Hindawi
Wireless Communications and Mobile Computing
Volume 2017, Article ID 2135182, 10 pages
https://doi.org/10.1155/2017/2135182

WILEY | Hindawi

Research Article

A High Throughput Anticollision Protocol to Decrease the Energy Consumption in a Passive RFID System

Hugo Landaluce,[1,2] Laura Arjona,[1,2] Asier Perallos,[1,2] Lars Bengtsson,[3] and Nikola Cmiljanic[1,2]

[1]*DeustoTech-Deusto Foundation, Avda. Universidades, 48007 Bilbao, Spain*
[2]*University of Deusto, Avda. Universidades, 48007 Bilbao, Spain*
[3]*Department of Computer Science and Engineering, Chalmers University of Technology, Göteborg, Sweden*

Correspondence should be addressed to Hugo Landaluce; hlandaluce@deusto.es

Received 28 April 2017; Accepted 12 October 2017; Published 8 November 2017

Academic Editor: Haiyu Huang

One of the main existing problems in Radio Frequency Identification (RFID) technology is the tag collision problem. When several tags try to respond to the reader under the coverage of the same reader antenna their messages collide, degrading bandwidth and increasing the number of transmitted bits. An anticollision protocol, based on the classical Binary Tree (BT) protocol, with the ability to decrease the number of bits transmitted by the reader and the tags, is proposed here. Simulations results show that the proposed protocol increases the throughput with respect to other recent state-of-the-art protocols while keeping a low energy consumption of a passive RFID system.

1. Introduction

The Radio Frequency Identification (RFID) technology use is being increased in applications where autoidentification methods are needed [1–5]. RFID is used to identify codes stored in devices, called tags, using radio frequency waves. These tags are attached to different objects which can be identified without an existing line of sight. RFID is a low intrusive technology which can be easily adapted to the Internet of Things [2].

An RFID system is basically composed of two elements: a reader and one or more tags. The reader is an electronic device with a RF module, a control unit, and one or more antennas which establish a bidirectional communication with the second device, the tags; tags include an IC-chip and an antenna and are attached to the object to be identified. Tags can be active (battery-powered) or passive (obtain power from reader's signal). Passive tags are widely used due to their low price and the absence of batteries; however, they have a lower coverage than the active ones. This paper is focused on passive RFID systems consisting of a reader and different numbers of passive tags.

Typically, an RFID system may contain several tags coexisting and transmitting to the reader at the same time using the same channel (the air). This fact can cause the cancellation of tags' responses. The reader may not be able to decode their waveforms and tags will be forced to retransmit their messages causing a decrease in the time needed to be identified and an increase in the energy consumed by the reader. This problem is called the tag collision problem [1].

This problem is faced using anticollision protocols. Several protocols are presented in the literature and can be mainly classified into Aloha based and tree based protocols [3]. Aloha based protocols, classified as probabilistic since tags' responses, are randomly organized and distribute responses among slots. The current standard EPC global Class 1 Gen 2 [6] belongs to this category. Research in these types of protocols is focused on the optimal distribution of tags' responses in a timeline. Tree based protocols, on the other hand, are classified as deterministic since they are supposed to read all the tags in the interrogation zone [3]. These protocols usually split the set of tags upon collisions until achieving a successful response from all the tags; however, some recent solutions provide an estimation phase to define

initial subsets which are easier to be identified by the reader [7, 8].

The energy consumption of an anticollision protocol has been directly related to active RFID systems due to the use of batteries in active tags [9–12]. Passive RFID systems are increasingly being used with portable readers which means that the energy consumed by the reader is becoming important, and the anticollision protocol used affects it [13]. A deep analysis of the energy consumption of anticollision protocols in passive RFID is given in [14]. The window procedure is presented and applied to the query tree (QT) protocol [15], in the query window tree protocol (QwT), to decrease the energy consumed by a passive RFID system. QwT manages to decrease the number of bits transmitted by the tags which produces a significant decrease in the energy consumed by the reader. The use of the window forces the anticollision protocol to add a new type of slot, called go-on slot, which increases the total number of slots and, therefore, the total number of bits transmitted by the reader. This is specially problematic when the reader is asking the tags for the last part of their identification code (ID), since it needs to transmit longer commands [14].

This paper presents the contribution of a new protocol called binary window tree (BwT). This protocol proposes adding an additional internal counter to the tags which indicates the last bit of their transmitted ID. This modification allows the adoption of the window in BT. Additionally, the reader is adapted to working with the window which is tuned to dynamically adapt its size during the procedure of the identification using tags with memory. This protocol is later compared to other state-of-the-art protocols in the literature. The results show that the novel proposed protocol increases the throughput of the RFID system while it saves energy using passive tags.

Subsequently, the rest of the paper is organized as follows. Section 2 provides background information and related work on anticollision protocols. Section 3 presents the proposed BwT protocol. In Section 4 the simulation results of the comparison with state-of-the-art protocols are presented. And Section 5 concludes this paper.

2. Background

In order to properly understand the proposed work, some concept definitions are introduced here:

(i) A *slot* determines the period of time that includes a reader command and a tag response. This time is usually fixed, but some state-of-the-art works consider it dynamic, as is the case of this work. Three types of slots are usually considered upon the number of tags' responses: a collision occurs when more than one tag responds in the same slot period; a success slot is given when only one tag responds to a reader command; and an idle slot occurs when no tag responds to a reader command.

(ii) An *interrogation cycle* is the period of time the reader needs to identify the whole set of available tags. An interrogation cycle is composed of several slots.

