A Fast RFID Identification Protocol with Low Tag Complexity

Hugo Landaluce, Asier Perallos, *Member, IEEE,* and I. J. Garcia Zuazola, *Member, IEEE*

***Abstract*—In this letter, a novel Query window Tree (QwT) protocol is proposed and presented. Based on the Query Tree (QT), a fast anti-collision protocol with low tag complexity (which preserves the memoryless feature of the QT) is achieved. The QwT effectively controls the number of responded bits from tags by adopting a dynamic-size 'window'. As a result, the tags of the proposed QwT deliver exclusively the bits that are required by the window controlled at the reader, rather than sending their full ID. Simulations show that the QwT positively decreases the total number of bits that are transmitted by a tag and as a result mitigates latency.**

***Index Terms*—RFID, anti-collision, tree, tag identification.**

I. Introduction

Radio frequency identification (RFID) technology is becoming popular in assets identification and track-and-trace applications with highly increasing the density of available tags in interrogation zones. The coexistence of tags sharing the same communication channel requires solutions to likely collisions. Anti-collision protocols are sought to mitigate the degradation of the readers bandwidth, its power consumption and delay caused by collisions [1]–[5]. In the literature, two renowned anti-collision protocols have been reported: aloha based and tree based protocols. Aloha based are probabilistic and rely on more complex tags than tree based, which are deterministic. Because we aim to preserve low complexity we concentrate in the latter candidate and attempt to alleviate reader collisions and delays. Smart trend traversal protocol (STT) [1], and the fast tree traversal protocol (FTTP) [2], are among the most recent and advanced tree-based candidates and will be used later for comparative results.

Tree based arbitration protocols, in essence, split colliding tags into subsets, and further split the subsets repeatedly up to the successful response of all the tags that are within the interrogation zone. Tree based are divided into two major groups: binary tree (BT) and query tree (QT) protocols this is according to the need of tags to store information between responses. In the tree splitting protocol (TS) [3], based on BT, each tag has a counter and a random number generator. Collided tags choose a binary number 0-1 randomly, and on-hold tags add +1 to their counter, or -1 for an idle or success response. FTTP [2] is an improved variant of the TS, where collided tags choose a natural number randomly. The maximum of this number is calculated using maximum likelihood estimation and is used to split tags in groups reducing latency, specifically when the idle slots duration is shorter than collision slots duration.

Manuscript received May 14, 2013. The associate editor coordinating the review of this letter and approving it for publication was A. Mitrokotsa.

The authors are with the Deusto Institute of Technology (DeustoTech), University of Deusto, Spain (e-mail: {hlandaluce, perallos, i.j.garcia}@deusto.es).

Digital Object Identifier 10.1109/LCOMM.2013.070913.131111

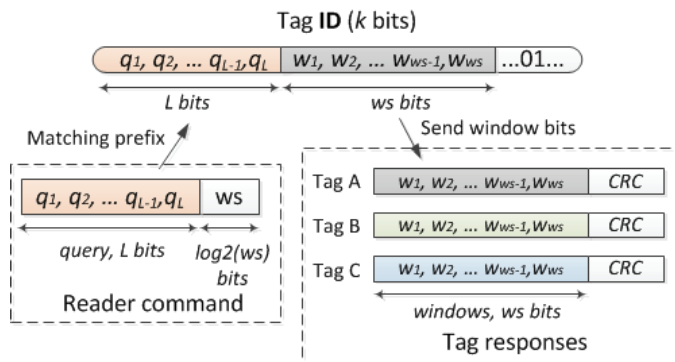

Fig. 1. Format of reader and tag commands.

The QT in contrast is memoryless [4] since the tags do not record counter values. The reader must provide them with a query including a bit string or a prefix. The tags respond with their ID upon successful matching of the prefix with their ID. Where a collision occurs, two new queries are generated adding '0' and '1' to the collided prefix. Queries are discarded when an idle or a successful response is met. QT however, is at the expense of a delay penalty that we aim to reduce in this work. The STT protocol [1] is used to alleviate the number of collisions in QT using a tree traversal path. The collisions are however reduced but not extinguished, and a lot of bits sent by the tags are wasted on collisions. A QT with bit-tracking technology is given in [5]. It uses Manchester coding to estimate tags and to identify collided bits individually. Collisions are alleviated but the complexity of tags is increased. A proposed fast, reliable and memoryless protocol is hereby presented and named query window tree (QwT). The QwT protocol restricts the number of bits transmitted by the tags on a collision, mitigating latency on RFID communication. To corroborate this claim, we simulate and compare results from our proposed QwT with the STT [1] and the non-memoryless FTTP [2].

