

# Herramienta de Simulación del Protocolo EPC GEN2 Class1 de RFID

Hugo Landaluce, Asier Perallos and Ignacio Angulo  
Instituto Tecnológico Deusto (DeustoTech), Universidad de Deusto  
Avda. de las Universidades, 24. 48007 Bilbao, España  
{hlandaluce; perallos; ignacio.angulo}@deusto.es

**Abstract**—Debido al creciente desarrollo de la tecnología RFID, las aplicaciones en las que se ve involucrada esta tecnología son cada vez más sofisticadas, empleando grandes cantidades de tags. Por ello es importante disponer de una herramienta de testeo de la aplicación antes de implementarse de forma definitiva. En este artículo se presenta un simulador del protocolo EPC Gen2 Class1 (para tags pasivos), para permitir al usuario conocer el comportamiento del sistema RFID con un determinado número de tags, ya que es uno de los protocolos más empleados actualmente. Para la validación del simulador, se han realizado una serie de pruebas con un lector RFID comercial y se han comparado las latencias obtenidas con las proporcionadas por el simulador.

**Keywords**—RFID; simulador; EPC; Gen2; anti-colisión

## I. INTRODUCCIÓN

La identificación por radio frecuencia (RFID) es uno de los principales métodos de auto identificación (autoID) cuyo principal cometido es leer los códigos almacenados en pequeñas etiquetas que a su vez son colocadas en objetos de diferentes tipos, empleando para ello las ondas de radio frecuencia (RF). Estas etiquetas o tags se adhieren a los objetos que se desean identificar o monitorizar sin necesidad de la existencia de una línea de visión directa.

Un sistema RFID está compuesto principalmente por tres elementos [1]:

- Uno o varios tags: también conocidos como transpondedores. Están formados por un microchip y una pequeña antena y se adhieren al objeto que se desea identificar. Los tags pueden ser activos o pasivos. Los activos se alimentan por medio de una batería. Mientras que los pasivos obtienen la alimentación de la señal que el lector envía en el momento de la detección, y por ello el alcance de estos últimos es significativamente menor.
- El lector: compuesto por un módulo de RF, una unidad de control y una o más antenas. Ofrece una comunicación bidireccional con los tags y con algún dispositivo o subsistema externo.
- Subsistema de almacenamiento/procesamiento de datos: conectado al lector, puede actuar como aplicación de control de éste o de almacenamiento de los datos que le proporciona.

RFID se está empleando de forma cada vez más frecuente como tecnología de auto identificación (autoID) ya que, no solo cubre las mismas funcionalidades que tecnologías análogas como el código de barras, sino que además mejora sus prestaciones. No requiere intervención de una persona para la detección, no es necesaria la existencia de una línea de visión directa entre lector y tag para detectarlo gracias a las propiedades de la RF y posee mayor memoria de almacenamiento (64 bits, 96 bits y algunos tipos de tags mayor cantidad). Dado que los tags pasivos son significativamente más baratos que los activos y no usan baterías, su uso es más común, sobre todo en aplicaciones de seguimiento, trazabilidad o control de stock. Además, esta tecnología se está mostrando muy prometedora por las características mencionadas y porque su naturaleza encaja muy bien en el paradigma de la computación ubicua [2]. Por ello se ha empezado a utilizar a lo largo de toda la cadena de suministro y en aplicaciones de automatización industrial. Aun considerando todas estas ventajas, los códigos de barras no van a ser sustituidos a corto plazo por RFID. Dado que éstos están ampliamente extendidos, se prevé que ambas tecnologías convivan aprovechándose de sus respectivas ventajas.

Actualmente la tecnología RFID se encuentra a un nivel en que sus aplicaciones son cada vez más sofisticadas, las cantidades de tags son mayores y los lectores tienen que trabajar en entornos muy ruidosos. A causa de todos estos factores, aparece la necesidad de testear la aplicación antes de su implementación definitiva. Pero dicho testeo conlleva disponer de equipamiento de alto nivel dadas las limitaciones impuestas por los tags pasivos y sus limitaciones de coste. En este paper se desarrolla un simulador que cubre esta necesidad. Un simulador del protocolo EPC Gen2 Class1 (para tags pasivos) para permitir al usuario conocer el comportamiento del sistema RFID con un determinado número de tags.

