Simplified computation in memoryless anti-collision RFID identification protocols

H. Landaluce, A. Perallos, L. Bengtsson and I.J. Garcia Zuazola

A memoryless-based Collision window Tree plus (CwT+) protocol for simplified computation in anti-collision radio frequency identification (RFID) is proposed and presented. The CwT+ makes effective use of a *threshold* to accurately enhance bit-tracking and in turn lowers the identification time of the communication response. The *threshold* limits the bitstream of responding tags in an accurate fashion for simplified computation in the interrogation procedure. Simulation results show the outperformance of the CwT+ compared with earlier protocols.

Introduction: Low computation is sought for anti-collision radio frequency identification (RFID) protocols in dense tag/transponder interrogation zones. The practice is accomplished by a smart arbitration of simultaneously responding tags, alleviating interference (garbled messages) to individually identify tags with reduced identification time; this relaxes the degradation of the reader's bandwidth [1–4].

Although the literature reports two competing methods for the protocol proposals, the probabilistic and the deterministic, the present authors use the latter, since unlike the former, it allows for memoryless (less complex) tags [4]. Among the deterministic, the following tree-based memoryless protocols are the foremost. The Query Tree (QT) protocol, a low-cost low-complex solution as a result of exempt randomisation, does not require a random number generator in the tag, but suffers from inherent multiple collisions [1].

An attempt to alleviate collisions and delays is the adoption of bit-tracking technologies (the reader's ability to detect the locations of collided bits in a collision slot). The imminent Collision Tree (CT) protocol realisation requires synchronised tag responses [2] and eradicates idle slots. A more advanced technique has been presented in [3], where an Optimal Query Tracking Tree (OQTT) protocol uses bit-tracking and an incorporated estimator (statistical information from collided bits in a slot to estimate the number of available tags) to improve general performance.

More recently, the present authors have presented the Query window Tree (QwT) protocol in [4] for the purpose, which essentially adopts an adaptive dynamic-size window (defined as a frame containing the number of transmitting bits of a tag in a slot) at the reader-side to decrease the responded bits of tags. Although this is positive to efficient interrogations (round trip computation), the inherent error-detecting codes can increase complexity and it will be shown that collisions are yet further manageable for simplified computation by the adoption of the hereby proposed Collision window Tree plus (CwT+) protocol.

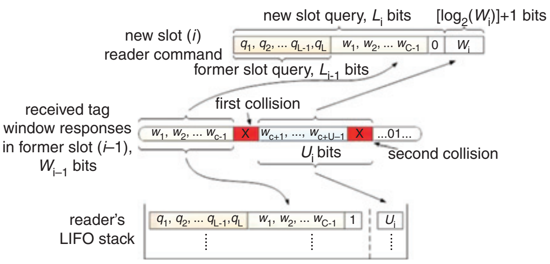

Fig. 1 *Format of proposed CwT+ protocol upon a collision*

Proposed CwT+ protocol: Based on a *threshold* to accurately enhance bit-tracking, reduce identification time and improve bit computation in the interrogation procedure, an original CwT+ is presented in this Section. As an extension to CT, the *threshold* allows the reader to have the ability to detect a pair of locations of collided bits in a collision slot and with an evolved bit-tracking. Used in conjunction with a sliding window, *W*, the number of bits that are employed between the immediately succeeding first collision and the including second collision of that slot (Fig. 1) is denoted by the *threshold* value *U* and *i* represents the slot number. We now proceed with the description of the proposed protocol of Fig. 1. Upon the first collision, the reader generates two new queries to re-interrogate the tags; and the second collision is aimed to inform the likely location of a foreseen collision in the consecutive slot. *U* adequately back informs the reader with the bitstream required by subsequent tag responses – this is valid for the two queries right after the first collision; the resulting responded slots append (when applicable) a collision sign, and if so two new queries are subsequently generated for the reception of necessary bits exclusively. If a single or no collision materialises, the lowermost feasible value of $U = 1$ is used; this turns into a minimum number of transmitted bits per tag (less wasted no.) and a simplified round trip computation of bits as a result.

