



Evitando el efecto *random walk* en osciladores de bajo coste para mejorar la sincronización entre nodos WSN

S. Felici-Castell*, JJ. Perez-Solano*, A. Soriano-Asensi*, J. Segura-Garcia*, E. A. Vargas,[†] G. Mas[‡]

* ETSE, Universitat de València,[†] DEI, Universidad Católica Asunción (PY),[‡] BSG Ingenieros
felici@uv.es, jjperezs@uv.es, soan@uv.es, evargas@uc.edu.py, gemma@bsg.es

Resumen

Las redes de sensores inalámbricos (Wireless Sensor Nodes) están en continua evolución y mejorando sus prestaciones introduciendo nuevas tecnologías. Para poder desplegarse a gran escala, estas redes se basan en nodos inalámbricos de bajo coste. Los componentes de bajo coste, y en particular los sistemas de reloj de estos nodos, les impiden lograr una sincronización precisa, que es importante para reducir el número de paquetes transmitidos así como necesaria para ciertas aplicaciones (localización de eventos sonoros, beamforming colaborativo, etc.). Para mejorar la sincronización entre nodos, en este artículo proponemos el uso de un nodo inalámbrico basado en tecnología Ultra Wide Band utilizando componentes disponibles en el mercado, como el transceptor DWM1000 de Decawave. Para ello, modelamos el comportamiento de los relojes utilizando paseos aleatorios (random walks) y proponemos diferentes alternativas para conseguir una sincronización temporal a escala de subnanosegundos utilizando regresiones, con un protocolo de sincronización "one way". Se concluye que con estos nodos se puede llegar a conseguir un error de sincronización temporal de 149 ps entre transceptores.

Palabras Clave—redes dedicadas, ultra wide band, wireless sensor networks, time synchronization, random walk

I. INTRODUCCIÓN

Las Redes de Sensores Inalámbricos (Wireless Sensor Nodes, WSN) se enfrentan a nuevos retos y aplicaciones complejas como la vigilancia, la localización de eventos sonoros, la formación de haces colaborativos [1] por nombrar algunos. Los despliegues de estas redes suelen tener requisitos de bajo coste y bajo consumo de energía. Estas redes se basan en nodos inalámbricos colaborativos que tienen sus propias unidades de procesamiento, sensores, memorias y comunicaciones inalámbricas, basadas en componentes de bajo coste para abaratarlas.

En este contexto, el estándar IEEE802.15.4 centrado en capas físicas para velocidades binarias bajas y de bajo consumo, propone las tecnologías Ultra Wide Band

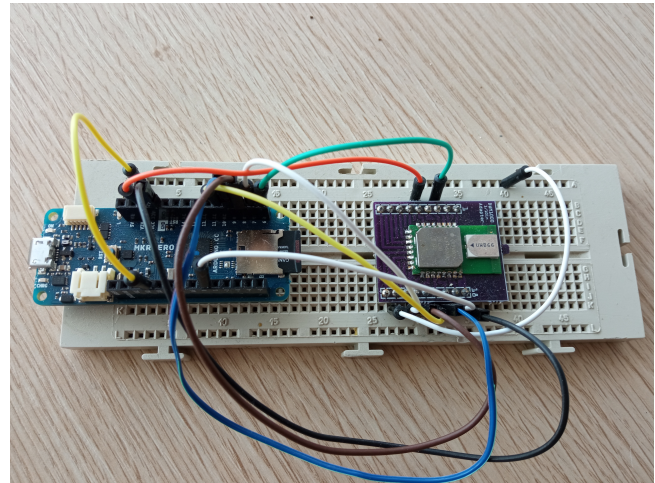


Figura 1: Foto del prototipo del nodo inalámbrico basado en el transceptor Decawave DWM1000 UWB conectado a Arduino MKRZero

(UWB) como una de las alternativas a considerar en WSN. Nos centraremos en esta tecnología, conocida como 'pulse radio', ya que se basa en la transmisión de pulsos de corta duración que requieren de un gran ancho de banda. La tecnología UWB permite realizar mediciones precisas del tiempo de vuelo (Time of Flight, TOF) en entornos densos con múltiples trayectorias.

