

Análisis de Mecanismos Anticolisión en Sistemas RFID Pasivos UHF

**M^a Victoria Bueno Delgado¹, Javier Vales Alonso¹, Javier González Castaño²,
Esteban Egea López¹, Joan García Haro¹**

¹ Área de Ingeniería Telemática. E.T.S. Ingeniería de Telecomunicación, Universidad Politécnica de Cartagena, Plaza del Hospital nº 1, 30202, Cartagena, Murcia. {mvictoria.bueno, javier.vales, esteban.egea, joang.haro}@upct.es. ² Departamento de Ingeniería Telemática. E.T.S. Ingeniería de Telecomunicación, Universidad de Vigo, Campus, 36310, Vigo, Pontevedra. javier@det.uvigo.es

Resumen

Los sistemas RFID pasivos UHF utilizan mecanismos anticolisión FSA/DFSA para disminuir el impacto de las colisiones que se producen cuando hay un gran número de tags en la zona de cobertura del lector. La tasa de identificación óptima de FSA/DFSA se obtiene cuando la longitud de cada ciclo de identificación (número de slots de contienda) coincide con el número de tags a identificar. Los sistemas RFID pasivos UHF se suelen instalar en entornos donde no se conoce el número de tags que compite en cada ciclo. Por tanto, el lector debe decidir la longitud de cada ciclo basándose en diversas estimaciones como la predicción del número de tags en cobertura, el número de slots con colisión, etc., adaptándose a las limitaciones que impone el estándar actual EPCglobal Class-1 Gen-2 sobre la longitud de ciclo (sólo potencias de 2). En este trabajo presentamos el análisis, simulación y comparación de los mecanismos anticolisión más destacados de la literatura científica.

Palabras clave: FSA/DFSA, tasa de identificación máxima, EPCglobal Class-1 Gen-2.

1. Introducción

En los sistemas RFID pasivos UHF la comunicación entre el lector (*reader*) y los *tags* se realiza mediante el acceso a un canal de comunicación compartido. Por ello, es necesario un mecanismo de acceso al medio (MAC) para minimizar el impacto de las colisiones que se producen por las posibles transmisiones simultáneas de los *tags* y disminuir así, el tiempo total de identificación. La simplicidad del hardware en los *tags* pasivos obliga a trasladar la complejidad del protocolo o mecanismo de anticolisión al lector (p.e., la sincronización).

La importancia del protocolo anticolisión en un sistema RFID pasivo UHF se demuestra en el siguiente ejemplo: en un almacén de una empresa industrial hay una cinta transportadora con un sistema RFID pasivo UHF en un punto intermedio de la cinta (Fig.1). Conforme la cinta se mueve, las cajas etiquetadas con *tags* van entrando a la zona de cobertura, se identifican y vuelcan los datos de los productos al lector. El lector está configurado de forma que, mientras haya colisiones, la cinta transportadora no avanza. Este mecanismo evita que un *tag* salga de la zona de cobertura sin haber enviado la información del producto, pero a costa de parar la cinta y ralentizar el sistema. Por tanto, la capacidad de avance de la cinta dependerá de las prestaciones del protocolo anticolisión implementado en el lector.

Los mecanismos anticolisión para sistemas RFID pasivos UHF, incluyendo los estándares actuales, se basan principalmente en dos variaciones del protocolo Aloha Ranurado (*Slotted Aloha*), que divide el medio compartido en ranuras temporales, denominadas *slots*: (i) Aloha

ranurado por trama FSA (*Frame Slotted Aloha*), una variación de Aloha ranurado donde los *slots* están confinados en tramas consecutivas llamadas ciclos y (ii) Aloha ranurado por trama dinámico DFSA (*Dynamic Frame Slotted Aloha*), una variación de FSA que permite variar la longitud de cada ciclo, es decir, el número de *slots* que conforman una trama [1].

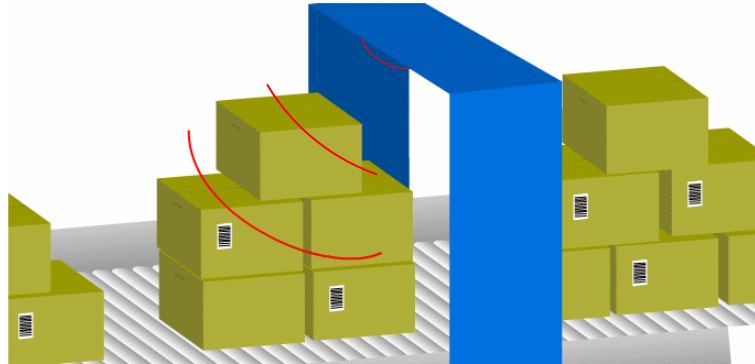


Figura 1. Sistema RFID pasivo instalado en cinta transportadora.

En un sistema RFID pasivo que implementa un protocolo basado en FSA/DFSA el lector inicia un ciclo de identificación anunciando la longitud (en número de *slots*) de ese ciclo. Todos los *tags* en cobertura reciben esa información y escogen aleatoriamente un *slot* de ese ciclo para transmitir su identificador. Cuando finaliza un ciclo, el lector inicia un nuevo ciclo manteniendo el mismo número de *slots* por ciclo (FSA), o modificando el número de *slots* de contienda para el siguiente ciclo (DFSA).