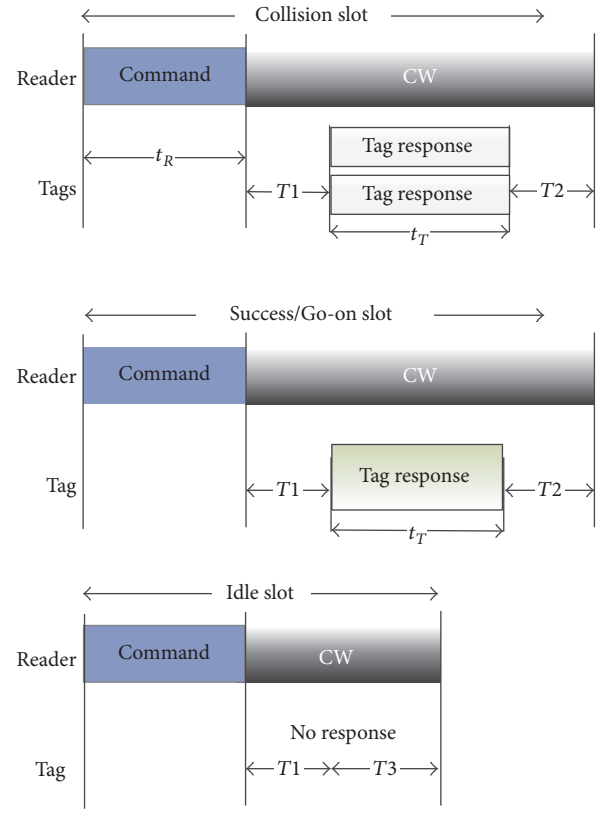


Figure 1: Example of a collision/idle/success/go-on slot in the transmission model used.

(iii) The metric *throughput* is conceived as the number of read tags in the unit of time.

Using these main concepts the transmission model between the reader and the tags can be explained. This model assumes an ideal channel for transmissions. No physical-layer and no capture effect (when the reader decodes only the tag response with the highest power in a collision slot) are considered here. All tags in the antenna range remain correctly energized during the interrogation cycle; all tags' responses are synchronized; and lastly, a collision occurs only when two different messages or bits are simultaneously transmitted since error transmissions are not considered. These assumptions are extensively made in other similar anticollision protocol works proposed [7, 8, 14, 16]. The transmission model is explained below.

2.1. Transmission Model. The transmission model used is defined in [6], which corresponds to the EPC global C1G2 standard.

Figure 1 shows the link timing of the three types of slots mentioned (collision, idle, and success) and the go-on slot explained below. Also, in Definition of Symbols and Variables a list of all the variables used in the paper is included. The reader starts transmissions using commands during time t_R. The reader holds the RF downlink carrier, also called continuous wave CW, so that tags can harvest energy and respond with their ID. After every reader command there is a

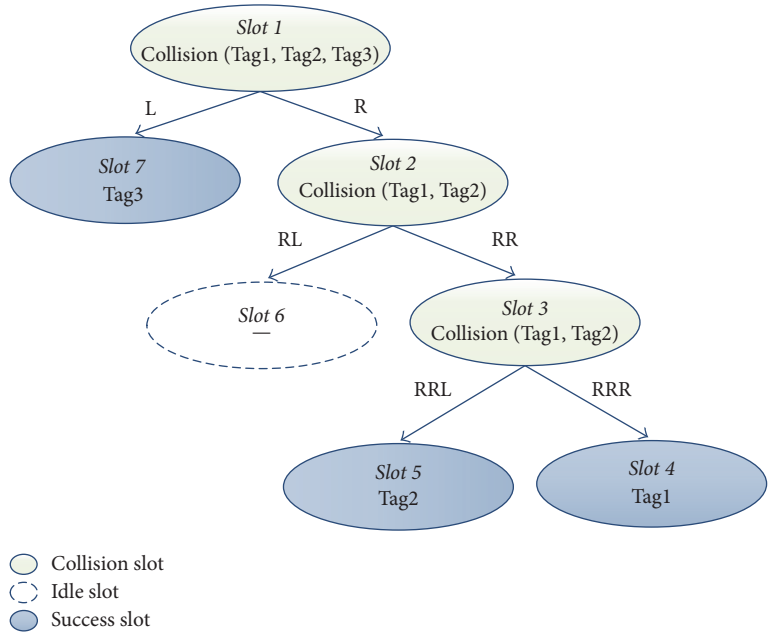


Figure 2: Example of an identification cycle with BT.

time T_1 needed for the tags to generate their responses and a time T_2 needed for the reader to receive all the transmissions; a slot will be considered idle when the reader waits for the tags' responses for a time T_3. Additionally, the tag's response is produced during time t_T.

With respect to the energy consumption model, E represents the energy consumed by the reader. It is a function of the time it spends transmitting and receiving information. An energy model is proposed where the reader transmits the command and the CW to power up passive tags with power P_{tx}. In addition, the reader will require extra power P_{rx} when receiving data from the tags. Therefore, the expression used to calculate the total energy consumed during the interrogation cycle is shown in

$$\begin{aligned} E &= E_c + E_i + E_s \\ &= \sum_{j=0}^{c+s} \left[P_{tx} \times \left(t_{Rj} + T_1 + t_{Tj} + T_2 \right) + P_{rx} \times t_{Tj} \right] \\ &\quad + \sum_{j=0}^{i} \left[P_{tx} \times \left(t_{Rj} + T_1 + T_3 \right) \right]. \end{aligned} \tag{1}$$

Here, E_c, E_s, and E_i represent the energy consumed during collision, success, and idle slots and c, s, and i represent the number of collision, success, and idle slots, respectively.