En los sistemas RFID que trabajan en el rango ultra alto de frecuencias (UHF) suele existir más de un tag en la zona de interrogación. Y dado que todos los tags emplean el mismo canal para la transmisión, es posible que varios de ellos contesten al mismo tiempo. Si se da el caso, se produce una colisión de señales, es decir un solapamiento de señales que las hace ininteligibles para el lector lo que produce pérdidas de ancho de banda y un incremento en el tiempo de detección de objetos. Este fenómeno se denomina “el problema de las colisiones de tags”. Para evitarlo se necesitan procedimientos de acceso múltiple [3]. Entre los que se encuentra el algoritmo

Slot Counter utilizado por el protocolo EPC Gen2 Class1 e implementado en el simulador presentado.

Varios simuladores de este protocolo se han realizado. En [4] y [5], Cheng y Helmut realizan varias simulaciones de protocolos basados en Aloha, incluyendo el estudiado en este artículo. Y en [6], Floerkemeier diseña un completo simulador del protocolo EPC Gen2. Pero en ninguno de los casos presentados, realizan una validación con un lector RFID real, tomando medidas de tiempo y comparándolas con las simulaciones realizadas.

El resto del artículo está organizado de la siguiente manera: En la Sección 2 se explican las principales características del protocolo EPC Gen2 Class1. En la Sección 3 se describen las pruebas realizadas con el lector RFID real y en la Sección 4 el simulador desarrollado. Para, en la Sección 5, realizar la comparación de los resultados obtenidos por ambos caminos. Por último, en la Sección 6 se finaliza con las conclusiones obtenidas.

## II. PROTOCOLO EPC GEN 2 CLASS 1

Acogido en la norma ISO 18000-6C. La indicación “Class1” se refiere a la subsección de tags pasivos [7][5]. Se trata de un protocolo Reader Talk First (RTF), es decir, el lector siempre es el que inicia la comunicación. Y el rango de frecuencias de operación es UHF, más concretamente entre 860 y 960 MHz según indique el status de regulación de cada país [8].

El protocolo se divide en dos capas: una capa física y una capa de identificación.

### A. Capa física

En la capa física se presentan las posibles modulaciones y codificaciones que debe soportar un dispositivo que cumpla este protocolo. Además se especifican las características que debe cumplir la onda continua (CW) que se emplea para alimentar los tags pasivos y los parámetros para la comunicación por backscatter.

#### 1) Comunicaciones lector-tag

Las modulaciones permitidas por el protocolo son: Single (SSB) ó Double (DSB) Side Band, ASK y PR-ASK. Siendo esta última, la empleada por el lector utilizado en este artículo. La codificación de los datos se realiza mediante Pulse Interval Encoding (PIE). El  $T_{ari}$  es la referencia de intervalo de tiempo para la comunicación en el sentido lector-tag y marca la duración de un 0 binario. El 1 binario se representa con un símbolo de duración 1.5 a 2.0  $T_{ari}$  [7].

El lector siempre comienza la comunicación con un preámbulo ( $P$ ) o un sincronizador de trama ( $FS$ ). El preámbulo se compone de un delimitador de 12.5  $\mu s$  aproximadamente, un 0 binario, un símbolo de calibración lector-tag ( $RTCal$ ), que equivale a la duración de un 0 binario más un 1 binario, y un símbolo de calibración tag-lector ( $TRCal$ ). Este último se emplea para que el tag calcule la velocidad de su respuesta o Backscatter Link Frequency ( $BLF$ ) en base a (1), siendo  $DR$  un ratio de división que puede adquirir los valores 64/3 y 8. El sincronizador de trama ( $FS$ ) es igual que el preámbulo ( $P$ ) pero no incluye el símbolo  $TRCal$  [7].

$$BLF = DR/TRCal \quad (1)$$

#### 2) Comunicaciones tag-lector

El fabricante del tag escoge la modulación que va a soportar el tag entre ASK y PSK y el lector le dice al tag cómo debe codificar los datos mediante un comando enviado al comienzo de cada ciclo de lectura. Por ello el tag debe ser capaz de codificar tanto en FM0 como en Miller.

- FM0 banda base, invierte la fase de cada límite de símbolo. Además, cada 0 binario tiene un cambio de fase adicional.
- La codificación Miller invierte la fase entre dos 0 binarios consecutivos y en la mitad de cada 1 binario. La onda transmitida se forma multiplicando  $M$  veces la portadora por una onda cuadrada.