II. The Proposed QwT Protocol

A. The Adaptive Window

We define the window as the bits transmitted by a tag on a slot. The window is beneficial with the implicit assumption of a perfect synchronization among window responses. The adaptive window performance is now described, Fig. 1. The reader transmits a command including a query of length L bits and a window size parameter (ws). Tags matching a readers query q_1,q_2...q_L where $q_i \in \{0,1\}$ and $L > 0$, respond exclusively the bits specified by ws; $w_1, w_2 \ldots w_{ws}$, where $w_i \in \{0,1\}$, $0 < ws < k$ and k is the length of the tag ID. Tag responses are received by the reader as a unique response

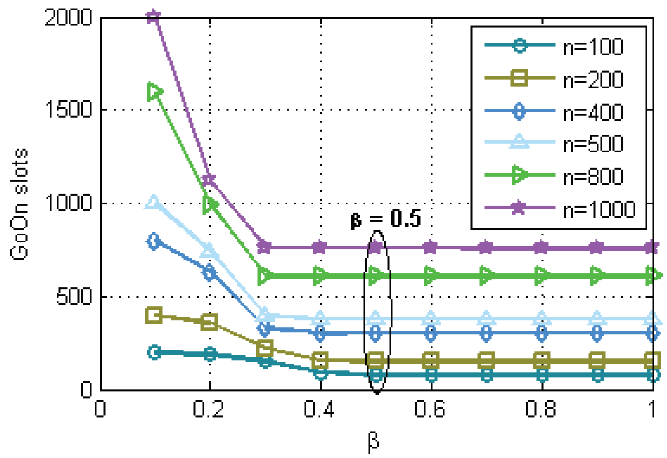

Fig. 2. Number of go-On slots used by the QwT for different β values and number of tags in the interrogation zone, n.

of ws bits and an appended cyclic redundancy check (CRC). Four possible slot status can happen:

- *Idle slot*: when the reader queries and receives no tag response.
- *Collision slot*: when several tags respond different window bits the received CRC will not match the received window at the reader.
- *Go-On slot*: when the reader receives corresponding windows of tags simultaneously with a positive CRC checksum response and obeys the following condition $L + ws < k$.
- *Success slot*: when a single tag response is received and $L + ws = k$.

The window alleviates tags from transmitting large number of bits upon a collision. This is very positive since tree based protocols are prone to high number of collisions, especially at the beginning of the identification cycle [4]. The first transmitted queries of low L values can lead to possible collisions. Therefore, a low ws value is preferred so that matching tags could transmit the least possible number of bits. In contrast, whilst the L increases, the likelihood of collision decreases. Low ws values can cause the number of go-On slots to increase, because the transmission of additional reader queries are required to obtain the full tag ID. Thus, ws must be increased to counteract that. A heuristic function which links L to ws (here as $f(L)$) is given in (1), where β is an adjustable parameter.

$$f(L) = k(1 - e^{-\beta L}), \quad 0 < L \leq k \tag{1}$$

How ws is adjusted is given in Section II-B. The number of go-On slots used by the QwT for different β values and various groups of tags, n, is depicted in Fig. 2. $\beta = 0.5$ is selected for its low go-On slots rate. This is a practiced deduction to reduce the number of bits transmitted per tag and consequently, leads to a low latency RFID system; this is corroborated and presented in Section III.