2.2. Related Work. Here, some of the most relevant tree based protocols in the literature are presented. This will later be simulated and compared in Section 4.

2.2.1. Binary Tree Protocol. The Binary Tree (BT) protocol was firstly applied to RFID by Hush and Wood in [17]. It uses a tree to organize and identify all the tags into the reader's antenna range. Every time a collision occurs between tags' responses the responding tags are split into two different subgroups. These subgroups become increasingly smaller until they are split into two tags, which can therefore be identified (see Figure 2).

The reader consecutively interrogates all these subgroups and tags outside these groups which wait until their subgroup is chosen for the interaction with the reader. Every time the protocol reaches a leave of the tree, the reader identifies the tag and goes back to the last subgroup produced which starts to be split in a similar manner.

Bertsekas and Gallager then included an internal counter on every tag which they modify upon the reader commands [18]. The reader can indicate the three possible slot states. Upon an idle slot, tags decrease their counters by 1; upon a collision slot, transmitting tags choose between 0 and 1 randomly, and waiting tags increase their counter by 1; upon a success, the transmitting tag goes to sleep mode and the rest of the tags decrease their counters by 1.

2.2.2. Fast Tree Traversal Protocol. Choi et al. proposed the Fast Tree Traversal Protocol (FTTP) [7] that uses the Maximum Likelihood Estimation (MLE) to calculate the number of available tags on the internal nodes of the left branch of the tree.

The protocol uses BT until the first tag is identified. This process covers the full left branch of the tree. Then, the protocol goes step by step back on the different internal nodes and calculates the number of available tags on the right branch of those nodes using MLE. FTTP estimates that the number of tags available on the right branch should be equal to the number of tags on the left one which is already known. This is used to modify the internal counters of the tags

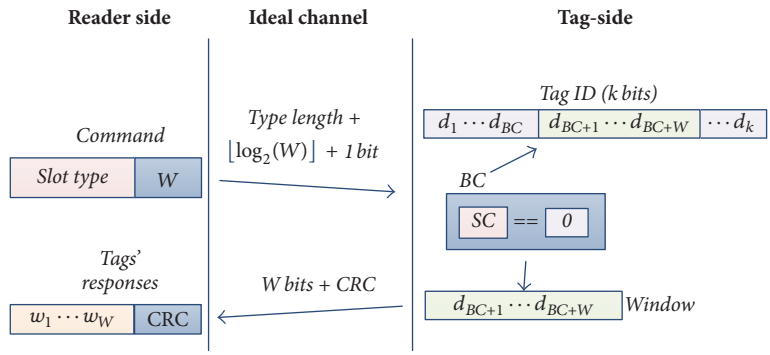


Figure 3: Example of a communication slot between the reader and a tag.

and separate their responses. FTTP then uses BT whenever a collision is produced during the identification.

2.2.3. Optimal Binary Tracking Tree Protocol. A BT based protocol with a bit estimator and bit tracking technology, referred to as the Optimal Binary Tracking Tree (OBTT) protocol, is given in [8]. It first employs a bit estimation algorithm to estimate the existing number of tags. Tags choose to swap one of the bits in a 1-bit string. This string is split into $x = 1/k - 1$ segments of the tag ID length k and each portion is transmitted separately to the reader. The reader uses Chebyshev's inequality [16] to calculate the estimated number of bits $\widehat{n}$ with the number of selected and nonselected bits received in the tags' responses. Once $\widehat{n}$ is obtained, the optimal number of queries m to initialize a query stack is calculated using $m = 0.595824\ \widehat{n}$. Afterwards, m queries are generated and pushed onto the stack. The rest of the process is solved using BT. Although the slot efficiency obtained by OBTT is very high, the preprocessing increases the energy consumption of the protocol, especially when $x > 1$.

2.2.4. Query Window Tree (QwT). A protocol named Query window Tree (QwT) protocol which uses a methodology to manage the number of bits transmitted by the tags is presented in [14]. The "window" is a bit string of size W, where $0 < W < k$, responded to by the tags instead of their full ID. The QwT is a modification of the query tree (QT) protocol [15] which includes the window methodology in tags' responses. As in QT, the reader transmits a command filled with a bit string, called query of length L, and the value of W calculated at the reader side. Tags compare the query with their initial part of their ID and those matching the reader's query respond exclusively to the following number of bits specified by W from the ID.

Three possible slot statuses can happen at the reader side, idle, collision, and success, as well as in QT. However, QwT introduces a new type of slot called go-on, when the reader receives similar windows of one or more tags simultaneously and obeys the following condition $L + W < k$. In this case, the reader has not received the full ID of the tag; therefore it cannot consider the tag identified.

The window alleviates tags from transmitting large number of bits upon a collision. However, low W values can cause the number of go-on slots to increase. Heuristic function (2) is proposed in [14] which provide an updated W value in case of a go-on slot to decrease its number. β parameter is used to tune the heuristic function.

$$W = \frac{k}{(k-\beta)^2} \times L^2. \tag{2}$$

Using the window, tags transmit fewer bits than that of the QT to be identified. However, the number of reader bits required by QwT is larger than that of QT since it uses a lot of longer queries than the latter.