Existen transceptores UWB comerciales, como el DWM1000 [2] de Decawave, que cumplen con la norma IEEE802.15.4 y que pueden conectarse a un microcontrolador (MCU) como se muestra en la Fig. 1, cumpliendo los requisitos mencionados anteriormente (bajo coste y bajo consumo). Este nodo inalámbrico es un prototipo utilizado y testeado en el presente trabajo.

Este transceptor UWB utiliza el marcado de tiempo (time stamping) basado en el mecanismo conocido como Delimitador de Inicio de Trama (Start Frame Delimiter,

SFD), permitiendo introducir la marca de tiempos en las tramas cuando se detecta este campo SFD, tanto en las tramas entrantes como en las salientes. Esta característica hace muy interesante la medición del TOF en aplicaciones de localización, para las que fue diseñado este transceptor [3]. Aprovechando la sinergia entre las aplicaciones de localización y la sincronización temporal. Nuestro objetivo es explotar las interesantes características de este transceptor desde un punto de vista diferente, para un uso distinto al de las aplicaciones de localización.

El mecanismo de marcado de tiempo SFD reduce los retrasos no deterministas introducidos por el proceso de comunicación, en particular por la capa de control de acceso al medio (Media Access Control, MAC). Este mecanismo también se ha explotado en otros protocolos de sincronización utilizados en las WSN [4] [5] [6]. Además, estos protocolos se basan en la sincronización temporal a largo plazo, utilizando algoritmos de regresión lineal sobre las últimas marcas de tiempo observadas.

Sin embargo, los relojes de estos nodos en general sufren derivas de reloj, produciendo desviaciones de reloj entre ellos. La desviación entre dos relojes (o skew) se define como la tasa de cambio entre ellos. Estas derivas se deben a su bajas prestaciones, y en particular se ven afectadas por las condiciones ambientales (como la temperatura, la humedad, el voltaje, el envejecimiento, etc.) [7][8], con valores típicos en el rango de 1 parte por millón (ppm) a 100 ppm [9]. Se puede considerar que el skew del reloj varía dinámicamente siguiendo una distribución de ruido blanco gaussiano, cuyo efecto se conoce como Random Walk (RW) [10] [11]. En este trabajo se analiza este comportamiento y tras revisar los trabajos relacionados, se plantean diferentes alternativas para mejorar la sincronización temporal en el diseño propuesto (Fig. 1), con el fin de conseguir una sincronización por debajo de nanosegundos, evitando en lo posible el efecto RW.

El resto del trabajo está estructurado como sigue. En la sección II, mostramos el trabajo relacionado. En la sección III, analizamos el efecto del paseo aleatorio en la desviación del reloj entre los nodos de la WSN. En la Sección IV, introducimos un sencillo protocolo de sincronización. En la Sección V, describimos la propuesta de nodo inalámbrico basado en UWB. En la Sección VI, presentamos los resultados y realizamos un análisis exhaustivo de las diferentes marcas de tiempo utilizando la característica SFD. Finalmente, en la Sección VII, resumimos las principales conclusiones del trabajo.

II. ESTADO DEL ARTE

La sincronización es un tema candente en las WSN. Cabe mencionar protocolos de sincronización como Timing-sync Protocol for Sensor Networks (TPSN) [12], Flooding Time-Synch Protocol (FTSP) [5], Rate Adaptive Time Synchronization (RATS) [4] por nombrar algunos. Estas referencias logran una precisión de sincronización de tiempo del orden de microsegundos, utilizando motas tradicionales como TelosB [13].