Al final de las transmisiones, ambas codificaciones terminan con un bit “dummy” o bit innecesario [7].

### B. Capa de identificación

Un lector RFID controla los conjuntos de tags utilizando 3 operaciones básicas:

- Selección: Se aplica para seleccionar un subconjunto particular de tags basado en un criterio específico.
- Inventario: Un lector comienza un ciclo de lectura transmitiendo un comando *Query*. Uno o más tags responden enviando su código electrónico de producto (EPC) siguiendo el dictado del algoritmo *Slot Counter*.
- Acceso: La operación mediante la cual se comunica (leyendo o escribiendo) con un tag, previamente identificado.

#### 1) Algoritmo Slot Counter

El algoritmo de anti-colisión empleado en el protocolo adopta una variación del Dynamic Framed Slotted Aloha (DFSA) [9]. También es conocido como algoritmo Q, y su principal diferencia con el protocolo DFSA es que la actualización del tamaño del frame se puede producir en cada slot o respuesta de tag. Para decidir si debe actualizar ese tamaño, el algoritmo emplea un sistema de actualización de un parámetro, denominado  $Q$ , en función del tipo de respuesta recibida en la iteración anterior. Si en la respuesta se produce una colisión, el lector incrementa un valor  $Q_{fp}$  que luego al redondearlo se convierte en  $Q$ . Si no responde ningún tag, el lector reduce el valor de  $Q_{fp}$  y si la respuesta es única este valor no cambia. Una vez redondeado, el valor se envía a los tags en los comandos *Query* o *QueryAdj* y el tag escoge un valor entre 0 y  $2^Q$  como slot en el que transmitir.

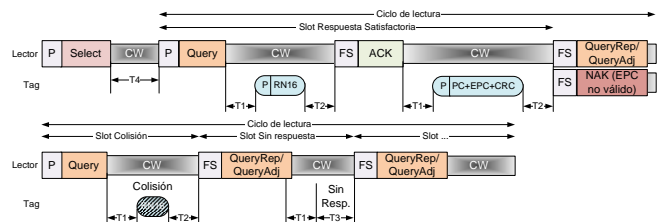


Figura 1. Comunicación Lector - Tag [5].

## 2) Interacción lector-tag

Como se muestra en la Fig.1, el lector comienza la comunicación enviando un comando *Select* con el que selecciona un subconjunto de la población de tags. Existen 4 sesiones en las que pueden trabajar los tags, y el lector selecciona una de ellas para realizar el ciclo de lecturas.

Mediante la sesión, el protocolo controla con qué frecuencia van a responder los tags, cuando el lector les interroga por 2ª vez, siendo las sesiones de número menor las que responden con mayor frecuencia. Además, los tags poseen un flag (*SL*), que también puede ser empleado para separar unos tags de otros, controlando la participación de los que poseen el flag activo o de los que no.

Después de un tiempo mínimo entre comandos,  $T_4$ , se envía un *Query* al subconjunto de tags. Este comando define las características del ciclo de lectura, como qué codificación se va a emplear o el tamaño inicial del frame ( $2^Q$ ). Los tags que cumplen las restricciones cambian su estado a *Arbitrate*, como se puede observar en la Fig. 2. Durante un tiempo  $T_1$ , dichos tags generan un número aleatorio (*RN16*) entre 0 y  $2^Q$  e inicializan sus contadores internos con ese valor. Si el contador de algún tag vale 0 cambia su estado a *Reply* y responde su *RN16* al lector. Como se puede ver en la Fig. 1, se pueden dar 3 tipos de respuesta en cada slot:

- *Slot con una respuesta:* si únicamente se envía un *RN16* a la vez estamos ante una situación de transmisión satisfactoria. En tal caso el lector contesta un mensaje de reconocimiento (*ACK*) conteniendo el *RN16* del tag recibido tras un tiempo  $T_2$ . El estado del tag cambia a *Acknowledged* y finalmente, después de un tiempo  $T_1$ , envía su EPC. Si el lector lo recibe correctamente, continúa la identificación y el tag pasa a estado *Ready*, donde permanece en silencio hasta un nuevo comando *Query*. Si se produce un error de transmisión, el lector solicita de nuevo el EPC del tag con un mensaje *NAK*.
- *Slot con colisión:* si varios tags envían su *RN16* al mismo tiempo, el lector incrementa el valor de  $Q_{fp}$  y si tras redondearse ( $Q$ ), su valor no ha cambiado con respecto al slot anterior, no es necesario enviar un nuevo tamaño de frame. Por tanto, el lector envía un comando *QueryRep* que disminuye los contadores de los tags. Los que habían contestado reinician su