Las prestaciones del mecanismo anticolidión dependen de la relación existente entre el número de *tags* a identificar y la longitud del ciclo. Por ejemplo, si el número de *tags* en la zona de cobertura del lector es mucho mayor que el número de *slots* de contienda, el tiempo de identificación se incrementa considerablemente ya que se producen muchas colisiones y se necesitan muchos ciclos de identificación para que todos los *tags* envíen su información. Por otro lado, si hay pocos *tags* en la zona de cobertura y la longitud del ciclo es elevada, se suceden muchos *slots* vacíos, lo que también incrementa innecesariamente el tiempo de identificación. Con FSA/DFSA, si en un ciclo i van a competir t_i *tags*, el número óptimo n_i de *slots* en ese ciclo que maximiza el número de *tags* identificados es exactamente $n_i = t_i$. En esta situación se obtiene una *tasa de identificación* máxima en ese ciclo de $\tau_i = e^{-1} \approx 0.36$ [1]. Por tanto, optimizar el número de *slots* por ciclo implica resolver los siguientes problemas:

- a) Los lectores RFID pasivos UHF habitualmente se instalan en entornos donde no se tiene un conocimiento *a priori* del número de *tags* que entran en la zona de cobertura del lector para identificarse. Conforme llegan *tags*, éstos se identifican, utilizando el protocolo anticolidión definido en los estándares actuales. El lector, al no conocer el número de *tags* que compiten en cada ciclo de identificación no puede establecer la longitud de ciclo que permite obtener la *tasa de identificación máxima* en ese ciclo.
- b) El estándar actual para sistemas RFID pasivos UHF es EPCglobal Class-1 Gen-2 [2]. Éste limita el número de *slots* por ciclo a potencias de 2, $n_i = 2^{Q_i}$, con $Q_i \in [0, \dots, 15]$, es decir, longitudes de ciclo de valor $n_i = [2^1, 2^2, 2^3, \dots, 2^{15}]$. Esta limitación implica que, en contadas ocasiones, se cumplirá $t_i = n_i = 2^{Q_i}$, y por tanto, el número de *tags* que compite en el siguiente ciclo será un valor comprendido entre $2^{(Q_i-1)} < t_i < 2^{Q_i}$. La cuestión es que el lector debe establecer una n_i entre las dos opciones posibles.

En la literatura científica hay numerosos trabajos donde se proponen algoritmos anticolidión basados en FSA/DFSA para sistemas RFID pasivos UHF. Surge el interés de realizar una comparativa entre los algoritmos propuestos más relevantes, valorando la problemática indicada en los puntos (a) y (b) anteriores. En esta línea, presentamos el análisis, simulación y comparación de los algoritmos anticolidión basados en FSA/DFSA más destacados, diseñados para sistemas RFID pasivos UHF y compatibles con el estándar EPCglobal Class-1 Gen-2.

El resto del artículo está organizado en las siguientes secciones: la sección 2 describe el estándar EPCglobal Class-1 Gen-2. La sección 3 introduce los algoritmos anticolidión DFSA más relevantes objeto de estudio en este trabajo. La sección 4 presenta las simulaciones realizadas y los resultados obtenidos. Por último, en la sección 5 se extraen las conclusiones.

2 El estándar EPCglobal Class-1 Gen-2 (FSA)

EPCglobal Class-1 Gen-2 [2] es un estándar de la institución EPCglobal, centrada en el desarrollo de estándares industriales para EPC (*Electronic Product Code*) mediante identificación por radiofrecuencia. Este trabajo se centra en EPCglobal Class-1 Gen-2 para sistemas RFID pasivos en UHF (860MHz-930MHz). Incluye un conjunto de especificaciones hardware y software para los *tags* pasivos y el sistema lector (en el cual reside la complejidad del sistema). Desde su publicación en el año 2005, este protocolo ha sido ampliamente adoptado por los fabricantes de sistemas RFID, siendo hoy día el protocolo mayoritario.

En el estándar EPCglobal Class-1 Gen-2, un ciclo de identificación se inicia cuando el lector envía un paquete tipo *Query*, incluyendo en uno de sus campos cuatro bits, que indican el valor de $Q \in [0, \dots, 15]$, es decir, el tamaño del ciclo (2^Q slots). Los *tags* en cobertura que reciben el paquete generan un número aleatorio r dentro del intervalo $[0, 2^Q-1]$. El valor r representa el *slot* del ciclo actual en el que el *tag* transmitirá su identificador $ID=r$. En cada ciclo, el comienzo de un nuevo *slot* lo indica el lector por medio del paquete *QueryRep*, exceptuando el *slot* 0, el cual se inicia automáticamente tras el envío del paquete *Query*¹. Los *tags* que compiten para identificarse utilizan un contador interno ($counter=r$) para contabilizar los *slots* que les quedan hasta alcanzar el elegido y enviar su identificador. Para ello, cada vez que les llega un paquete *QueryRep*, decrementan el contador y si éste alcanza el valor 0, envían su identificador ID , que corresponde al valor obtenido r , es decir, el *slot* elegido en el ciclo. Después de que un *tag* transmita su identificador puede ocurrir:

- Si más de un *tag* elige el mismo *slot* para transmitir su ID se produce una colisión. El lector la detecta y reacciona enviando un nuevo paquete *QueryRep* (*slot* 0 de Fig.2). Al recibir el paquete, los *tags* que han transmitido su ID asumen que ha habido una colisión y actualizan su contador $counter=2^Q-1$, evitando que vuelvan a competir para identificarse en ese ciclo.
- Si el lector recibe un ID correcto, responde con un paquete *Ack*. Los *tags* en cobertura reciben el paquete, pero sólo el *tag* que envió su ID contesta enviando un paquete *Data*, p. ej. un paquete EPC. Si el lector recibe el paquete correctamente responde con un nuevo paquete *QueryRep* comenzando un nuevo *slot*. El *tag* identificado finaliza su proceso (*slot* 1 de Fig.2). Si el lector no recibe el paquete *Data* correctamente o en un tiempo establecido, envía un paquete *Nack*. Los *tags* en cobertura lo reciben pero sólo el *tag* que envió los datos reacciona estableciendo su contador a $counter=2^Q-1$

¹ Los paquetes *Query* y *QueryRep* permiten la sincronización entre el lector y los *tags*.

evitando así que vuelva a competir por identificarse en ese ciclo (*slot* 3 de Fig.2). Tras el paquete *Nack* el lector envía de nuevo un paquete *QueryRep*.

Al finalizar un ciclo, el lector envía un nuevo paquete *Query*. Los *tags* que en el ciclo anterior no se identificaron vuelven a competir en el nuevo ciclo eligiendo un nuevo *slot* de contienda.

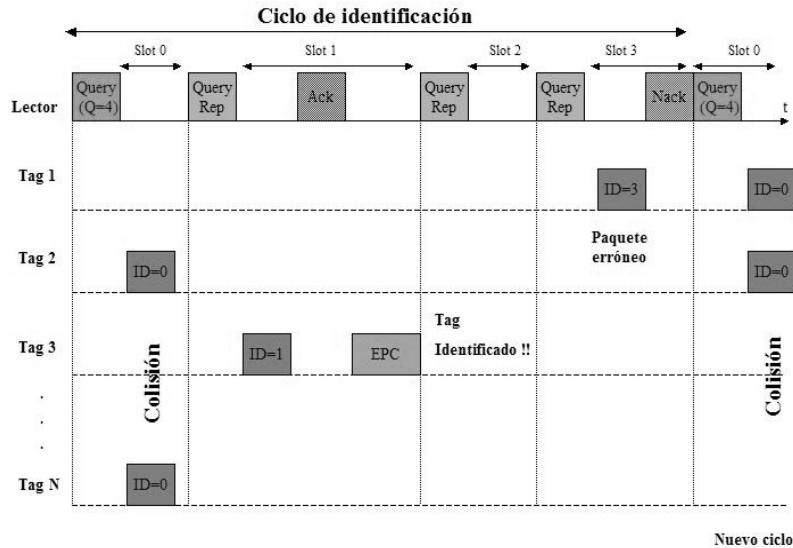


Figura 2. Mecanismo anticolisión EPCglobal Class-1 Gen-2

Este mecanismo anticolisión permite que el lector pueda controlar el número de *slots* por ciclo a través del valor Q , decidiendo si comenzar un nuevo ciclo en un instante determinado, enviando un nuevo paquete *Query* con el mismo o un nuevo valor de Q . La modificación del valor de Q puede afectar negativamente al tiempo de identificación de una población de *tags* en cobertura. Con valores de Q elevados y pocos *tags* en cobertura se suceden muchos *slots* vacíos por ciclo. Por otro lado, si Q se establece con un valor muy bajo y el número de *tags* en cobertura es muy alto, implica muchos *slots* con colisión por ciclo. Ambas situaciones se deben evitar ya que incrementan considerablemente el tiempo de identificación. EPCglobal Class-1 Gen-2 trabaja por defecto con el mismo valor de Q en todos los ciclos. Esta configuración se denomina configuración Q estática y los sistemas RFID pasivos del mercado que implementan EPCglobal Class-1 Gen-2 presentan dicha configuración [3][4].

3 Análisis de algoritmos anticolisión DFSA

Los algoritmos DFSA se pueden clasificar en función del mecanismo de adaptación de longitud de ciclo empleado, distinguiendo principalmente tres tipos: (i) algoritmos de corrección de trama (*Frame-Size-Correction*), (ii) algoritmos basados en heurísticos (*Heuristic algorithms*) y (iii) algoritmos de estimación bayesiana (*Bayesian Estimation algorithms*) o máxima verosimilitud (*Maximum Likelihood (ML) Estimation algorithms*). La implementación de estos últimos conlleva un elevado coste computacional. Por esta razón, en este trabajo sólo son objeto de estudio los algoritmos más relevantes clasificados en (i) y (ii).

3.1 Algoritmos de corrección de trama (*Frame-Size Correction*)

Los algoritmos de corrección de trama modifican la longitud del ciclo en función de diversos parámetros como, por ejemplo, el número de *slots* vacíos, con colisión, con identificador, etc.

Estos algoritmos no realizan una estimación del número de *tags* que compiten en cada ciclo de identificación, y por tanto, el lector no puede calcular la longitud de ciclo que permite obtener la *tasa de identificación máxima* definida en [1].