Sin embargo, utilizando la tecnología UWB, podemos mejorar esta precisión de sincronización temporal. En [14],

utilizando el kit Decawave DWM1001 Real-Time Location System (DRTLS) (basado en el transceptor DWM1000 UWB), con el que los autores consiguen generar pulsos de referencia temporal con una fluctuación máxima de $3,3 \mu\text{s}$ y una desviación estándar de $0,7 \mu\text{s}$ sin necesidad de hardware adicional. En [15], los autores diseñan un nodo inalámbrico basado en el mismo transceptor, pero añadiendo un circuito de bucle de enganche en fase (Phase Lock Loop, PLL) basado en un reloj atómico a escala de chip, que puede lograr una sincronización de reloj mejor que 5 ns entre dispositivos de interior o sin GPS, con una media de 2,12 ns (desviación estándar de 0,84 ns). Aunque este trabajo proporciona resultados muy interesantes, hay que destacar que en realidad el subsistema de reloj (reloj atómico a escala de chip) es caro, no siendo una solución de bajo coste factible en la práctica.

Por último, analizando el efecto RW en los relojes, si un temporizador se deja sin corregir, su crecimiento de error de temporización puede ser modelado por un RW unidimensional [16] o un proceso de Wiener. En [17] se describe un protocolo para estimar el skew y tiempo de vuelo, utilizando un modelo cuadrático para modelar el RW. Basándose en sus mediciones, su modelo es más preciso que un modelo lineal, concluyendo que el uso de un modelo cuadrático permite incrementar la validez temporal y que el modelo lineal puede funcionar sólo para períodos de tiempo cortos.

De todas estas referencias se desprende que la UWB puede proporcionar mejores resultados en la sincronización de las WSN, pero falta la evaluación del rendimiento de esta tecnología aplicada a la sincronización, y en particular cuando se utilizan productos disponibles en el mercado en despliegues reales.

III. EFECTO "RANDOM WALK" EN OSCILADORES DE BAJO COSTES

Para modelar el efecto RW, introduciremos el modelo general del tiempo entre dos relojes. Para ello, podemos asumir un comportamiento lineal para modelar la relación entre el tiempo local de dos nodos como $y(t) = m \cdot x(t) + h$ donde $y(t)$ y $x(t)$ son sus relojes locales, t es el tiempo global, m es la pendiente (por defecto 1), y h es el desfase del reloj en $t = 0$.

Dado que las condiciones ambientales cambian gradualmente, el efecto de RW se introduce sobre la pendiente del reloj (m). Por tanto, esta desviación (o skew) de (m) del reloj en un nodo A respecto a un reloj en el nodo B en el tiempo $t_0 + t$, puede calcularse como:

$$m_A^B(t_0 + t) = m_A^B(t_0) + \int_0^t \eta(u) \cdot du \quad (1)$$

donde $\eta(u) \sim N(0, \sigma_{RW}^2)$, siendo σ_{RW} la desviación estándar característica del RW. Para estos relojes de bajo coste, se asume que σ_{RW} es del orden de 10^{-8} y 10^{-9} [8] con 1 ppm a 100 ppm [9].

Usando el siguiente código Matlab, podemos modelar un RW para $y(t) = m \cdot x(t)$ incluyendo el efecto del ruido como:

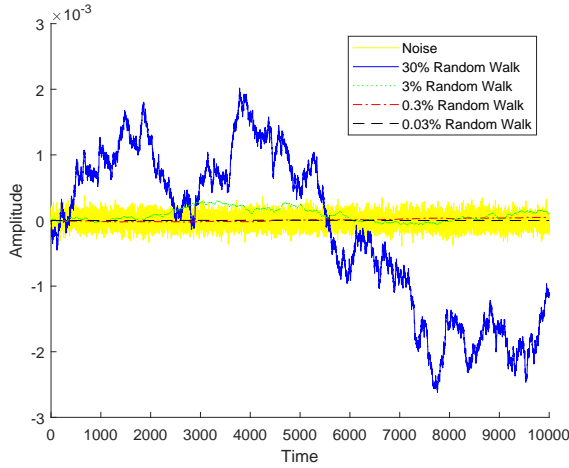


Figura 2: Representación de diferentes señales con ruido de fondo (amarillo) con deriva de 100 ppm, añadiendo efecto “random walk” con distintos % desde 30 hasta 0.03 de σ_{RW} vs ruido con σ_C .

```
%inicialización
rw= zeros(1, longitud);
ruido_w=normrnd(0, sigma_rw, 1, longitud);
ruido=normrnd(0, sigma_c, 1, longitud);

%construyendo el RW
for i=2:longitud
    rw(1, i)=rw(1, i-1)+ruido_w(1, i);
end

%modelo de tiempo añadiendo RW y ruido
y= x + rw.*x + ruido;
```

donde `normrnd` genera una matriz $1 \times longitud$ con distribución $N(0, \sigma)$ y `rw` denota la variable para generar el paseo aleatorio aplicado a la pendiente de y . Además, el ruido añadido (`ruido`) con distribución $N(0, \sigma_C^2)$, siendo σ_C la desviación estándar asociada a la suma de todos estos errores acumulados debidos a los retrasos aleatorios de las comunicaciones.

En la Fig. 2 se muestra el efecto de RW y la desviación que se crea con diferentes contribuciones, % de σ_C , desde el 30 % hasta el 0,03 %, respecto al ruido. En la Fig. 3 se muestra este efecto en el tiempo sobre la pendiente del reloj entre dos nodos.

IV. PROTOCOLO SIMPLE DE SINCRONIZACIÓN “ONE WAY”

La sincronización temporal se consigue mediante el intercambio de marcas de tiempo (time stamps) de los relojes locales, enviadas dentro de paquetes entre los nodos. Hay varios enfoques para estos intercambios, pero en este caso utilizaremos un enfoque simple, un enfoque unidireccional, en el que un nodo, llamado maestro, enviará paquetes de difusión con sus marcas de tiempo locales al resto de nodos (esclavos). Entonces, los esclavos pueden ajustar sus tiempos locales ($y(t_1), y(t_2), ..y(t_i)$) a los proporcionados

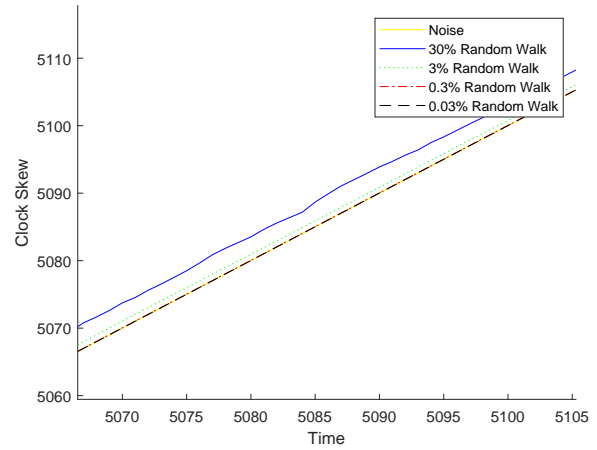


Figura 3: Efecto del “random walks” en la pendiente de un modelo linea de los relojes en segundos.

por el maestro ($x(t_1), x(t_2), \dots, x(t_i)$), siendo i el número del paquete. A partir de estas marcas de tiempo, podemos aplicar diferentes técnicas de regresión para mejorar la sincronización temporal reduciendo el número de paquetes enviados y así predecir las siguientes marcas de tiempo, denotadas como \hat{y}_{i+1} . En este caso, el error de predicción viene dado por $|e_{i+1}| = |y_{i+1} - \hat{y}_{i+1}|$.

