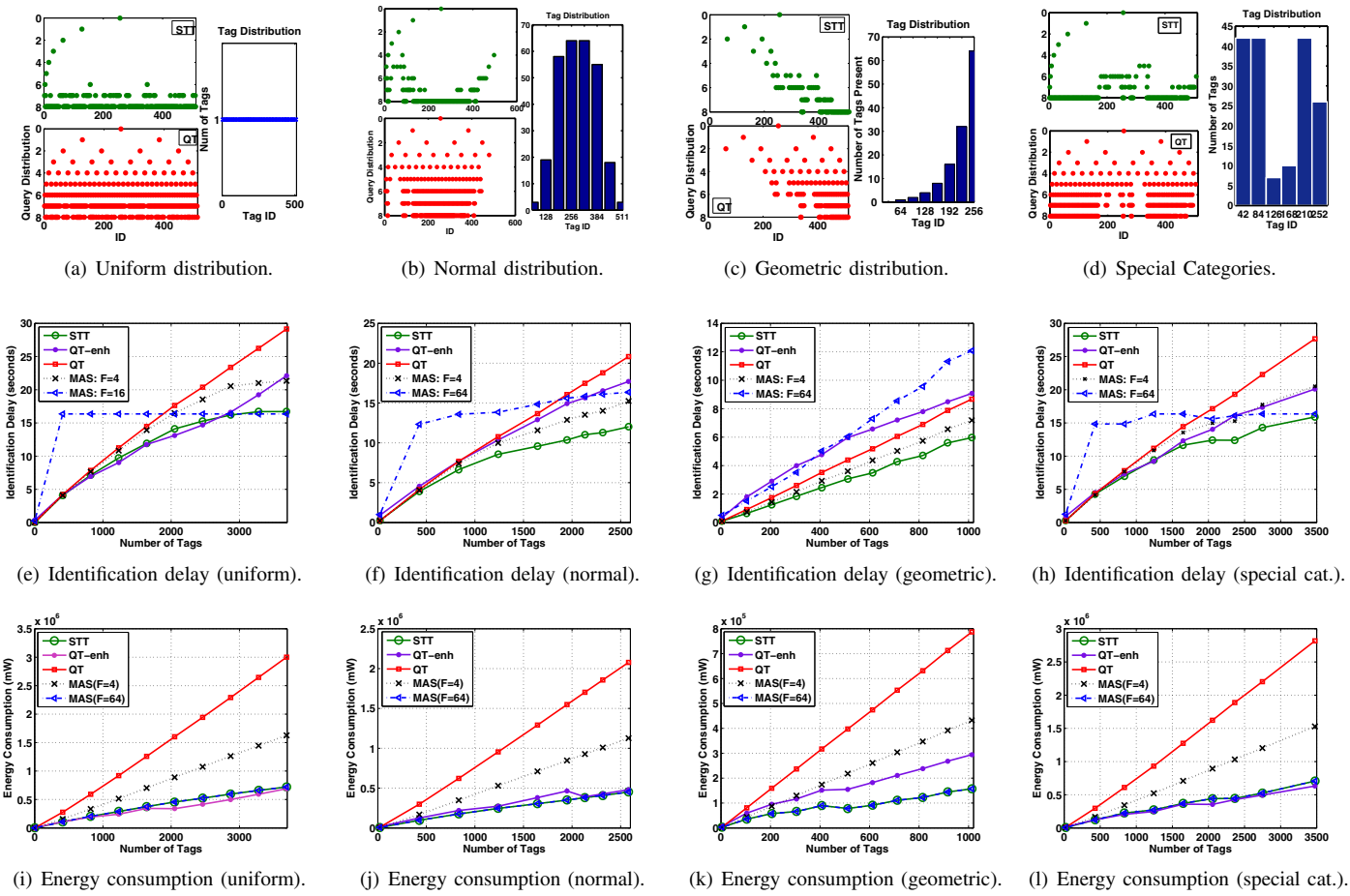
In this research, we consider four different tag distributions: 1) under the uniform distribution, tags are randomly generated with their ID's between $[1, 2^K - 1]$; 2) under the normal distribution, tags are distributed by the normal probability density function, with the mean of 2^{K-1} and the standard deviation of $\frac{2^{K-1}}{3}$; 3) under the geometric distribution, tags are partitioned into several groups by their ID's. Tags within the same group are randomly distributed. The number of tags in a group is approximately twice as that in the preceding group; and 4) under special categories, tags are again partitioned into several groups by their ID's. Tags within the same group are uniformly distributed, while each group has a different density.