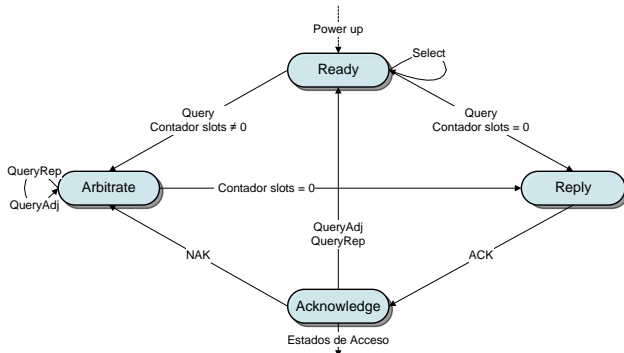


Figura 2. Estados de Inventario de los tags simplificados (no están todas las transiciones).

contador no interfiriendo en la ronda siguiente, y el resto le restan 1 y se repite el proceso. Si por el contrario el valor de  $Q$  ha cambiado con respecto al slot anterior, el comando que se envía es *QueryAdj*, que indica a los tags el nuevo tamaño del frame, y éstos recalculan su valor de contador entre los nuevos límites.

- *Slot sin respuesta:* en caso de que no conteste ningún tag en el slot actual, se considera un slot vacío o sin respuesta. El lector espera un tiempo  $T_3$ , y si en ese tiempo no recibe nada considera el slot vacío. El lector recalcula  $Q_{fp}$  reduciéndolo y sigue el mismo proceso que en el caso de colisión.

## III. PRUEBAS DEL LECTOR RFID

Las pruebas de lector se han realizado con un Mercury5e-EU UHF [10] conectado a una antena polarizada circularmente Poynting PATCH-A0025 y trabajando a una frecuencia de 860 MHz. Como subsistema de control y almacenamiento se ha empleado un ordenador conectado por RS-232 al lector RFID. Este subsistema ejecuta una aplicación que intercambia comandos de funcionamiento con el lector y almacena sus resultados. El modelo de tag empleado ha sido MR2052.

Se han realizado lecturas sobre grupos de entre 5 y 85 tags. Los tags se han situado cerca de la antena, a 1 m de distancia, de manera que no perdieran alimentación durante el ciclo de lectura. Además se han situado juntos para evitar, en la medida de lo posible, el efecto de captura y para facilitar que todos los tags respondan con la misma fuerza.

El Mercury5e-EU realiza ciclos de lectura y almacena EPCs de 96 bits de longitud, además de otro tipo de información útil en un buffer interno de 200 bits por entrada. El software desarrollado controla el lector, ordenándole cuándo comenzar y parar la toma de datos, y almacenando los resultados en tablas y archivos. A la información adicional del EPC recibido del lector se le denomina “Meta-data” y se compone de: un indicador de fuerza de la señal de respuesta (RSSI), la frecuencia a la que se ha leído el tag y el tiempo que ha tardado en leerlo desde que envía el primer comando hasta que recibe el último bit del EPC en milisegundos. Para la comparación con la simulación se ha empleado el tiempo total de lectura de todos los tags existentes en la zona de interrogación (latencia). La aplicación desarrollada también controla los parámetros de configuración del lector que se pueden observar en la Tabla 1. Además, de gestionar las

TABLA I. PARÁMETROS DEL LECTOR RFID

Parámetro	Valor
Ganancia Lectura	30 dBm
M	4
Q inicial	3
Q actualizado	Dinámico
Sesión	S1
Target	Sólo A

lecturas, la aplicación almacena los tiempos recibidos del primer y último tag leídos en el ciclo de lectura realizado, para conocer más información de todo el proceso de detección y calcula el tiempo total de lectura de todos los tags presentes, teniendo como referencia el dato del primer tag leído.