Nuestro objetivo es proporcionar un mecanismo para la estimación del tiempo, teniendo en cuenta las marcas de tiempo anteriores, así como la frecuencia utilizada para intercambiar estos paquetes. Esta frecuencia se conoce como periodos de sincronización (SP). Con este propósito, consideraremos diferentes parámetros para el análisis del error de sincronización: SP, tamaños de ventana utilizados para las técnicas de regresión, así como el grado de la misma. Respecto al SP, por un lado, cuanto más rápido intercambiamos paquetes, mayor será la precisión porque el reloj del esclavo no se desviará en exceso respecto al del maestro. Sin embargo, tiene un impacto en el consumo de energía porque estamos enviando más paquetes. Por otro lado, si queremos ahorrar energía reduciendo el número de paquetes intercambiados, la precisión del tiempo se ve afectada negativamente.

Además, el tamaño de la ventana utilizada para la estimación del tiempo, también influye en la minimización del error de sincronización, debido a los procesos aleatorios subyacentes en las desviaciones del reloj en estos nodos, así como el ruido introducido por los procesos de comunicación. Por tanto, dado que existen dos mecanismos subyacentes diferentes e independientes que influyen en estas regresiones, veremos diferentes comportamientos en función de σ_{RW} y σ_C . Por un lado si σ_C es mucho mayor que σ_{RW} , entonces se prefieren tamaños de ventana más grandes para la regresión. Por otro lado, si σ_{RW} es uno o dos órdenes de magnitud menor que σ_C , entonces se prefieren tamaños de ventana más cortos. Si ambos son similares o si σ_{RW} es mayor que σ_C , entonces la última muestra será la mejor estimación. En este contexto, existe

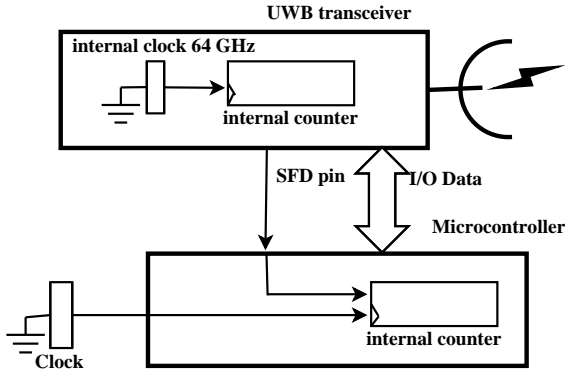


Figura 4: Detalle del transceptor DWM1000 UWB conectado al Arduino MKRZero.

un tamaño de ventana óptimo que minimizará el error para la estimación del tiempo [18].

A partir del comportamiento sintetizado del RW descrito en la Sección III, hemos evaluado experimentalmente el tamaño de ventana óptimo en función de la contribución del RW frente al ruido subyacente, midiendo ambos, en relación a σ_{RW} y σ_C , respectivamente. En la Tabla I, se muestran los tamaños de ventana óptimos para tres escenarios en los que se emplea una regresión lineal: a) σ_{RW} es mucho menor que σ_C , b) σ_{RW} es menor que σ_C , y c) σ_{RW} es similar o mayor que σ_C . Estos resultados se han determinado fijando un intervalo de confianza del 95 % para su significación estadística.

Obsérvese que cuando la contribución predominante es el ruido, entonces, como era de esperar, el tamaño óptimo de la ventana es lo más grande posible. Por contra, cuando predomina el RW, entonces el tamaño óptimo de la ventana es en la práctica la última marca de tiempo.

Cuadro I: Tamaños de ventana óptimos para regresión lineal considerando RW simultáneo con ruido, con desviaciones σ_{RW} y σ_C respectivamente.

Escenario	% de RW	Ventana óptima
$\sigma_{RW} \ll \sigma_C$	< 0,3	16 - ∞
$\sigma_{RW} < \sigma_C$	0,3 - 30	3-15
$\sigma_{RW} \geq \sigma_C$	> 30	2-3

V. PROTOTIPO DE NODO INALÁMBRICO

En la Fig. 4 se presenta un esquemático del nodo inalámbrico empleado en este trabajo, basado en el transceptor UWB Decawave DWM1000 [2]. El microcontrolador (o MCU) utilizado es Arduino MKRZero [19] de 32 bit basado en ARM@48 MHz con 32 KB de RAM y 256 KB de Flash.