Figs. 2(a), 2(b), 2(c) and 2(d) illustrate the tag distribution and the queries used (i.e., the nodes being visited) in the STT protocol and the QT protocol. The Y-axis of the query-distribution figures indicates the tree level, which is also the length of the query prefixes. The X-axis shows the vertical mapping of the queries on the horizontal line, sequentially indexed from 0 to $2^{K+1} - 1$. [1] The tag density is 0.7 (i.e., the number of tags is around 0.7×2^K, where $K = 8$).

The figures clearly demonstrate the STT protocol results in much less queries than QT, because to visit a node at the lower level, QT has to first visit all its ancestors (collision nodes). On the other hand, STT fully takes the advantages of its ability

[1]Every node in the tree has a unique position at the horizontal axis. The number of total nodes in a full K-level binary tree are $2^{K+1} - 1$.

Authorized licensed use limited to: University of Louisiana at Lafayette. Downloaded on August 26, 2009 at 08:24 from IEEE Xplore. Restrictions apply.



(a) Uniform distribution. (b) Normal distribution. (c) Geometric distribution. (d) Special Categories.

(e) Identification delay (uniform). (f) Identification delay (normal). (g) Identification delay (geometric). (h) Identification delay (special cat.).

(i) Energy consumption (uniform). (j) Energy consumption (normal). (k) Energy consumption (geometric). (l) Energy consumption (special cat.).

Fig. 2. Performance comparison of the tree-based protocols.

of dynamic traversal to avoid most of the collisions happened at high levels. From the figures, it can be observed that the traversal paths of STT actually follow the *reverse* patterns of tag distributions: when the tag density is high, STT keeps its path at the lower levels; when the tag density is low, STT moves up. This highlights the *trend-traversal* feature of STT.

The performance of the STT protocol is further evaluated in terms of identification delay and power consumption of active tags with $K = 12$. The number of tags to be arbitrated ranges from around 50 to nearly 4000, indicating various tag densities. We compare the results with the classic QT protocol and its enhancement [8], [9]. Since the enhanced QT protocol needs to know M to select the optimal level, we refer to the result in [11] to calculate the extra delay caused in the estimation of tag population, with a confidence interval of 20%. Our simulations also include the recently proposed MAS protocol [10], which has been briefly described in Sec. I. There are no details about how to choose F in [10], thus we study both small frame size ($F = 4$) and big frame size ($F = 64$) in our simulations, for better understanding on the performance.

Fig. 2(e) illustrates the identification delay under uniform distribution. A time slot of 3.93 ms in fast transmission mode [5] is adopted here to obtain the arbitration delay. We observe that when the tag density is sparse, the performances of all protocols are very close, except that MAS with $F = 64$ has much higher delay due to the excessive empty nodes visited. When the tag density increases, the collision overhead of QT and MAS with $F = 4$ becomes worse, while the delay of the enhanced QT and STT does not increase as much. In the end when the network is highly saturated, the overhead of the enhanced QT protocol quickly increases. On the other hand, the delay for MAS with $F = 64$ becomes constant, because many of the previously wasted empty nodes now become singleton ones when more tags are present, and thus the arbitration does not take much extra time.