3.1.1 EPCglobal Class-1 Gen-2 *Q*-dinámica

EPCglobal Class-1 Gen-2 [2] presenta una alternativa a la configuración *Q*-estática para adaptar la longitud de los ciclos. Esta configuración, denominada *Q*-dinámica, permite modificar el valor de *Q*, *slot* a *slot*, siguiendo el mecanismo descrito en Fig. 3. Al finalizar un *slot* de un ciclo de identificación, se observa si éste está vacío, con colisión o con solo un *ID* de *tag*. De acuerdo al estado del *slot*, se incrementa, decrementa o mantiene el valor de *Q*. Para ello se utiliza la variable $C \in (0.1, 0.5)$. El estándar no define los valores exactos de *C* a utilizar en cada caso. Sólomente recomienda utilizar valores altos de *C* en situaciones donde *Q* tiene un valor bajo y viceversa.

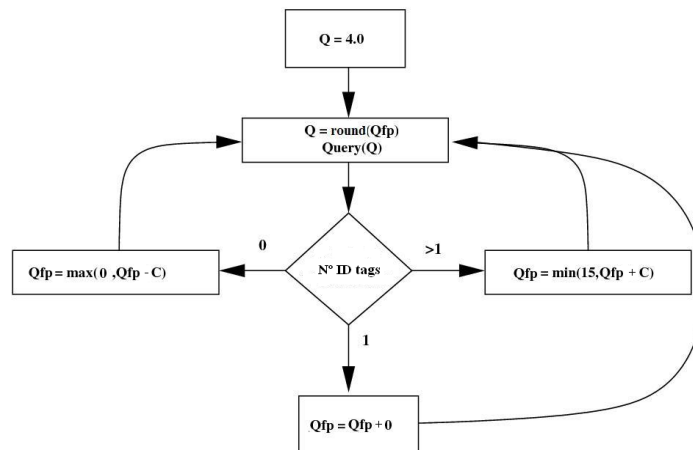


Figura 3. Algoritmo *Q*-dinámica en EPCglobal Class-1 Gen-2.

3.1.2 Algoritmo *Q* + -

En [5], se proponen varias modificaciones de *Q*-dinámica con el fin de obtener mejores resultados en tiempo medio de identificación. En primer lugar, como los paquetes *Query* son de mayor tamaño (22 bits) que los paquetes *QueryRep* (4 bits), se modifica el algoritmo para enviar paquetes *Query* sólomente cuando el valor de *Qfp* calculado al finalizar un *slot* sea distinto al valor del *Q* actual. Este mecanismo evita el envío innecesario de paquetes *Query* que no modifican la longitud de un ciclo. Por otro lado, se sustituye la variable *C* por las constantes C_i , para el caso de *slot* vacío y C_c para el caso de *slot* con colisión. La Tabla 1 muestra los valores de estas constantes, que difieren dependiendo de si se tiene o no un conocimiento *a priori* del número de *tags* que compite en un ciclo de identificación (t_i).

Tabla 1. Valores establecidos en constantes C_i y C_c

	t_i conocido	t_i no conocido
C_c	$-0.0491 \cdot \ln(t_i)$	$[0.1, 0.5]$
C_i	$(e-2) \cdot C_c$	$(e-2) \cdot C_c$

Como los lectores RFID pasivos UHF se suelen instalar en entornos donde no se conoce el número de *tags* a identificar, se asume que se utilizan los valores de C_i y C_c para el caso de t_i no conocido. En [5], no se indica qué valor es el que se debe establecer en C_c . Además,

realizan la comparativa del algoritmo propuesto con el algoritmo EPCglobal Class-1 Gen-2 *Q-estática* para distintos valores de Q y con el algoritmo propuesto en [6]. Sin embargo, los autores no contrastan los resultados obtenidos con el algoritmo *Q-dinámica* que es, a nuestro juicio, la comparativa de interés ya que el *algoritmo Q+-* es un algoritmo fruto de ciertas modificaciones para mejorar el estándar *Q-dinámica*.

3.1.3 Algoritmo *C* óptima

Como el estándar EPCglobal Class-1 Gen-2 no define cuales son los valores exactos del parámetro C , en [7] se calculan por simulación, obteniendo el valor óptimo de C en función de la longitud de ciclo para $Q \in [0, \dots, 15]$. El estándar define $C \in (0.1, 0.5)$. Sin embargo, en [7] los valores de C se asumen $C \in [0.1, 0.5]$. Al igual que en el algoritmo anterior, en [7] se compara el algoritmo propuesto con el protocolo EPCglobal Class-1 Gen-2 *Q-estática* y el propuesto en [8], en lugar de comparar los resultados obtenidos con *Q-dinámica*, que es como realmente se podrían observar las mejoras frente al estándar.

3.2 Algoritmos heurísticos (*Heuristic algorithms*)

A diferencia de los anteriores, los algoritmos basados en heurísticos calculan la longitud de cada nuevo ciclo *realizando una estimación previa del número de tags* que quedan por identificar al finalizar cada ciclo. Para ello se utiliza diversos mecanismos. Aunque presentan mejores resultados de tiempo medio de identificación que los algoritmos de corrección de trama, los algoritmos heurísticos realizan una estimación del número de *tags* por ciclo que se aleja del valor real de *tags* que compete en cada ciclo, lo que implica unos resultados lejanos a la *tasa de identificación máxima* definida en [1].