Cabe destacar que configurando adecuadamente el MCU es posible conseguir un reloj con una frecuencia de hasta 96 MHz, y que se dispone de una librería específica para configurar este transceptor [20].

En la Fig. 4, se muestra como tanto el transceptor como la MCU tienen su propio reloj local, utilizando osciladores de cristal de cuarzo de bajo coste independientes. Hay que

tener en cuenta que con este transceptor, utilizando un reloj externo de 38,4 MHz y la línea de sincronización (SYNC), podemos conseguir mejorar la sincronización entre ambos, como se muestra en [15]. Sin embargo, dificultaría el diseño ya que nuestra intención es definir un nodo inalámbrico sencillo.

Este transceptor UWB tiene un contador interno de 40 bits conectado a un reloj interno de 64 GHz. Este contador se utiliza para marcar el tiempo de la trama, tanto al transmitir como al recibir. Utilizando el SFD, el transceptor emisor puede sincronizar la marca de tiempo cuando se envía el campo SFD de la trama, y el transceptor receptor puede sincronizar su marca de tiempo, justo cuando se detecta el SFD. Además, para la sincronización externa, el transceptor proporciona un pin SFD, como se muestra en la Fig. 4, activado según el mecanismo SFD, pero sólo en la recepción de tramas (no en la transmisión), permitiendo en la MCU el marcado de tiempo de las tramas cuando se detecta este campo. Por tanto en la práctica, la activación del pin SFD limitará la sincronización con la MCU. Obsérvese que como este transceptor no fue diseñado para la sincronización, su caracterización no se encuentra en los manuales y/o especificaciones

VI. RESULTADOS

En esta sección, para analizar el error de sincronización en las estimaciones de tiempo, utilizaremos el valor medio de los errores absolutos de predicción, o Mean Absolute Prediction Error (*MAPE*) como se describe en la Sección IV. Para los experimentos utilizamos el protocolo descrito en la Sección IV y colocamos el maestro a un metro de distancia de los esclavos en una habitación con una temperatura estable en torno a los 25°C. El maestro envía paquetes de sincronización con las marcas de tiempo correspondientes a los envíos de su transceptor UWB, y los esclavos, utilizando el mecanismo SFD, determinarán el tiempo de estos paquetes tanto en el transceptor UWB como en la MCU. Durante las pruebas, el maestro envió 94080 paquetes con un SP de 200 ms.

Además, para evaluar el efecto del RW en los desvíos del reloj, hemos testeado el protocolo de sincronización utilizando diferentes SP, tal y como se comenta en la sección IV. Hemos utilizado varios SP: 200 ms, 1 s, 10 s, 30 s y 60 s. En las Fig. 5-9 se muestran los *MAPE* calculados a partir de las marcas de tiempo registradas en el UWB. Además, en estas figuras hemos evaluado diferentes técnicas de regresión, utilizando aproximaciones lineales, cuadráticas, cúbicas y cuárticas, así como modificando el tamaño de la ventana aplicada en la regresión.

Como era de esperar, el menor *MAPE* viene dado por la utilización del SP más corto, con 200 ms, y utilizando la regresión lineal, como se muestra en la Fig. 5. En el eje x se da el índice del tamaño de la ventana para cada regresión. Para una regresión lineal, el primer índice comienza con un tamaño de ventana mínimo de 2 muestras. Para las aproximaciones cuadrática, cúbica y cuártica los tamaños mínimos de ventana son de 3, 4 y 5 muestras respectivamente, el grado más uno. Entonces, según Fig. 5,