El lector trabaja en modo “Portal”. En este modo se define que los tags deben trabajar en la sesión 1. Con la sesión elegida, se pretende que cada tag sea leído una única vez en cada ciclo de lectura, facilitando la rápida detección de todos los tags del rango de interrogación.

#### IV. SIMULADOR DE EPC GEN 2 CLASS 1

Este simulador se ha desarrollado empleando la herramienta Matlab R2009b y siguiendo las especificaciones propuestas por el protocolo EPC Gen2 Class1. El principal objetivo del simulador es mostrar el comportamiento a nivel lógico del protocolo EPC, centrándose en el algoritmo Slot Counter, para conocer su evolución ante diferentes cantidades de tags en su zona de interrogación. Parámetros como el máximo número de slots en un ciclo de lectura, el número de colisiones, slots sin respuesta o la eficiencia, son necesarios para evaluar la puesta en escena de un algoritmo de anti-colisión. Además de obtener todos estos parámetros, el simulador también controla el tiempo empleado en la lectura de los tags presentes en la zona de interrogación o latencia. Para conocer este dato, es importante mencionar que se requieren una serie de características de la capa física del protocolo, como los tiempos de propagación o el tipo de modulación. En la especificación del protocolo EPC Gen2, se especifican cómo deben ser los tiempos entre comandos de tag y lector (T1, T2, T3, T4), la longitud de los símbolos y el contenido de los preámbulos. Para realizar una simulación lo más fiel posible al comportamiento de un lector real, se han medido los valores

TABLA II. PARÁMETROS DEL LECTOR PARA EL SIMULADOR.

Parámetro	Valor
Tari	25 μs
RTCal	65 μs
TRCal	84.5 μs
DR	64/3
BLF	252.47 kHz
M	4
Q inicial	3
TRext	0
Data Rate Lector	40 kbps
Data Rate Tag	63.31 kbps
T1	170.8 μs
T2	16.24 μs
T3	85.5 μs
T4	130 μs

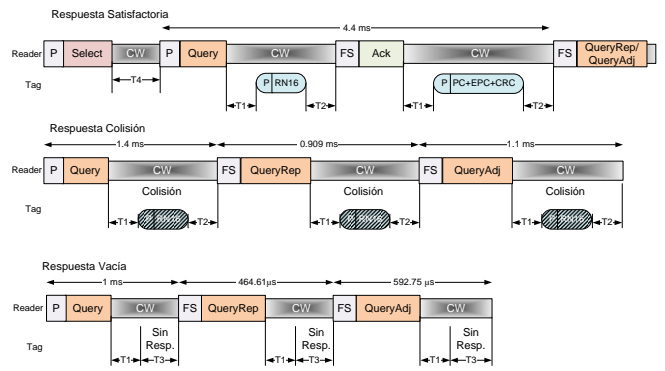


Figura 3. Cálculo de tiempos del simulador.

empleados por el lector con el que se han realizado las pruebas y se han aplicado a las simulaciones. Son los mostrados en la Tabla 2.

En la Fig. 4 se muestra un diagrama de flujo del proceso que sigue el simulador. Tras inicializar los parámetros necesarios con los valores mostrados en la Tabla 2 se envía un comando *Select* a los tags existentes en el rango de lectura. A continuación se envía el comando *Query* con la codificación, *M* y *BLF* que deben usar en el ciclo y el valor inicial de *Q*. Los cálculos de tiempos se realizan teniendo en cuenta la velocidad de transmisión tanto de lector como de tag, el número de bits de cada comando y los tiempos entre mensajes para las tres respuestas posibles, como se muestra en la Fig. 3.

Por tanto, tras enviar el comando *Query*, se realizan los cálculos de tiempo con *T4* y *T1*. Los tags escogen aleatoriamente un slot teniendo en cuenta *Q*, y aquellos cuyo contador vale 0 responden enviando su *RN16*. En la Fig. 3, se observan las tres respuestas ya mencionadas y cada una de ellas afecta de forma diferente al tiempo calculado. Si no hay respuesta de tag el lector espera un tiempo *T3* y disminuye  $Q_{fp}$  con el valor de la variable *C*. Si únicamente responde un tag, la respuesta es satisfactoria. Actualiza el tiempo con *T2* y el tiempo de *RN16*, y a continuación suma *T1* y el tiempo de transmisión del comando *ACK*. Y por último, si se produce una colisión por la respuesta simultánea de varios tags, tras añadir *T2* al tiempo de ciclo, aumenta  $Q_{fp}$  con el valor de *C*. Este parámetro *C* se calcula utilizando (2), propuesta por [5], para obtener el mejor comportamiento.