3.2.1 Algoritmos basados en estimador de Schoute.

En 1983, se propuso en [8] un algoritmo para estimar el número de estaciones móviles no identificadas (*backlog*) al finalizar un ciclo de identificación en FSA y la longitud del siguiente ciclo. Se asume que el número de estaciones que transmiten en cada *slot* sigue una distribución de Poisson. En la ecuación (1) se muestra el estimador propuesto en [8]. Bt es el número de estaciones móviles sin identificar y c el número de *slots* con colisión que se han sucedido en el último ciclo de identificación.

$$Bt = 2.39 \cdot c \quad (1)$$

El cálculo del número de *slots* para el siguiente ciclo se presenta en la ecuación (2), donde n es el número de *slots* para el siguiente ciclo y $\text{round}^+(Bt)$ indica el valor entero superior del número de estaciones sin identificar.

$$n = \text{round}^+(Bt) \quad (2)$$

Sorprendentemente, en algunos trabajos de la literatura científica actual como p.ej. en [7], [9] y [10] se deduce el mismo estimador que en [8] presentándolo como un mecanismo novedoso. De estos trabajos, sólo en [7] se presenta una tabla con una relación entre el número de *tags* por identificar, y la longitud de ciclo asociado en el formato del estándar ($n_i = 2^Q$). Sin embargo, esta tabla es incompleta, ya que propone valores de $Q \in [5, \dots, 10]$, mientras que el estándar establece valores de $Q \in [0, \dots, 15]$.

3.2.2 Algoritmo Lower Bound

Dado que una colisión implica que al menos dos *tags* han transmitido su identificador en el mismo *slot*, Vogt propone en [6] un estimador del número de *tags* observando únicamente número de *slots* con colisión al finalizar un ciclo:

$$t_{i+1} = 2 \cdot c_i \quad (3)$$

Siendo c_i el número de *slots* con colisión en el ciclo i , y considerando que la longitud del siguiente ciclo será igual al número de *tags* estimado, sea o no una potencia de 2 entre 0 y 15.

3.2.4 Algoritmo C-ratio

En [9] también se propone el algoritmo *C-ratio*, que estima el número de *tags* a identificar utilizando la ecuación (4). El problema de este algoritmo es que si $c_i = n_i$, entonces, $t_{i+1} = n_i + 1$, lo que resulta incongruente. De nuevo, la longitud de ciclo se asume igual al número de *tags* estimado, sea o no una potencia de 2 entre 0 y 15.

$$\frac{c_i}{n_i} = 1 - \left(1 - \frac{1}{n_i}\right)^{t_{i+1}} \left(1 + \frac{t_{i+1}}{n_i - 1}\right) \quad (4)$$

3.2.6 Algoritmo Chen

El algoritmo propuesto en [11] estima el número de *tags* por identificar utilizando la información obtenida sobre el número de *slots* con identificador y número de *slots* vacíos al finalizar un ciclo i . Para ello, los autores proponen la ecuación (7).

$$t_{i+1} = (n_i - 1) \frac{s_i}{e_i} \quad (7)$$

Si al finalizar un ciclo, $e_i = 0$ entonces, el número de *tags* a identificar se estima como en la ecuación (3). Sin embargo, este estimador es inestable. Cuando aparece una población elevada de *tags* en cobertura con el lector y la longitud del ciclo es muy pequeña, el sistema puede no ser capaz de incrementar la longitud del ciclo para adaptarse a la situación. Esto es debido a que puede que el ciclo finalice y sólo se hayan sucedido *slots* vacíos y con colisión, con lo cual $s_i = 0$. Esta situación provoca que el algoritmo estime $t_{i+1} = 0$, decidiendo una longitud para el siguiente ciclo muy pequeña, lo que puede seguir provocando un $s_i = 0$. De nuevo, en [11] no se indica que la longitud del ciclo deba ser una potencia de 2.

4 Resultados de simulación

Los protocolos analizados en la sección anterior y resumidos en la Tabla 2, se han implementado en un simulador de sistemas RFID pasivos UHF desarrollado con la herramienta Matlab [12]. Con el fin de obtener resultados realistas, las simulaciones se han realizado teniendo en cuenta los parámetros hardware de los dispositivos del kit de desarrollo Alien 8800 [3] que se muestran en la Tabla 3. El simulador recolecta estadísticos de número medio de ciclos de identificación, número medio de *slots* de identificación, número medio de *slots* por *tag* (tasa de identificación), tiempo medio de identificación, etc.

Evaluar los algoritmos anticolidión DFSA únicamente en número medio de ciclos puede llevarnos a conclusiones erróneas ya que, en estos algoritmos, el número de *slots* en cada ciclo es variable. Por ello, en este trabajo se presentan resultados de simulación en términos de número de medio de *slots* para la identificación, y número medio de *slots* utilizados por *tag* para la identificación (tasa de identificación).

Para comparar la tasa de identificación de los mecanismos estudiados respecto de la *tasa de identificación máxima* ha sido necesario implementar un *algoritmo anticolidión óptimo* que permita obtener dicha tasa. Para ello, se ha asumido que el lector tiene un conocimiento real del número de *tags* que compite en cada ciclo y que establece el valor de longitud de ciclo 2^Q que maximiza el número de identificaciones por ciclo. Este algoritmo se ha implementado utilizando los valores de Q óptima obtenidos en el trabajo que realizamos en [13].