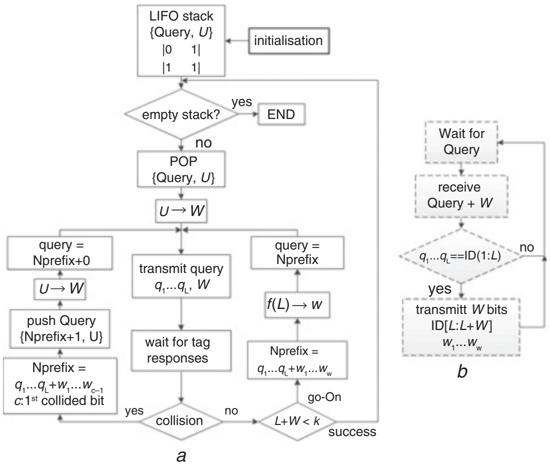

Fig. 2 *Flowchart of proposed CwT+ protocol*

a For reader
b For tags

The *threshold* maintains previous protocols tag synchronisation responses and relaxes from the utilisation of error-detecting codes, the cyclic redundancy check (CRC) and the storage of the associated data – this is perceived as an improvement towards recognising collisions in lower time-delay and therefore an improved computation efficiency (reduced number of round trip bits in lower time).

The $W(W \in \mathbb{N})$, however, is conditioned to the support of go-On slots (later defined) till a successful tag identification is made and therefore a higher number of reader bits from multiple go-On slots situations is common; this is because the tag responses, $w_1, \ldots, w_W$ (see Fig. 1), are in fact divisions of the full tag identifier (tag-ID), and supplementary slots are needed to be able to gain the full ID of a single tag (*k* bits). This higher number of slots (from small size *W*) can decrease the overall performance as it raises the number of bits transmitted by the reader. On the contrary, a larger size *W* allows for fewer slots, by increasing the quantity of bits of tags in the response mode – yet the number of go-On slots is convenient to decrease. For the use of an enhanced bit-tracking technique, the *W* must have room to accommodate the necessary bits, but be well-adjusted in size to remain small (a large *W* produces the contrary effect). Since the received *W* bits immediately preceding the first collision (prior to the collided bit) are used for the formation of the succeeding queries, $q_1, \ldots, q_L, w_1, \ldots, w_{c-1}$, 0 and $q_1, \ldots, q_L, w_1, \ldots, w_{c-1}$, 1 (see Fig. 1), the instigation for defining a truthful *W* size.

Identification procedure: The process of identification for the proposed CwT+ protocol in the form of a flowchart is depicted in Fig. 2 and described subsequently. The course starts with the initialisation of the reader (Fig. 2*a*), which pushes opening queries (0, 1), both with a *threshold* value $U = 1$, into an implemented last-in first-out (LIFO) pre-emptied stack. The reader pops a query of *L* bits from the stack and appends the corresponding *U* as *W* ($U \rightarrow W$) to the new reader command (see Fig. 1); the *W* is made of $\lfloor \log_2 W \rfloor + 1$ bits, where $\lfloor\ \rfloor$ rounds to the nearest integer towards $-\infty$. The tag sequence is given in Fig. 2*b* and described subsequently. Tags matching the reader query $q_1, \ldots, q_L$, respond exclusively to their window frame (of *W* bits); $w_1, \ldots, w_W$, where $q_i, w_i \in \{0, 1\}$ and $0 < W < k$. The reader receives the responses and, depending whether a collision occurs, indicates with $t = 1$ when affirmative and $t = 0$ otherwise. Three possible slot statuses can happen:

- collision slot ($t = 1$): when at least one collided bit is met.

• go-On slot ($t = 0$): when no collision is found but the reader is unable to identify the tag due to condition ($L + W < k$).
• Success slot ($t = 0$): when a response of a single tag is received and the tag-ID acknowledged; implies condition ($L + W = k$).