$$C = 0.8/Q_{fp} \quad (2)$$

Tras la actualización de  $Q_{fp}$  se redondea su valor, y si todavía quedan tags por detectar, el simulador vuelve a interrogar a los tags. Mirando el valor ya redondeado *Q*, se decide si enviar un *QueryAdj* o bien un *QueryRep* que disminuye en 1 el contador de los tags para repetir de nuevo el proceso anterior (Ver Fig. 4).

##### A. Validación del simulador

Al inicio de la simulación, se realiza una comparación entre la disposición de los tags en el frame indicado por el primer comando *Query* y los valores de las expresiones teóricas, como se realiza en [6]. Se emplean las formulas (3,4,5).

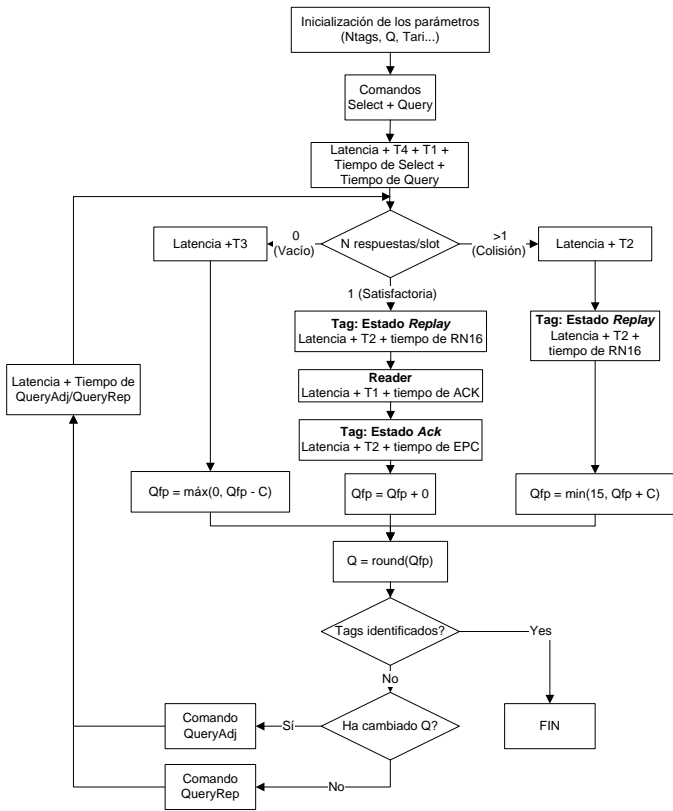


Figura 4. Diagrama de flujo del simulador del protocolo EPC Gen 2.

$$S(N, L) = N \left(1 - \frac{1}{L}\right)^{N-1} \quad (3)$$

$$H(N, L) = L \left(1 - \frac{1}{L}\right)^N \quad (4)$$

$$C(N, L) = L - S(N, L) - H(N, L) \quad (5)$$

$S$  es el número de slots satisfactorios o con una única respuesta de tag.  $H$  es el número de slots vacíos o sin respuesta y  $C$  es el número de slots con colisión en el frame de tamaño  $L$ . El número total de tags en la zona de interrogación está representado por  $N$ . A continuación, en la Fig. 5, se muestran los resultados para el caso en que la  $Q$  inicial es 3, es decir, que el tamaño inicial del frame son 8 slots. También, para un valor escogido al azar como  $Q$  igual a 6, con tamaño del frame 64, que permite conocer la respuesta del protocolo de forma diferente, es mostrado en dicha figura. Para cada conjunto de tags se muestran 6 cantidades. Los 3 valores teóricos ( $T$ ) de slots satisfactorios, vacíos o en colisión y las 3 medias de los valores obtenidos en la simulación ( $S$ ) de esas mismas variables en 100 ejecuciones de la simulación.