Tabla 2. Mecanismos DFSA evaluados compatibles con EPCglobal Class-1 Gen2

Mecanismo anticolidión	Estimación del número de <i>tags</i> por ciclo	Cálculo de longitud de ciclo	Ajuste de Q
Q -dinámica [2]	No	<i>Slot a slot</i> , parámetro C estándar	SI
Q + [5]	No	<i>Slot a slot</i> , parámetros C_i, C_c	SI
C óptima [7]	No	<i>Slot a slot</i> , parámetro C óptimo	SI
Schoute [7-10]	$t_{i+1} = 2.39 \cdot c_i$	$n_{i+1} = t_{i+1}$	NO
Lower Bound[6]	$t_{i+1} = 2 \cdot c_i$	$n_{i+1} = t_{i+1}$	NO
C-ratio [9]	$\frac{c_i}{n_i} = 1 - \left(1 - \frac{1}{n_i}\right)^{t_{i+1}} \left(1 + \frac{t_{i+1}}{n_i - 1}\right)$	$n_{i+1} = t_{i+1}$	NO
Chen [11]	$t_{i+1} = (n_i - 1) \frac{S_i}{e_i}$	$n_{i+1} = t_{i+1}$	NO

Para simular los mecanismos que no ajustan la longitud de ciclo a las limitaciones que impone el estándar, se ha supuesto que si $2^{(Q_i-1)} < t_i < 2^{Q_i}$, $n_i = 2^{Q_i}$. Los mecanismos se han evaluado para distintas poblaciones de *tags* en cobertura. Para todos los mecanismos evaluados se ha tenido en cuenta que la Q inicial es $Q=4$.

Tabla 3. Parámetros físicos del kit Alien 8800 establecidos en las simulaciones realizadas

Parámetros	Tags pasivos	Lector/antena
Frecuencia de trabajo	UHF 868-928MHz	UHF 868-928MHz
Rango de cobertura	10cm-3m	Hasta 10m
Memoria disponible	96-256 bits	--
Modulación	ASK	ASK/PSK
Tasa Tx/Rx	40 Kbps	80Kbps
Potencia máxima de la antena	--	10 W
Ganancia antenas	--	6dBi
Máxima potencia RF del lector	--	4W

4.1 Comparativa algoritmos de corrección de trama

En [5] y [7] se proponen mecanismos anticolidión resultado de la modificación del estándar Q -dinámica [2]. Sin embargo, estos trabajos sólo se comparan con el estándar Q -estática. Por ello, en primer lugar realizamos una comparativa de los algoritmos de corrección de trama [2], [5] y [7] para comprobar si las variaciones propuestas en [5] y [7] suponen una mejora respecto al estándar. En la figura 4(a) se muestra el número medio de *slots* para la identificación, para distintas poblaciones de *tags*. Se comprueba como el número medio de *slots* utilizando el *algoritmo* Q + [5] es mucho menor que el de los algoritmos algoritmo C -

optima [7] y *Q-dinámico* [2]. En la figura 6(a) se muestra la tasa de identificación de los algoritmos estudiados. En la siguiente sección se demuestra como estos últimos resultados se alejan mucho de la tasa de identificación máxima.

4.2 Comparativa entre algoritmos basados en heurísticos y algoritmo óptimo

Esta sección evalúa los algoritmos basados en heurísticos y los compara con el algoritmo óptimo. En la figura 4(b) se muestran los resultados de simulación al evaluar el número medio de *slots* para la identificación. El *algoritmo óptimo* es el que presenta los mejores resultados, seguido del *algoritmo C-ratio*. Aunque a primera vista parece que los resultados de todos los algoritmos heurísticos estudiados son cercanos a los resultados del *algoritmo óptimo*, observando atentamente la figura 5 se muestra como, p.ej. para 600 *tags* en cobertura, la diferencia entre el *algoritmo óptimo* y el más cercano (*C-ratio*) supera los 100 *slots*. Esto significa que, si la duración de un *slot* es de 2.5 ms [13], los algoritmos evaluados están a más de 2.5 s del óptimo. En la figura 5 también se observa cómo, al aumentar el número de *tags* en cobertura, la distancia entre los algoritmos evaluados y el *algoritmo óptimo* aumenta, lo que implica una distancia temporal aún mayor.