3. The Proposed BwT Protocol

Here the BwT protocol is presented. This protocol implements the window methodology into BT. BwT tags use two counters: a slot counter SC and a bit counter BC; and the reader, on the other hand, uses another counter rBC. All tags update their SC on every slot and transmit when $SC = 0$. The BC indicates the first bit of its ID to transmit and is updated adding the W value only when the tag is in the transmitting state ($SC = 0$) and a notification of a go-on slot is received. Therefore, whenever a tag has its counter $SC = 0$, it transmits W bits starting from BC bit (see Figure 3) and a cyclic redundancy check code (CRC), similar to that demanded on the EPC C1G2 standard [6], to differentiate their responses at the reader.

The reader of BwT follows the same procedure on every slot. First, it receives the window transmitted by the tags and checks the type of slot it has received. According to the type of slot, the reader updates its counter rBC to monitor the length of the already acquired ID of the transmitting tag or tags and calculates the new W value to be transmitted. Then, it transmits a new command containing the status of the last slot received and the newly calculated W.

As mentioned before, slots can be idle, success, collision, and go-on. Pseudocode for the BwT reader and tags is shown in Algorithms 1 and 2 to perform an identification cycle. This pseudocode is executed during a specified time until all the tags in the interrogation zone are identified. In the beginning, the reader is initialized with $rBC = 0$ and $W = 1$. The new command is assembled with the type of the last slot and the value of W and is broadcast to the tags.

```
(1) Command = ε
(2) W = 1
(3) rBC = 0
(4) tagID = []
(5) k = ID.length
(6) while(unidentified tags) do
(7)     broadcast(Command,W)
(8)     [winID,crcOK] = receiveResponses
(9)     if isempty(winMatch) then Command = Idle
(10)    else crcOK==0 then
(11)      Command = Collision; W = 1
(12)      LIFOpush(tagID)
(13)    else
(14)      tagID = tagID+winID
(15)      rBC = tagID.length
(16)      if rBC + W < k then
(17)        Command = Go-On
(18)        W = f(rBC)
(19)      else
(20)        Command = Success
(21)        W = 1; LIFOpop(tagID)
```

Algorithm 1: Pseudocode of BwT. Reader procedure.

```
(22) sleep = false
(23) SC = 0; BC = 0
(24) nW = 1; oW = 1
(25) while (not sleep) do
(26)   receive(Command,nW)
(27)   switchCommand
(28)   case Idle:
(29)     SC=SC−1
(30)   case Collision:
(31)     ifSC==0 then SC=rand()%2
(32)     else SC=SC+1
(33)   case Go-On:
(34)     ifSC==0 then BC = BC + oW
(35)   case Success:
(36)     ifSC==0 then sleep=true
(37)     else SC=SC−1
(38)   ifSC==0 then
(39)     backscatter(ID[BC:BC+nW]); oW = nW
```

Algorithm 2: Pseudocode of BwT. Tag procedure.

After a certain period of time or after receiving a response, the reader identifies the type of slot according to the CRC consistency and takes the following actions depending on the slot identified:

(i) Idle slot (Algorithm 1 line (9)): when no response is received, the reader broadcasts a new command "Idle" with an invariant W.

(ii) Collision slot (Algorithm 1 lines (10)–(12)) occurs when the reader decodes the received windows and these are not CRC consistent. The reader needs to remember the already acquired ID since it may belong to different tags with a common partial ID. Therefore, the accumulated tag ID received at that point is stored into a Last Input First Output (LIFO) stack. Then the reader indicates the new command "Collision" with W set to 1 and broadcasts it to the tags.

(iii) Go-on slot (Algorithm 1 lines (16)–(18)) is when the CRC validates the received window and the condition $rBC + W < k$ is met. The reader updates the partial ID received and its length with the received window (Algorithms 1 lines (14)-(15)). Then, W is updated using the exponential heuristic function shown in (3). How W is adjusted is given in Section 3.1.

$$f(rBC) = k\left(1 - e^{-\beta \times rBC}\right). \tag{3}$$

This expression is a practiced deduction to balance the number of tag transmitting bits and go-on slots. It allows the reader to choose small W when the probability of collision is prone to increase, providing a small colliding tag bit rate; while it offers larger W when rBC increases (and, thus, the probability of collision decreases), contributing to decrease of the number of go-on slots. In addition, this W value is always delimited by the expression

$$W = \begin{cases} f(rBC), & W \leq k - rBC \\ k - rBC, & W > k - rBC. \end{cases} \tag{4}$$

Lastly, a new command "go-on" with the calculated W is broadcast to the tags.

(iv) Success slot (Algorithm 1 lines (20)-(21)) is when the CRC validates the received window and the tag ID is uniquely defined: $rBC+W = k$. Then, the ID is stored in a database and a new partially received ID is popped from the LIFO stack to continue the identification of the rest of the tags.

The tags' operation in Algorithm 2 line (25) starts receiving the reader's command and acts differently if they have transmitted in the previous slot ($SC = 0$).

(v) If tags remained silent in the previous slot, they update their SC counter adding in case of collision (Algorithm 2 line (32)) or subtracting for idle or success (Algorithm 2 lines (29), (37)).

(vi) If a tag transmitted in the previous slot ($SC = 0$), it performs differently. Upon a collision, the tag chooses a new SC randomly between 0 and 1; in case of success it goes to "sleep" state until the next interrogation cycle; and in case of a go-on command, the tag updates its BC counter with the previous W transmitted (oW), transmits by backscattering the immediately received W bits (nW) from the bit indicated by BC, and attaches the calculated CRC in the response.

An example of an identification of three tags using BwT is shown in Table 1. This example shows how the counters are updated upon the different types of slots.

3.1. Tuning β in BwT. As previously explained, the reception of a CRC consistent tag response does not necessarily mean the identification of a tag. The use of the window can cause the tag response to be only part of the ID; that is, $rBC + W < k$.