Fig. 2(i) presents the energy consumption of active tags, where the power for transmitting is $35mW$ and the power for receiving is $28mW$ [10]. The energy consumption figure for the reader is similar to the delay figure and thus is omitted here. It shows QT and MAS with $F = 4$ consume much more energy than others. The energy consumption of MAS with $F = 64$ is low, which is a trade-off to its long identification delay, as depicted in Fig. 2(e). The enhanced QT protocol slightly outperforms STT, with the help of the given M and the uniform distribution of tag ID's.

The remaining figures in Fig. 2 demonstrate the protocol performances under other distributions. Similar patterns are observed for the MAS and QT protocols. However, the en-

Authorized licensed use limited to: University of Louisiana at Lafayette. Downloaded on August 26, 2009 at 08:24 from IEEE Xplore. Restrictions apply.

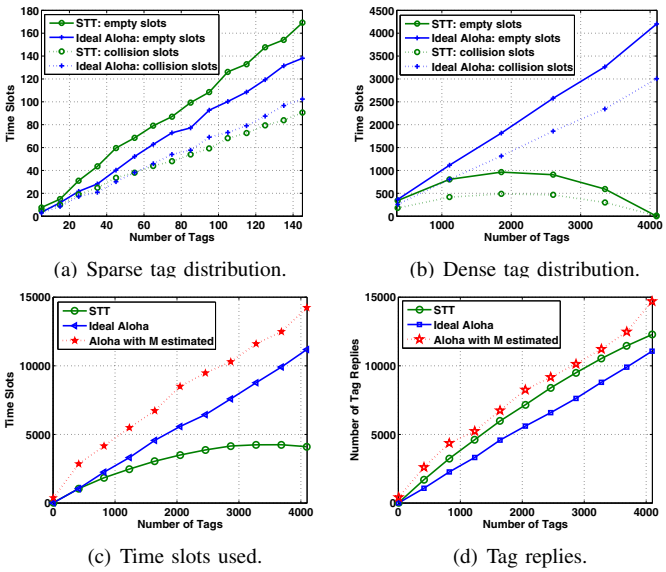
hanced QT protocol has much worse performance when the tag distribution is not uniform, because the protocol is not able to choose an optimal level to start. Especially under the geometric distribution where tag densities in different groups are changing dramatically, the "enhanced" QT protocol even under-performs its original counterpart in terms of delay, as shown in Fig. 2(g). The figures clearly show that STT has the best performance under all scenarios, regardless of the densities of the networks and the tag distributions.

Fig. 3 compares the performance of STT with the frame slotted Aloha protocol. Since the reader does not issue query prefixes in the Aloha-based protocols, we simply assume uniform distribution here. In Figs. 3(a) and 3(b), it is assumed that M is *known* to eliminate the estimation error as well as the overhead incurred in tag population estimation. This allows us to compare STT with the best possible performance of the Aloha protocol, although such an ideal case is rare in reality. Fig. 3(a) illustrates the arbitration delay in terms of time slots used when the tag distribution is sparse, by breaking down the identification overhead in two types: the overhead caused by collision and by empty responses. It is shown that the ideal frame slotted Aloha protocol marginally outperforms STT under sparse topology, due to the extra empty responses incurred in STT. When the tag distribution becomes dense, the performance of Aloha decreases dramatically, yielding excessive collision and empty slots as depicted in Fig. 3(b). In the mean time, STT maintains its efficiency. Especially when the network is more saturated, i.e., the tree is close to be full, the numbers of the empty nodes and the collision nodes visited in STT are actually decreasing. This is because STT moves at the leaf level most of the time, and therefore generates less fluctuations in its QTP. When all tags are present, the default STT protocol takes total $K + M$ slots in arbitration.