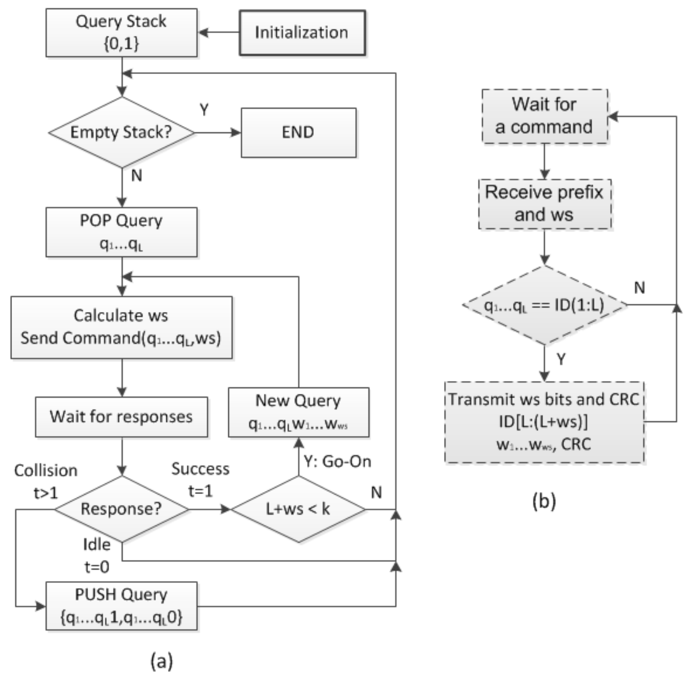

Fig. 3. Flow chart of the proposed QwT protocol: (a) for reader, (b) for tags.

B. The algorithm

We now describe the algorithm involved in our proposed QwT. As inherited from QT, it follows a similar procedure but with the additional window method described in Section II-A. During a cycle, the reader sends a query with adjusted ws parameter on every slot. Thus, QwT tags only need to compare their ID with the received query, and tags matching that query send a bit string $w_1, w_2 \dots w_{ws}$, of ws bits maintaining the memoryless feature. Fig. 3 shows the flow chart of the proposed QwT. The reader and tag sequences are respectively given Fig. 3.(a) and (b). First, Fig. 3.(a), the reader initializes by pushing two queries (0;1) into an empty stack - Last Input First Output (LIFO). The reader then pops a query and subsequently calculates ws using (2). For the initial ws calculation, it uses $ws = 1$; in other consecutive cases the reader needs the calculated ws and L of the immediately earlier slot (ws_{i-1}, L_{i-1}). Whether the latest slot, $i-1$, was an idle ($t = 0$), a collision ($t > 1$), a success ($t = 1$) or a go-On ($t = 1$), needs to be reflected in (2), where t is the number of guessed responses to the latest query (L_{i-1}).

$$ws_i = \begin{cases} f(L_i), & t = 1, L_{i-1} < L_i \\ ws_{i-1}, & t \neq 1, L_{i-1} < L_i \\ 1, & t \geq 0, L_{i-1} \geq L_i \end{cases} \tag{2}$$

Once ws is calculated, a new command (including the query and ws) is sent to the tags and awaits for tags responses, Fig. 3.(b). Tags matching the query respond the window of the remaining bits $w_1, w_2 \dots w_{ws}$, and appended CRC. On arrival, the reader, Fig. 3.(a), searches for possible collisions and that happens when $t > 1$. If so, two further queries are made $q_1, q_2 \dots q_L$'1' and $q_1, q_2 \dots q_L$'0' (they add a '1' and a '0'). For cases when an empty window is received,

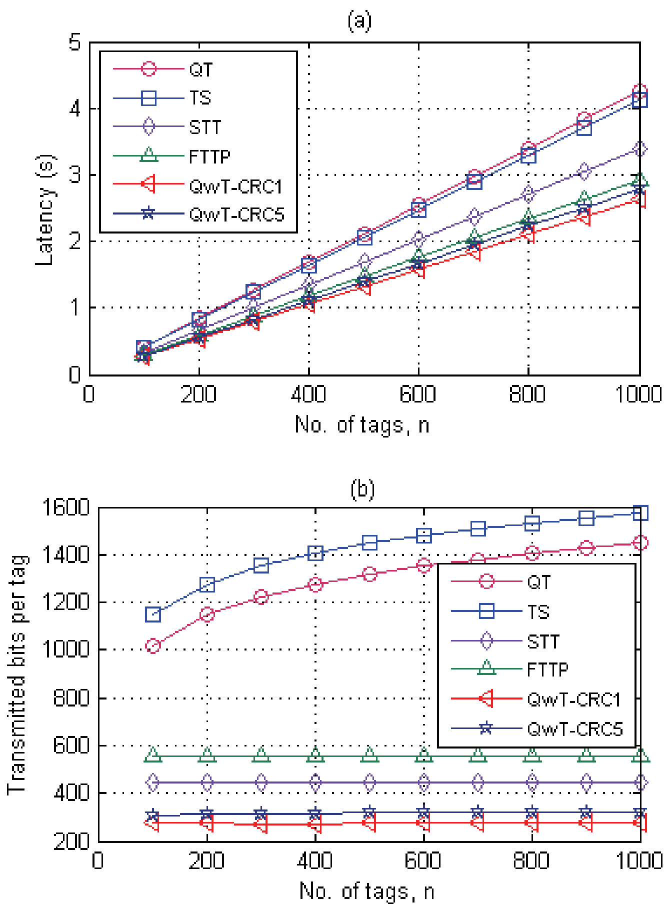

Fig. 4. Performance of the QwT: (a) latency, (b) transmitted bits per tag.