## V. COMPARACIÓN: SIMULADOR VS LECTOR RFID

Para establecer una comparativa entre la simulación y la realidad se han superpuesto los resultados de ambos procesos sobre la misma gráfica en Matlab. En el proceso de toma de medidas del lector RFID, se han realizado dos por cada ciclo de lectura, el tiempo que tarda en leer el primer tag y el tiempo

que tarda en leer el último (latencia). La aplicación ha almacenado esos valores en un archivo que posteriormente se representado en dicha gráfica. Posteriormente, el resultado de la simulación propuesta con los mismos parámetros del lector se ha representado en la misma gráfica. Dado que el interés del experimento reside en observar la evolución de la latencia en la detección de los tags, no se le ha dado importancia al hecho de que en la zona de interrogación existieran p. ej. 30 tags y el lector detectara 28. El dato del tiempo se ha considerado aceptable y se ha representado como tal.

Se ha observado que la respuesta del simulador tiene un error de offset con respecto a la recta de mínimos cuadrados de las muestras obtenidas. Gracias a las medidas tomadas de las lecturas del 1<sup>er</sup> tag de ciclo, se ha observado que ese tiempo es prácticamente constante e independiente del número de tags existentes en el rango de lectura, siendo de 50 ms aproximadamente. Durante este tiempo el lector realiza una serie de operaciones desconocidas para los autores de este artículo. Por tanto, se ha optado por tomar las medidas referenciadas al tiempo de lectura del primer tag, que es donde se empiezan a aplicar las directrices del protocolo.

En la Fig. 6 se muestra una simulación de las latencias de lectura de grupos de 5 a 85 tags con incrementos de 5, habiéndose realizado 100 ciclos completos con cada grupo. De cada uno de los ciclos se ha obtenido y mostrado la latencia media referenciada a la lectura de tiempo del 1er tag de su correspondiente ciclo. Frente a las latencias medias de la simulación, se muestran las latencias referenciadas del mismo modo que en la simulación, de cada correspondiente ciclo de lectura del lector RFID real. Se han representado un total de 50 muestras por cada conjunto de tags, entre 10 y 80 aproximadamente, con incrementos de 10 tags de un grupo a otro.

La gráfica muestra cómo la simulación cumple para las muestras más rápidas del lector RFID. Dado que el propósito del desarrollo del simulador es imitar y mostrar el comportamiento del algoritmo *Slot Counter*, se puede dar por buena dicha simulación, teniendo en cuenta que los datos

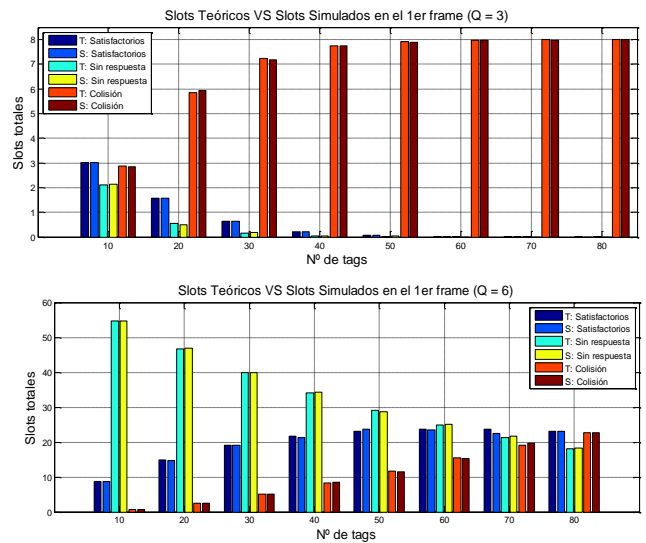


Figura 5. Slots Teóricos VS Slots Simulados.



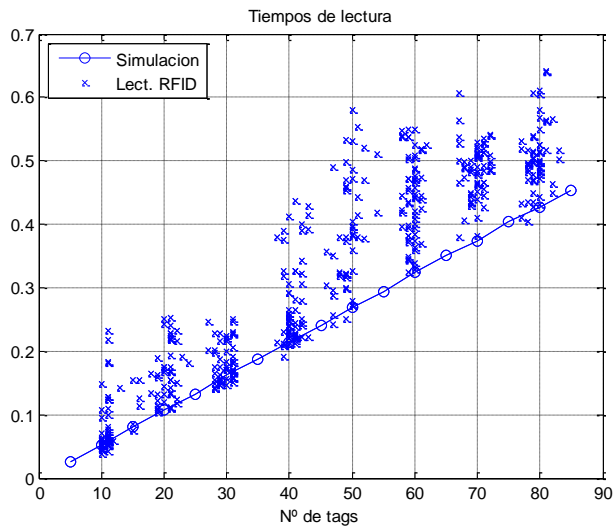


Figura 6. Latencias referenciadas al tiempo del 1er tag.

obtenidos son para un caso considerado prácticamente ideal. La dispersión existente en las muestras tomadas, indica que se producen errores de transmisión que aumentan la latencia de cada ciclo. Estos errores provocan un aumento de mensajes de tipo NAK, que aumentan la latencia total.