Figs. 3(c) and 3(d) further illustrate the impact on the numbers of time slots and tag replies when the overhead of estimating M is considered. The assumption used for the estimation is similar to the one discussed earlier for Fig. 2. It is observed that the estimation overhead significantly impairs the efficiency of the frame slotted Aloha protocol, especially when the network is dense. The ideal Aloha has less tag replies than the STT protocol, while its performance is worse than STT when the estimation overhead is considered. Note that, the tag distribution can greatly impact the estimation accuracy of M. With this into consideration, the actual performance of the frame slotted Aloha protocol will degrade even more.

IV. Concluding Remarks

In this paper, a Smart Trend-Traversal (STT) tag arbitration protocol with online self-learning ability has been proposed, which dynamically issues queries according to online learned tag density and distribution. By following the pattern of tag distribution and taking smart query traversal path, STT significantly saves collisions and reduces the delay and energy consumption comparing with the existing Tree-based and Aloha-based protocols. Our studies have further shown the optimality of STT does not rely on any presumed network conditions,



(a) Sparse tag distribution. (b) Dense tag distribution.

(c) Time slots used. (d) Tag replies.

Fig. 3. Performance comparison of STT and the frame slotted Aloha.

which is in sharp contrast to other available schemes and renders it a highly desirable and practical solution.

References

[1] N. Abramson, "The Aloha System - Another Alternative for Computer Communications," in *Proceedings of AFIPS Conf.*, Apr. 1970, pp. 295–298.

[2] F. Schoute, "Dynamic Frame Length ALOHA," *IEEE Trans. Commun.*, vol. 31, no. 4, pp. 565–568, Apr. 1983.

[3] V. Anantharam, "The Stability Region of the Finite-user Slotted ALOHA Protocol," *IEEE Trans. Inf. Theory*, vol. 37, no. 3, pp. 535 – 540, May 1991.

[4] H. Vogt, "Efficient Object Identification with Passive RFID Tags," in *Proceedings of the First Int'l Conf. on Pervasive Comp.*, Feb. 2002, pp. 98–113.

[5] Philips Semiconductors, "Philips I*Code System Design Guide: Application Note AN00025," May 2002.

[6] EPC Global, Inc., "EPC^{TM} Radio Frequency Identity Protocols Class 1 Generation 2 UHF RFID Protocol for Communications at 860MHz-960MHz Version 1.1.0," Oct. 2007.

[7] J. I. Capetanakis, "Tree Algorithms for Packet Broadcast Channels," *IEEE Trans. Inf. Theory*, vol. IT-25, no. 5, pp. 505–515, Sep. 1979.

[8] C. Law, K. Lee, and K.-Y. Siu, "Efficient Memoryless Protocol for Tag Identification," in *Proceedings of Int'l Workshop on Disc. Alg. and Meth. for Mobile Comp. and Comm.*, Aug. 2000, pp. 75–84.

[9] D. R. Hush and C. Wood, "Analysis of Tree Algorithm for RFID Arbitration," in *Preceedings of IEEE Int'l Symp. on Inform. Theory*, Aug. 1998, p. 107.

[10] V. Namboodiri and L. Gao, "Energy-Aware Tag Anti-Collision Protocols for RFID Systems," in *Proceedings of IEEE Int'l Conf. on Perv. Comp. and Comm. (PerCom)*, Mar. 2007, pp. 23–36.

[11] M. Kodialam and T. Nandagopal, "Fast and Reliable Estimation Schemes in RFID Systems," in *Proceedings of ACM Int'l Conf. on Mobile Comp. and Netw. (MobiCom)*, Sep. 2006, pp. 322–333.

[12] C. Qian, H. Ngan, and Y. Liu, "Cardinality Estimation for Large-scale RFID Systems," in *Preceedings of IEEE Int'l Conf. on Perv. Comp. and Comm. (PerCom)*, Mar. 2008, pp. 30–39.

Authorized licensed use limited to: University of Louisiana at Lafayette. Downloaded on August 26, 2009 at 08:24 from IEEE Xplore. Restrictions apply.