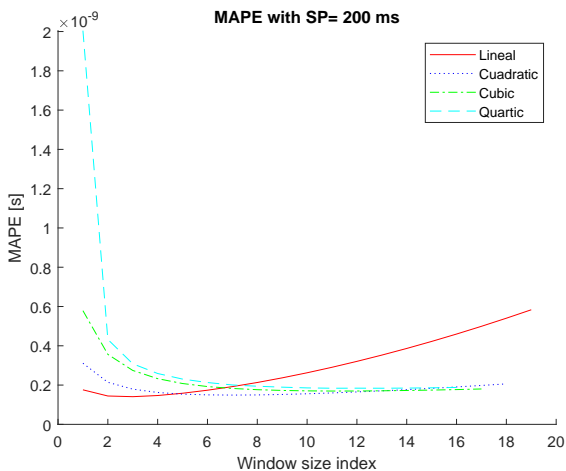


Figura 5: MAPE para un SP de 200 ms utilizando diferentes regresiones.

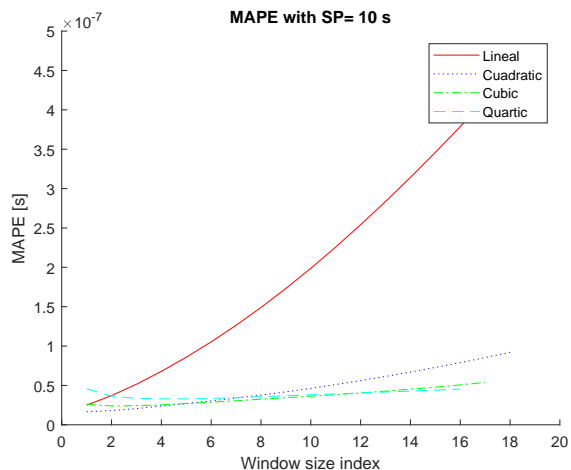


Figura 7: MAPE para un SP de 10 s utilizando diferentes regresiones.

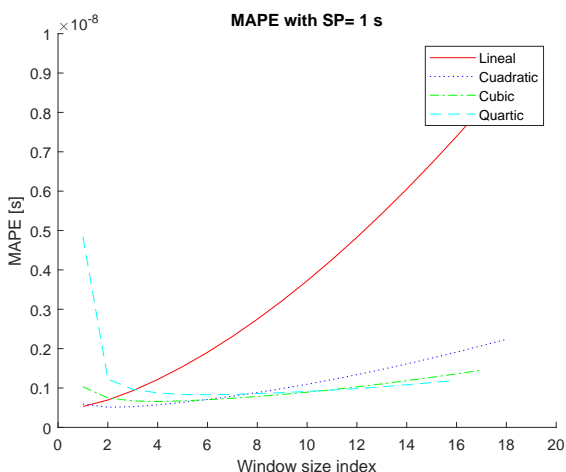


Figura 6: MAPE para un SP de 1 s utilizando diferentes regresiones.

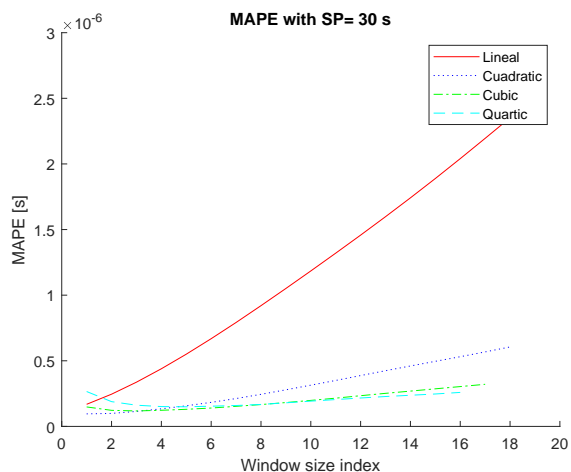


Figura 8: MAPE para un SP de 30 s utilizando diferentes regresiones.

el óptimo se consigue con un índice de tamaño de ventana de 3, es decir un tamaño de ventana de 4 muestras, con un MAPE de 149 ps.