The practice for the three cases above is now described. Upon a collision slot, two supplementary queries are generated with notations $q_1, \ldots, q_L, w_1, \ldots, w_{c-1}$, 0 and $q_1, \ldots, q_L, w_1, \ldots, w_{c-1}$, 1; basically made of the former query $q_1, \ldots, q_L$ appended with either $w_1, \ldots, w_{c-1}$, 0 or $w_1, \ldots, w_{c-1}$, 1, which indicates the location of the collision (originating collided bit) with the suffix c. From these supplementary queries, the first is sent downlink as a new reader command for interrogation and the second (affixes U) is stored in the LIFO stack for later use; W is updated in the new reader command ($U \rightarrow W$). Upon a go-On slot, this is composed of a new generated query with sequence $q_1, \ldots, q_L, w_1, \ldots, w_W$; that is, the immediately received window ($w_1, \ldots, w_W$) affixed to the former query ($q_1, \ldots, q_L$). Upon a success slot, the reader retrieves (pops) the awaiting query (with affixed U) from the LIFO stack and the procedure sequences until it gets the empty stack.

The update of W, for the actual slot i is found using (1) and denoted by W_i. The resultant U_i is the planned *threshold* according to the latest received W_{i-1}, which defines the reconfigured size of the new generated query (length L_i) (Fig. 1). The L_i and L_{i-1} include indication as to whether the slot was successful, collided or go-On. The awaiting query from the LIFO stack is retrieved (pop) under the success condition $L_{i-1} \geq L_i$. Upon a collision or a go-On slot, condition $Li - 1 < L_i$ applies. $t = 1$ indicates the collision case and $U_i = W_i$ is assigned; for the go-On slot, $t = 0$ and the heuristic function $f(L_i)$ (2) is utilised:

$$W_i = \begin{cases} U_i, & L_{i-1} \geq L_i \\ U_i, & t = 1, L_{i-1} < L_i \\ f(L_i), & t = 0, L_{i-1} < L_i \end{cases} \quad (1)$$

$$f(L_i) = \frac{1}{1 - \frac{\beta}{k}} \times L_i, \quad 0 < L_i \leq k \quad (2)$$

β defines an appropriate value for decreasing the occurrence of go-On slots when the tag bitstream is considerably low. Fig. 3 shows the case where the lower number of transmitting bits per tag, (Fig. 3*a*) when the number of go-On slots is at its lowest, (Fig. 3*b*): that is $\beta = 124$ for n number of tags in the interrogation zone. Successive ith slots of W (W_i) are calculated using (1), and $k - L_i$ is the completing limiter.

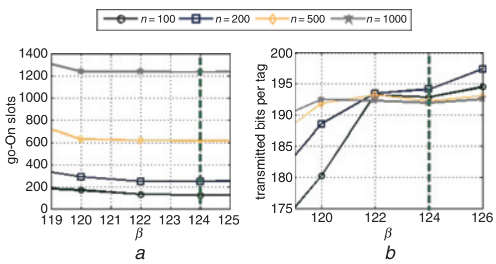

Fig. 3 *Selected β for n number of tags in interrogation zone*
a Number of go-On slots
b Transmitted bits per tag

Simulation results: To validate the proposed CwT+, simulation responses are compared with the memoryless anti-collision protocol CT [2], the more recent OQTT [3] and the QwT [4]. For the simulations, a reader and a varying number of tags (both set to a data rate of 160 kbit/s), from 100 to 1000 units, were used on every simulation, and as in [1–4] the wireless channel had no capture effect (when in a collision, slots are deciphered by the reader's capability of interpreting those being stronger in power strength). The tag-IDs were $k = 128$ bits length, uniformly distributed and dynamically generated for every simulation with varying random seed values, and the simulated responses were averaged over 100 iterations for accuracy.