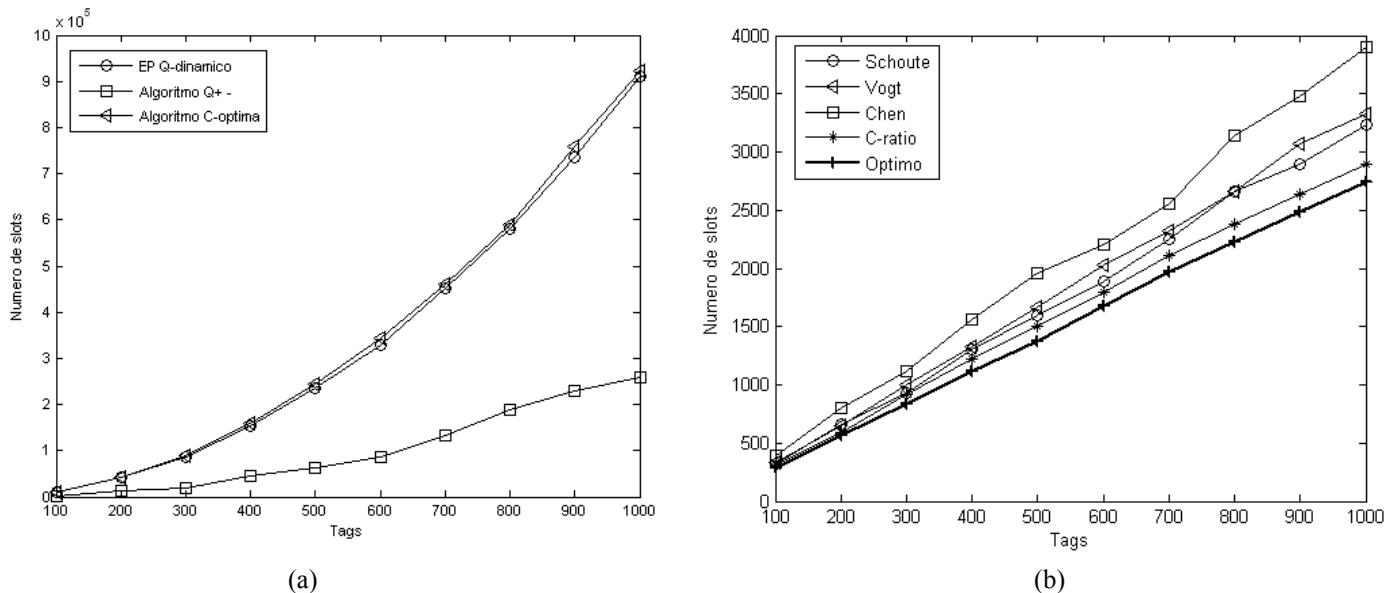


Figura 4 (a) N° medio de *slots* vs población de *tags* en cobertura en mecanismos de corrección de trama, (b) N° medio de *slots* vs población de *tags* en cobertura en mecanismos heurísticos.

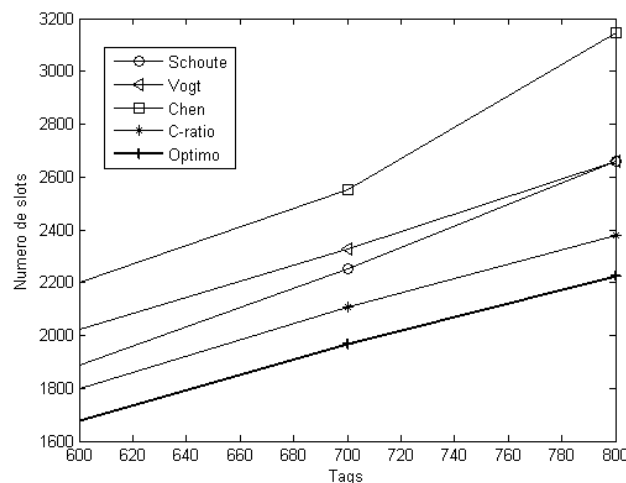


Figura 5. Zoom de la figura 5(b).

En la figura 6(b) se muestra la tasa de identificación de los algoritmos heurísticos estudiados. Los valores obtenidos se alejan bastante de la *tasa de identificación máxima*. Si comparamos el *algoritmo óptimo* de la figura 6(b) respecto a los resultados de la figura 6(a) se comprueba como los algoritmos de corrección de trama se adaptan mucho más lentamente, por lo que presentan una *tasa de identificación* muy degradada respecto a los algoritmos heurísticos. P.ej. con una población de 500 *tags*, el mejor algoritmo de corrección de trama (*C-óptima*) presenta una tasa de identificación por *tag* de 100 *slots*, frente a los 2 *slots* por *tag* con el *algoritmo óptimo*. Esto significa que, si la duración de un *slot* es de 2.5 ms [13], y 500 *tags* en cobertura con el lector, con el *algoritmo óptimo* cada uno de los *tags* tardaría una media de 5 ms en identificarse mientras que con el algoritmo *C-óptima* tardarían una media de 2.5 s.

Conociendo la duración temporal de los *slots*, de los resultados anteriores se pueden deducir los tiempos de identificación de cada uno de los algoritmos. Las curvas obtenidas son similares a las presentadas en las figuras 4(a) y (b).