There is a need of dynamically recalculating W in order to decrease the number of go-on slots while keeping the number of tag transmitting bits low. The higher the value of W, the

Table 1: Example of an identification cycle of BwT.

Slot	Reader				Tag 1-0001011		Tag 2-1001010		Tag 3-1110001		Reader interpretation
	rBC [tagID]	LIFO	Command	*W*	*SC*	*BC*	*SC*	*BC*	*SC*	*BC*	
(1)	0 [ ]	{ }	ε	1	0	0	0	0	0	0	X
(2)	0 [ ]	{ }	Collision	1	1	0	0	0	0	0	1
(3)	1 [1]	{ }	Go-on	6	1	0	0	1	0	1	X
(4)	1 [1]	{1}	Collision	1	2	0	1	1	0	1	1
(5)	2 [11]	{1}	Go-on	5	2	0	1	1	0	2	10001
(6)	1 [1]	{ }	Success	1	1	0	0	1	Sleep	—	0
(7)	2 [10]	{ }	Go-on	5	1	0	0	2			1010
(8)	0 [ ]	{ }	Success	1	0	0	Sleep	—			0
(9)	1 [0]	{ }	Go-on	6	0	1					1011
(10)	0 [ ]	{ }	Success	1	Sleep	—					—

lower the number of go-on slots and the higher the number of tag transmitting bits. A desirable behavior can be found using exponential equation (3) to search for a balance between these parameters. The parameter β is tuned in order to seek for that balance.

The simulation results of the energy consumed by the RFID reader to identify 1 tag, the throughput of the system, and the number of slots and reader bits needed per tag for different values of β are shown in Figure 4 under a set of 1000 tags (more details about the simulation parameters are also given in Section 4). The results show that for β = [0,13–0,27] the energy consumed by the reader, and the number of slots, and reader bits per tag are at their lowest values and the throughput shows the highest values achieving a compensated balance between go-on slots and tag transmitting bits. Therefore, a value of β = 0,2 is chosen for the comparison with the state-of-the-art protocols.

4. Simulation Results

This section presents the simulation results of the proposed protocol using Matlab R2016b with an evaluation of the outcomes. A comparison between the proposed BwT protocol and the presented protocols in Section 2.2, BT [17], FTTP [7], OBTT [8], and QwT [14], is presented here.

A scenario with one reader and a varying set of tags from 100 to 1000 tags is proposed. These tags are uniformly distributed and k is assumed as 128 bits since it is the most common ID length that is currently used in the standard EPC C1 G2 (96 bits of Electronic Product Code + 16 bits of Protocol Control + 16 bits of CRC) [6]. The tag IDs are uniformly distributed and dynamically generated with varying random seed values for every simulation iteration. The simulated responses have been averaged over 100 iterations for accuracy in the results. Table 2 shows the parameters used in the simulations. Tari, the time interval for a data 0 transmission, is set to the standard's minimum of 6.25 μs for the highest data rate (same for reader and tag), conditioning, RTCal, TRCal, T1, T2, and T3 in accordance with the EPC standard [6]. P_{tx} and P_{rx} were obtained from [15].

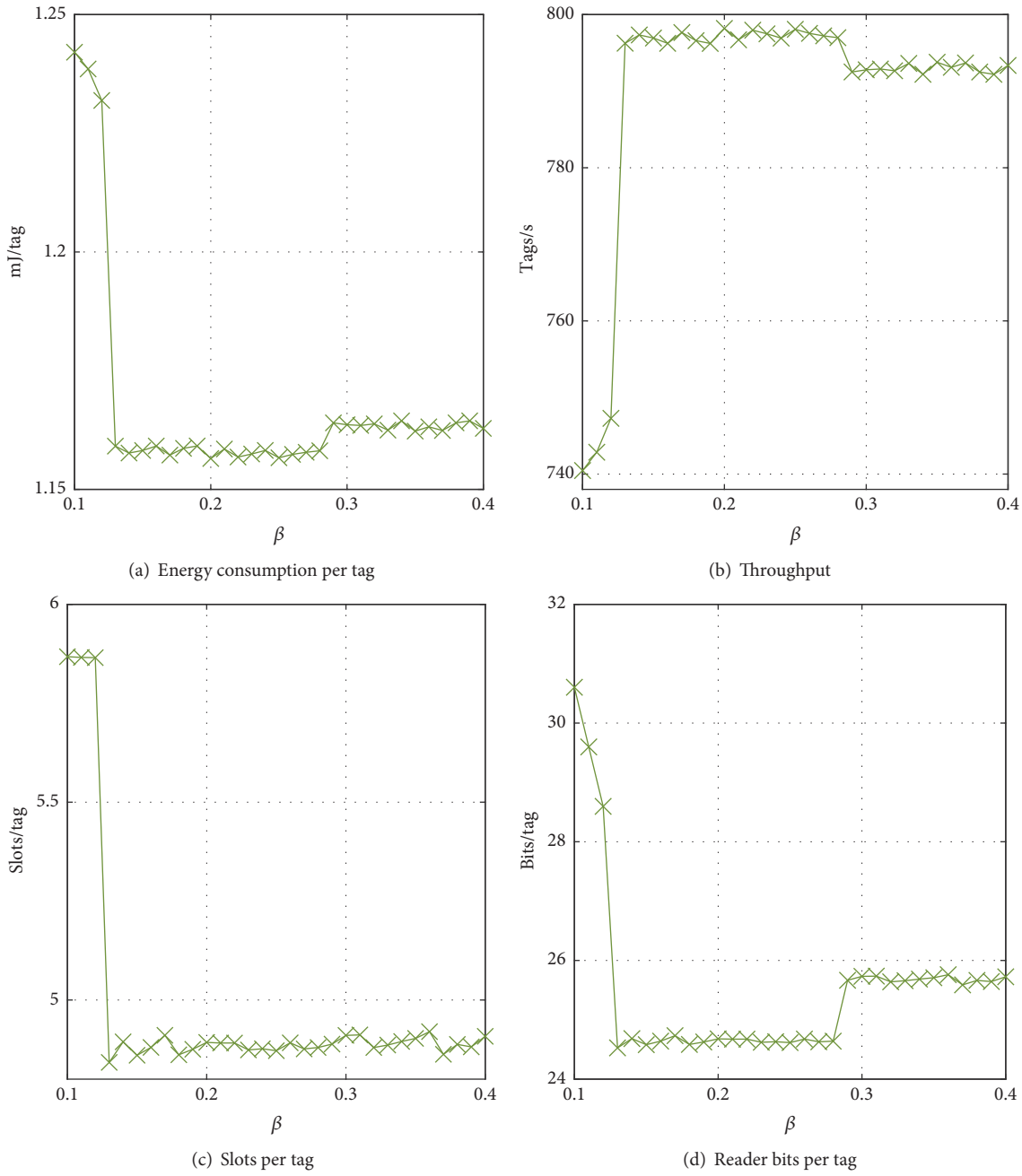
Presented in Figure 1 is the link timing of the four typical types of slots to perform identification time calculations and (1) for energy calculations. The duration of each slot can be different, and bits 0 and 1 have been considered as 1 Tari for easiness in calculations. This, in fact, has been applied to all the protocols, ensuring fairness in the comparison.