Fig. 4 shows the comparative performance of these protocols and shows evidence of an outperforming CwT+ in transmitted bits per tag (Fig. 4*a*), round trip bit computation (Fig. 4*b*) and identification time (Fig. 4*c*). Compared with the QwT protocol, as the CwT+ uses the *threshold U*, the non-need of error-detecting codes (i.e.: CRC) itself leads to a reduced number of round trip bits in lower time, and therefore an enhanced computation efficiency. Compared with the OQTT protocol, the CwT+ does not interrogate tags by estimations and thus transmits fewer bits per tag response. That for the CT went beyond 600 bits/tag and is not shown for brevity. The owing performance of the CwT+ is principally given by the integrated *threshold* technique with accommodated sliding window and is corroborated by the results depicted in Fig. 4. Controversially, a low number of bits per tag increases the required go-On slots and the associated reader bitstream for the CwT+ protocol, but to overcome the constraint, (2) confines these bits to a low number. Unlike QwT and CwT+, the OQTT and CT protocols require fewer slots (without go-On), but they suffer from an inherently higher number of transmitting bits per tag. In addition, like the CT, the CwT+ protocol eradicates idle slots for fast computation. The total computation, in regard to average percentage efficiency, is enhanced by 6, 21 and 27%, compared with the OQTT, QwT and CT, respectively.

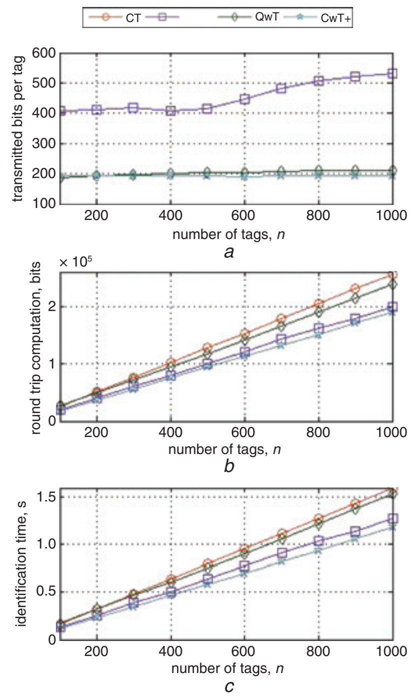

Fig. 4 *Performance of CwT+*
a Transmitted bits per tag
b Round trip bit computation
c Identification time

Conclusion: A *threshold* technique for simplified computation in memoryless anti-collision RFID identification protocols has been proposed and presented as the CwT+ protocol. The technique adjusted conveniently a sliding window with an evolved bit-tracking to decrease the transmitted bits per tag and lowered the identification time. Simulated results are presented which corroborated the CwT+ performance over other recent memoryless anti-collision protocols.

15 May 2014
doi: 10.1049/el.2014.1552
One or more of the Figures in this Letter are available in colour online.

H. Landaluce, A. Perallos and I.J. Garcia Zuazola (*DeustoTech, Spain*)

E-mail: hlandaluce@deusto.es

L. Bengtsson (*Chalmers University of Technology, Sweden*)

References

1 Law, C., Lee, K., and Siu, K.-Y.: 'Efficient memoryless protocol for tag identification'. Proc. 4th Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, Cambridge, MA, USA, 2000, pp. 75–84
2 Jia, X., Feng, Q., and Ma, C.: 'An efficient anti-collision protocol for RFID tag identification', *IEEE Commun. Lett.* 2010, **14**, (11), pp. 1014–1016

3 Lai, Y.-C., Hsiao, L.-Y., Chen, H.-J., Lai, C.-N., and Lin, J.-W.: 'A novel query tree protocol with bit tracking in RFID tag identification', *IEEE Trans. Mobile Comput.*, 2013, **12**, (10), pp. 2063–2075

4 Landaluce, H., Perallos, A., and Zuazola, I.J.G.: 'A fast RFID identification protocol with low tag complexity', *IEEE Commun. Lett.*, 2013, **17**, (9), pp. 1704–1706