the transmitted query is neglected and denoted as $t = 0$. In addition, when the received CRC is positively checked, the window is completely understood and denoted as $t = 1$; this means at least one tag responded successfully. The reader checks if the lengths of the query sent and window received are enough to complete the tag ID length. If so ($L + ws = k$), the responding tag is successfully identified, and if $L + ws < k$, the slot will be a go-On; this implies a new query of format $q_1, q_2 \dots q_L w_1, w_2 \dots w_{ws}$. In a close-loop, the procedure sequences repeatedly until it gets the empty stack.

III. Simulation Results

Simulation results are first presented and the section concludes with an evaluation of the outcomes. For the simulation, we consider the proposed QwT with appended CRCs of different lengths and compare it to current anti-collision protocols: STT [1], FTTP [2], TS [3], and QT [4]. A reader and a varying number of tags, from 100 to 1000 units, are used on every simulation, and as in [1]–[5] a non-impaired channel and no capture effect was considered. The tags are uniformly distributed and k is assumed 128 bits. The tag IDs are dynamically generated for every simulation with varying random seed values. The reader data rate is set to 160 kbps and the tag data rate to 80 kbps. The simulated responses were averaged over 500 iterations for accuracy in the results.

Many anti-collision protocols are verified using the number of slots, but not the slot duration (time) and length (bits) [2], e.g. the time of a collided slot is longer than for an idle or a successful slot and for this reason we use the latency and the number of transmitted bits in this work. Fig. 4.(a) and (b) show the performance of the QwT in latency and number of transmitted bits per tag respectively. Depicted results show evidence of improved latency for the proposed QwT protocol, especially in dense tag environments. QwT was best when the bits contained in the CRC are at its low, QwT-CRC1 (uses a single bit). However, the impact of a CRC with a higher number of bits, when QwT-CRC5 (uses 5 bits) is yet away from the other protocols in latency and transmitted bits per tag. As a rule of thumb, QwT window responses with appended CRCs of higher number of bits are more accurate, but conduct to a lower performance. Besides, memoryless feature leads QT and STT to need larger reader commands than TS and FTTP, which makes FTTP to outperform QT and STT in latency, Fig. 4.(a). Non-memoryless TS, however, needs more bits per tag than the other protocols, Fig. 4.(b), and it can only improve QT in latency. The results suggest that QwT is a dexterity and memoryless protocol, which using an adaptive window, a relative low number of transmitted bits per tag is completed. The proposed QwT protocol is therefore a suitable candidate where fast identification with low complexity is sought.

IV. Conclusion

A novel and simple protocol for fast RFID tag identification has been presented. The contributing QwT protocol makes effective use of an adaptive window, controlled at the side of the reader, to restrict the total number of bits that are responded from tags and decreases the number of transmitted bits as well as the latency. Simulated results proved the dexterity of the QwT and outperformed comparative protocols, the reported protocol can be considered as a good RFID anti-collision candidate. Future research will look at the possibility of implementing compliant bit-tracking algorithm [5] to not only decrease tag bits but also collisions of the proposed QwT.

References

[1] L. Pan and H. Wu, "Smart trend-traversal protocol for rfid tag arbitration," *IEEE Trans. Wireless Commun.*, vol. 10, no. 11, pp. 3565–3569, 2011.

[2] J. Choi, I. Lee, D.-Z. Du, and W. Lee, "FTTP: a fast tree traversal protocol for efficient tag identification in rfid networks," *IEEE Commun. Lett.*, vol. 14, no. 8, pp. 713–715, 2010.

[3] D. R. Hush and C. Wood, "Analysis of tree algorithms for RFID arbitration," in *Proc. 1998 IEEE International Symposium on Information Theory*, p. 107.

[4] C. Law, K. Lee, and K.-Y. Siu, "Efficient memoryless protocol for tag identification," in *Proc. 2000 International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pp. 75–84. Available: http://doi.acm.org/10.1145/345848.345865

[5] Y. Lai, L. Hsiao, H. Chen, C. Lai, and J. Lin, "A novel query tree protocol with bit tracking in rfid tag identification," *IEEE Trans. Mobile Computing*, vol. 99, no. PrePrints, 2012.