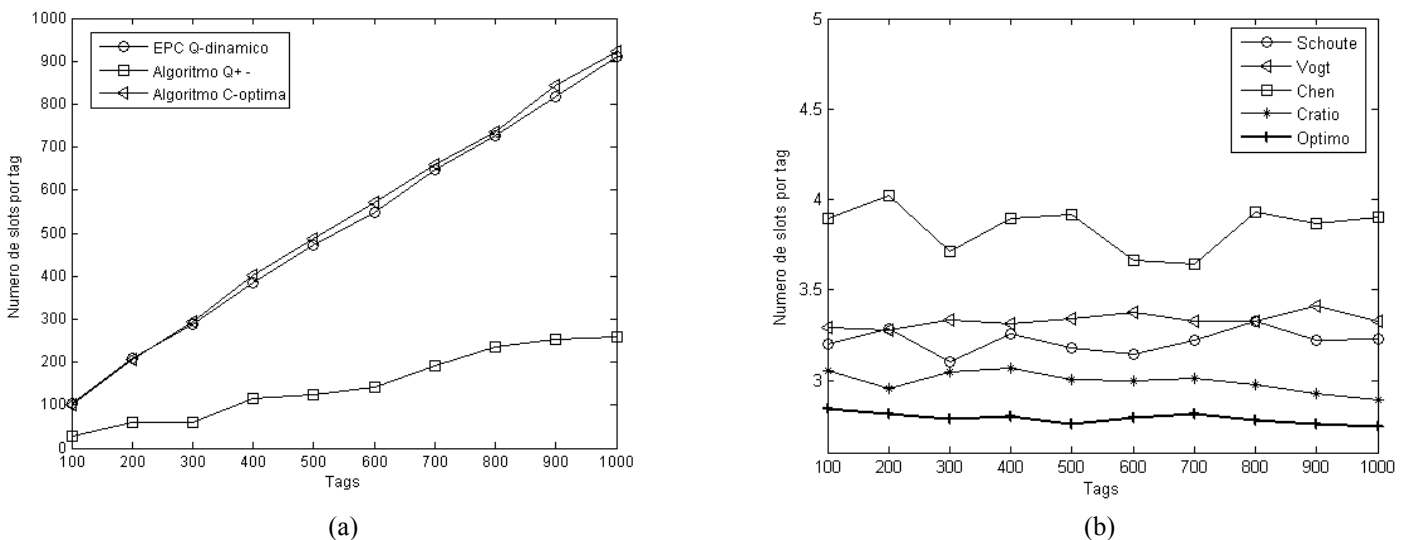


Figura 6 (a) Tasa de identificación vs *tags* en cobertura con mecanismos de corrección de trama, (b) tasa de identificación vs *tags* en cobertura con mecanismos heurísticos y *algoritmo óptimo*.

4. Conclusiones

En este trabajo se comparan diversos algoritmos anticolidión compatibles con el estándar EPCglobal Class-2 Gen-2 y basados en FSA/DFSA. Se ha observado que muchos algoritmos no adaptan la longitud de ciclo a una potencia de 2 entre 1 y 15, no siendo conformes, por tanto, con el estándar EPCglobal Class-1 Gen-2. Se han completado comparativas realizadas entre los algoritmos estudiados, empleando como indicador el tiempo medio de identificación medido en *slots*, y no en ciclos, algo necesario en un escenario donde el tamaño del ciclo es variable. Esto ha permitido sacar a la luz las prestaciones cercanas al óptimo de algunos algoritmos [9] y fuertemente degradadas de otros [5-10].

Agradecimientos

Este trabajo ha sido financiado por los proyectos DEP2006-56158-C03-03/EQUI del Ministerio de Educación y Ciencia y TEC2007-67966-01/TCM (CON-PARTE-1) del Ministerio de Industria, Turismo y Comercio. Asimismo, se ha desarrollado en el contexto del “Programa de Ayudas a Grupos e Excelencia de la Región de Murcia”, de la Fundación

Séneca, Agencia de Ciencia y Tecnología de la Región de Murcia (Plan Regional de Ciencia y Tecnología 2007/2010).

Referencias

- [1] Wieselthier, J. E., Ephremides, A., Michaels, L. A., "An exact analysis and performance evaluation of framed ALOHA with capture", IEEE Transactions on Communications, vol. 37(2), pp. 125-137, 1988.
- [2] "EPC Radio-Frequency Identify protocol for communications at 868-960 MHz, Version 1.0.9: EPCglobal Standard Specification", 2004. Disponible on-line en: <http://www.epcglobalinc.org/standards>
- [3] Alien. Documentación disponible on-line en: <http://www.alientechnologies.com>
- [4] Intermec. Documentación on-line en: <http://www.intermec.es>
- [5] Lee, D., Kim, K., Lee, W. "Q+- Algorithm: An enhanced RFID Tag Collision Arbitration Algorithm". Ubiquitous Intelligence and Computing, LNCS, pp. 23-32. August, 2007.
- [6] Vogt, H., "Efficient Object Identification with Passive RFID tags", Lecture Notes in Computer Science, vol. 2414, pp. 98-113, 2002.
- [7] Joe, I., Lee, J., "A Novel Anti-Collision Algorithm with Optimal Frame Size for RFID System", Fifth International Conference on Software Engineering Research, Management and Applications, 2007
- [8] F.C.Schoute, "Dynamic Frame Length ALOHA", IEEE Transactions on Communications, COM-31(4): pp.565-568. April 1983.
- [9] J. Cha, J. Kim, "Novel Anti-collision Algorithms for Fast Object Identification RFID System", in Proc. of the 11th Conference on Parallel and Distributed Systems, vol. 2, pp.63-67, 2005.
- [10] Tong, Q., Zou, X., Liu, D., Dai, Y. "Modeling the Anti-collision Process of RFID System by Markov Chain", In Proc of. International Conference on Wireless Communications, Networking and Mobile Computing WiCom 2007, pp. 2054-2057, 21-25 September 2007.
- [11] Wen-Tzu Chen, "An efficient Anti-Collision Method for Tag Identification in a RFID System", IEICE Transactions on Communications, Vol. E89-B, no. 12, pp. 3386-3392, 2006.
- [12] Matlab on line: <http://www.mathworks.com>
- [13] M^a Victoria Bueno Delgado, Javier Vales Alonso, Esteban Egea López, Joan García Haro, "Estudio de la configuración óptima de longitud de ciclo en sistemas RFID", aceptado en VII Jornadas de Ingeniería Telemática, JITEL 2008, Alcalá de Henares, 16-18 Septiembre 2008.