Table 2: Parameters used in simulations.

Parameter	Value
Tari	6.25 μs
data rate	160 kbps
RTCal	18.75 μs
TRCal	24.38 μs
$T1$	18.86 μs
$T2$	8.13 μs
$T3$	37.5 μs
P_{tx}	825 mW
P_{rx}	125 mW

The length of the reader commands is set to 3 bits for all the compared protocols, enough to encode all the needed commands. BwT, in addition, attaches W represented with $\log_2 W + 1$ bits; and QwT uses the length of the query on every slot plus the corresponding W bits. Tag responses are k bits long, except for BwT and QwT which use W bits and a CRC of 5 bits and OBTT which transmits the following ID bits to the received query after the estimation phase.

Figure 5 shows the throughput and the total number of bits used by the RFID system in the comparison of the simulated protocols. The calculation of the throughput is based on the total number of bits transmitted by the reader and the tags. BwT shows the highest throughput, slightly over OBTT. The use of the window increases the number of slots needed to identify the set of tags due to the generation of go-on slots. OBTT splits the initial set of tags in smaller subsets and therefore decreases the number of collisions and total slots. BwT, however, reduces the length of tag responses causing a BwT collision to spend less time than that of an OBTT. This is directly reflected on the throughput, meaning that the time BwT spends with go-on slots plus the time saved in collisions with low W values is less than the time OBTT spends on collisions.



(a) Energy consumption per tag

(b) Throughput

(c) Slots per tag

(d) Reader bits per tag

FIGURE 4: Selected $\beta = 0{,}2$ to obtain a high throughput, and a reduced energy consumption, low number of slots, and reader bits per tag in an interrogation cycle.

The window, therefore, contributes to reducing the number of tag transmitting bits, which is shown on Figure 5(b) showing BwT as the least bit consuming protocol. Although QwT also uses the window, the excessive number of bits demanded by the reader transmitting queries causes a higher increase in the total number of bits than that of the BwT, decreasing also the throughput of the system. OBTT presents good results also in both metrics thanks to its estimation phase at the beginning of the identification and the use of Manchester coding in the interrogation of the tags. This codification helps the reader to track collisions bit by bit, which in the end affects the total number of bits transmitted by reducing them. FTTP also presents an estimation phase in the beginning; however, it does not use Manchester coding and cannot reach the throughput of OBTT.

Simulations results per tag are shown in Figure 6. The energy consumed by the reader to identify 1 tag is shown in Figure 6(a). The energy consumed has been calculated using (1). In this comparison, BwT outperforms the other compared protocols for all the different sets of tags. The

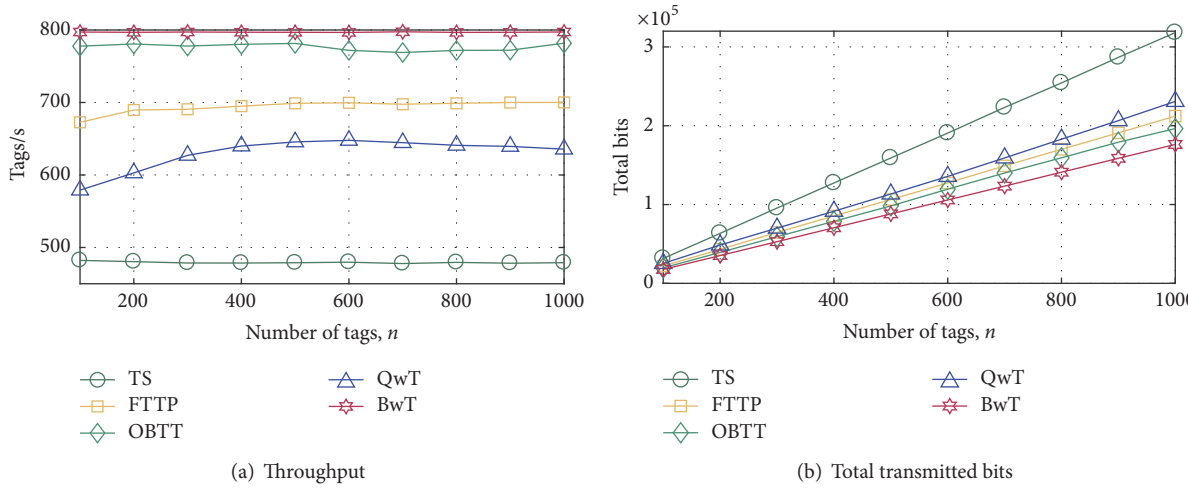


FIGURE 5: Simulation results of the throughput (a) and the total number of bits transmitted in the identification of several sets of tags (b).