#### A. Trabajo futuro

Como trabajo futuro se propone investigar el tiempo inicial que no está especificado en el protocolo y que es prácticamente constante e independiente del número de tags presentes en la zona de interrogación. Para ello se pretende medir con una antena circular la comunicación lector-tag y analizar las tramas enviadas por ambos, para poder discernir qué se está llevando a cabo durante ese tiempo.

Además, se pretende ir completando las funcionalidades del simulador, añadiendo el modelo físico del escenario donde se sitúan los tags, simulando la propagación de la señal y teniendo en cuenta el efecto de captura, de forma que la simulación sea cada vez más fiel al lector real. También se pretende implementar otros protocolos de anti-colisión diferentes, tanto de naturaleza probabilística como determinista.

## VI. CONCLUSIONES

La tecnología RFID está creciendo muy rápido, y cada vez es más necesario testear la aplicación antes de su implementación definitiva. Con esta finalidad, se ha desarrollado un simulador del protocolo EPC Gen2 Class1 (para tags pasivos) para permitir al usuario conocer el comportamiento del sistema RFID con un determinado número de tags.

Con el objetivo de la validación del simulador, se han comparado los resultados teóricos con los simulados de forma satisfactoria. También, se ha comparado el resultado de la latencia o tiempo máximo de lectura de tags obtenida del simulador con los tiempos medidos en un lector RFID comercial. En el caso de las medidas referenciadas al tiempo de detección del primer tag, éstas coinciden con las simuladas en los mínimos valores medidos. Ésto demuestra que el simulador

es fiable, pero solo tiene en cuenta los casos más ideales. La dispersión en las medidas tomadas, afirma que en un sistema real siempre se van a producir errores de transmisión que deben considerarse como estudio futuro para una ampliación de la funcionalidad del simulador.

## REFERENCIAS

- [1] K. Finkenzeller, *RFID Handbook: Fundamentals and applications in contactless smart cards and identification*, 2nd ed., John Wiley & Sons Ltd, 2003.
- [2] V. Stanford, *Pervasive Computing Goes the Last Hundred Feet with RFID Systems*, *IEEE Pervasive Computing*, vol. 2, no. 2, pp. 9-14, April-June 2003.
- [3] D.K. Klair, K-W Chin, R. Raad, *A survey and tutorial of RFID anti-collision protocols*. *IEEE Communications Surveys & Tutorials*, vol. 12, no. 3, pp. 400-421, August 2010.
- [4] T. Cheng, L. Jin, *Analysis and Simulation of RFID Anti-collision Algorithms*, *The 9th International Conference on Advanced Communication Technology*, vol. 1, pp. 697-701, Feb. 2007
- [5] W. Helmut, "A simulation tool for RFID anti-collision algorithms based on ALOHA", *16th International Conference on Systems, Signals and Image Processing*, pp. 1-4, 2009.
- [6] C. Floerkemier, S. Sarma, "RFIDSim-A physical and logical layer simulation engine for passive RFID", *IEEE Transactions on Automation Science and Engineering*, vol. 6, no. 1, pp. 33-43, Enero 2009.
- [7] EPC Global, "EPC™ Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz – 960 MHz Version 1.2.0", Octubre 2008.
- [8] GS1 EPC Global, *Regulatory status for using RFID in the UHF spectrum*, pp 14, Noviembre 2011.
- [9] L. Barletta, F. Borgonovo, M. Cesana, *Performance of Dynamic-Frame-Aloha protocols: Closing the gap with tree protocols*, *The 10th IFIP Annual Mediterranean Ad Hoc Networking Workshop*, pp. 9-16, June 2011.
- [10] Mercury5e RFID reader, <http://www.thingmagic.com/embedded-rfid-readers/mercury5e/1-embedded-rfid-readers/3-mercury5e>