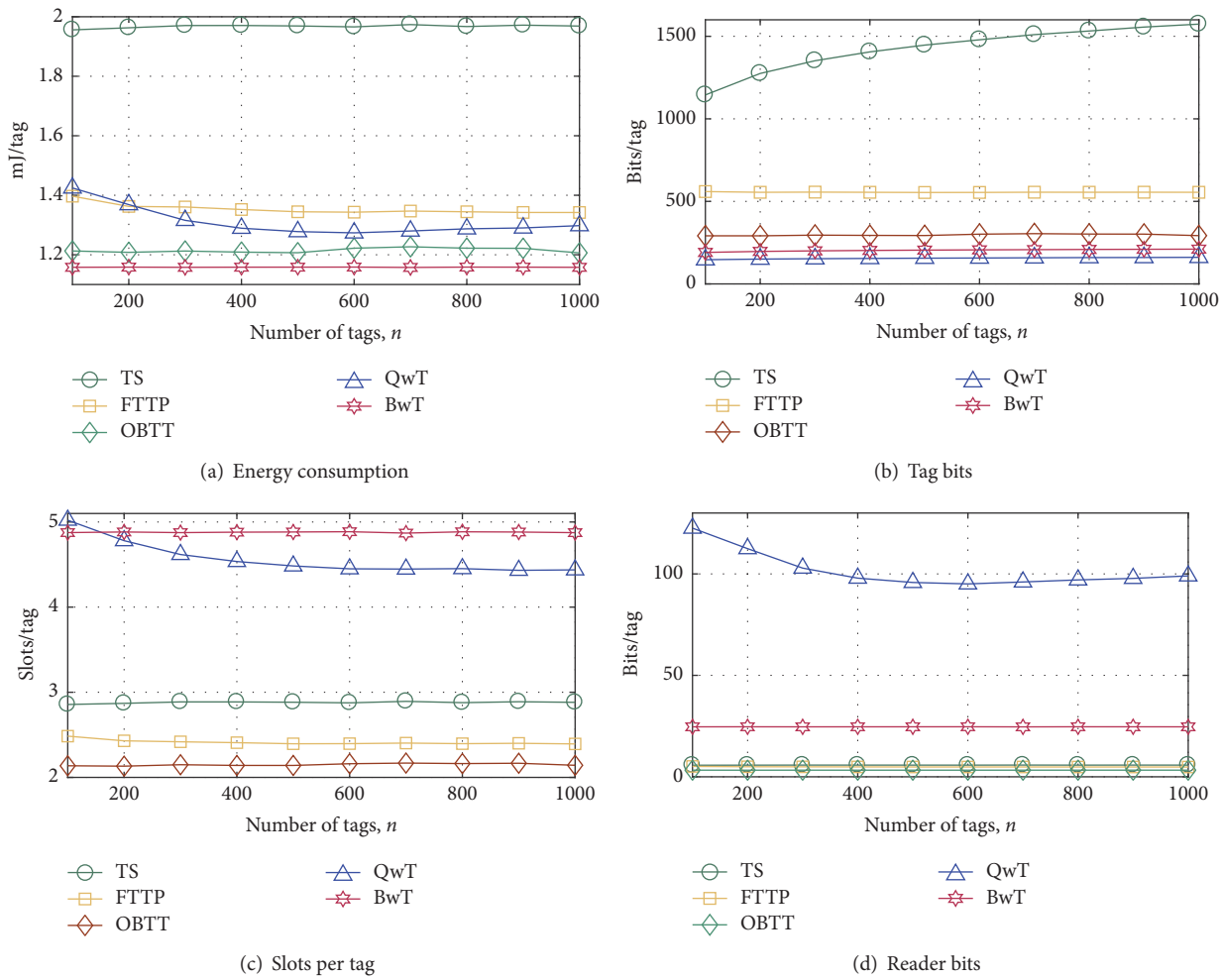


FIGURE 6: Simulation results of the energy consumption (a), the tag transmitting bits (b), the number of slots (c), and the reader transmitting bits (d) needed to identify 1 tag.

energy consumed to identify 1 tag is not affected by the total number of tags existing in the range of the antenna in BwT. OBTT and FTTP, however, show slight increases in dense and low populated tag environments, respectively. QwT shows a high increase when there are a few tags on the range of the reader's antenna and BT shows the worst results of the comparison.

As the bit window technique modifies the number of tag transmitting bits per slot, Figure 6(b) shows the improvement caused by the window in QwT and BwT as the least tag bit transmitting protocols, quite the opposite of BT. Although BwT and QwT tags need to use CRCs, they transmit the lowest number of bits.

The decrease in tag transmitting bits shown by the windowed protocols BwT and QwT is achieved at the cost of an increase in go-on slots. Figure 6(c) presents both protocols as the most slot consuming protocols to identify a tag. FTTP and OBTT, using the estimation phase, provide the best performance in terms of slots, where the bit tracking protocol OBTT stands out with only 2 slots to identify a tag. Notice also that despite the need of both windowed protocols of the highest number of slots, BwT saves more than 50% of the reader bits transmitted compared with that of QwT. QwT tags demand that the reader transmit long queries, which increases the number of reader bits.

Summing up, BwT reduces the number of tag transmitting bits thanks to the use of the window and avoids transmitting queries. This fact results in a deep reduction of the total transmitted bits reducing the energy consumed by the reader and increasing the throughput of the RFID system. These results show evidence of BwT being a good candidate which seeks for a high throughput under low energy consumption.

5. Conclusions

An anticollision protocol, called BwT, has been presented in this paper. BwT applies the window procedure to the BT protocol including an additional counter in the tags in order to manage the number of ID bits they transmit and the bits they have already transmitted. BwT has been compared to several state-of-the-art anticollision protocols outperforming them in terms of throughput and decreasing the energy consumed by the reader to identify 1 tag. Therefore, simulations showed that BwT can be considered as a good RFID anticollision candidate in passive RFID systems.

Definition of Symbols and Variables

t_R: Time needed to transmit a reader command
t_T: Time needed to transmit a tag response
T_1: Time the tags need to generate a response
T_2: Time the reader needs to receive a response
T_3: Max. time a reader waits before considering the slot idle
P_{tx}: Reader transmission power
P_{rx}: Reader reception power
E: Energy consumed by reader
k: Length of a tag ID
L: Length of a query
W: Size of the window.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] K. Finkenzeller, "RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and near-Field Communication," *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and near-Field Communication*, 2010.

[2] R. Want, B. N. Schilit, and S. Jenson, "Enabling the internet of things," *The Computer Journal*, vol. 48, no. 1, pp. 28–35, 2015.

[3] D. K. Klair, K.-W. Chin, and R. Raad, "A survey and tutorial of RFID anti-collision protocols," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 3, pp. 400–421, 2010.

[4] H. Gao, Z. Yang, J. Bhimani et al., "AutoPath: Harnessing Parallel Execution Paths for Efficient Resource Allocation in Multi-Stage Big Data Frameworks," in *Proceedings of the 2017 26th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–9, July 2017.

[5] Z. Yang, J. Wang, D. Evans, and N. Mi, "AutoReplica: Automatic data replica manager in distributed caching and data processing systems," in *Proceedings of the 35th IEEE International Performance Computing and Communications Conference, IPCCC 2016*, December 2016.

[6] GS1, 2015, EPC™ Radio-Frequency Identity Protocols Generation-2 UHF RFID - Specification for RFID Air Interface - Protocol for Communications at 860 MHz–960 MHz.

[7] J. Choi, I. Lee, D.-Z. Du, and W. Lee, "FTTP: A fast tree traversal protocol for efficient tag identification in RFID networks," *IEEE Communications Letters*, vol. 14, no. 8, pp. 713–715, 2010.

[8] Y. C. Lai, L. Y. Hsiao, and B. S. Lin, "Optimal slot assignment for binary tracking tree protocol in RFID tag identification," *IEEE/ACM Transactions on Networking*, vol. 23, no. 1, pp. 255–268, 2015.

[9] V. Namboodiri and L. Gao, "Energy-aware tag anticollision protocols for RFID systems," *IEEE Transactions on Mobile Computing*, vol. 9, no. 1, pp. 44–59, 2010.

[10] X. Yan and X. Liu, "Evaluating the energy consumption of the RFID tag collision resolution protocols," *Telecommunication Systems*, vol. 52, no. 4, pp. 2561–2568, 2013.

[11] D. K. Klair, K.-W. Chin, and R. Raad, "An investigation into the energy efficiency of pure and Slotted Aloha based RFID anti-collision protocols," in *Proceedings of the 2007 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WOWMOM*, 4, 1 pages, June 2007.

[12] Z. Yang, M. Awasthi, M. Ghosh, and N. Mi, "A fresh perspective on total cost of ownership models for flash storage in datacenters," in *Proceedings of the 8th IEEE International Conference on Cloud Computing Technology and Science, CloudCom 2016*, pp. 245–252, December 2016.

[13] F. Hessar and S. Roy, "Energy based performance evaluation of passive EPC Gen 2 class 1 RFID systems," *IEEE Transactions on Communications*, vol. 61, no. 4, pp. 1337–1348, 2013.

[14] H. Landaluce, A. Perallos, E. Onieva, L. Arjona, and L. Bengtsson, "An Energy and Identification Time Decreasing Procedure for Memoryless RFID Tag Anticollision Protocols," *IEEE Transactions on Wireless Communications*, vol. 15, no. 6, pp. 4234–4247, 2016.

[15] C. Law, K. Lee, and K.-Y. Siu, "Efficient memoryless protocol for tag identification," in *Proceedings of the 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pp. 75–84, Boston, Mass, USA, August 2000.

[16] H. Vogt, "Efficient object identification with passive RFID tags," in *Pervasive Computing*, vol. 2414 of *Lecture Notes in Computer Science*, pp. 98–113, Springer, Berlin, Germany, 2002.

[17] D. R. Hush and C. Wood, "Analysis of tree algorithms for RFID arbitration," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '98)*, pp. 107–114, IEEE, Cambridge, Mass, USA, August 1998.

[18] D. Bertsekas and R. Gallager, *Data Networks*, Pearson, 2nd edition.