Con un SP de 1 s, el óptimo se consigue con un tamaño de ventana en el tercer índice utilizando la regresión cuadrática, es decir, un tamaño de ventana de 4 muestras, ya que el índice de ventana óptimo es 2 y para esta regresión, el tamaño de ventana mínimo es 3, como se muestra en las Fig. 6, con un MAPE de 511,8 ps. Para SP más altos (10, 30 y 60 s), como se muestra en la Fig. 7-9, siempre se consiguen los mejores resultados con una regresión cuadrática y con el menor tamaño de ventana posible. Esto se debe a la prevalencia del efecto RW, como se ha comentado anteriormente. En la Tabla II, resumimos los valores óptimos utilizando las marcas de tiempo del UWB, indicando el SP, MAPE, grado (D) y tamaño de ventana (WS).

En las Fig. 10-11 se muestran los histogramas del error cuando se utiliza un SP de 200 ms tanto con regresión

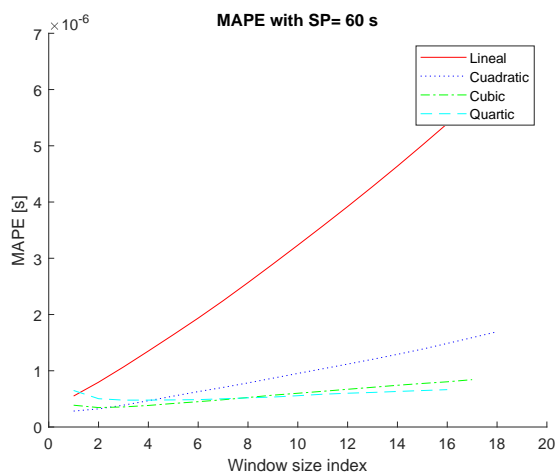


Figura 9: MAPE para un SP de 60 s utilizando diferentes regresiones.

Cuadro II: Resumen de los valores óptimos, con el valor mínimo de *MAPE* [ns], según *SP*, grado (*D*) y tamaño de ventana (*WS*) con las marcas temporales de UWB.

SP [s]	0,2	1	10	30	60
MAPE [ns]	0,141	0,512	16,56	95,5	283,4
D	1	2	2	2	2
WS	4	4	3	3	3

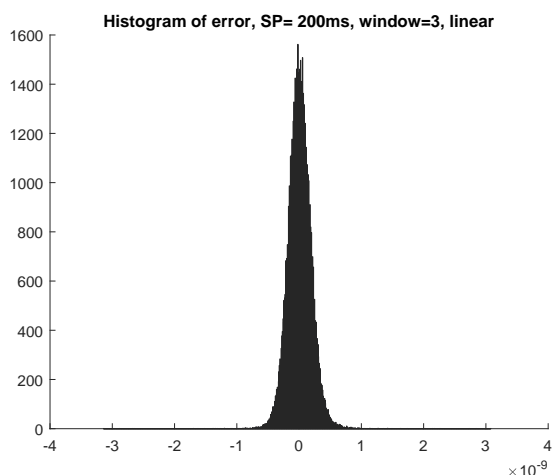


Figura 10: Histograma del error utilizando regresión lineal con un SP de 1s y una ventana de 3.

lineal como cuadrática respectivamente. En este caso, utilizando la regresión lineal la precisión aumenta en comparación con las otras técnicas de regresión.

Por último, en las Fig. 12-16 se muestran los errores basados en las marcas de tiempo de la MCU, utilizando los mismos escenarios y configuraciones. A primera vista, observamos que los errores aumentan en comparación con los anteriores, obtenidos utilizando la marca de tiempo interna en el transceptor UWB. Como hemos comentado, esto se debe a que el marcado de tiempo SFD en la MCU depende se basa en la activación de la salida del pin SFD.

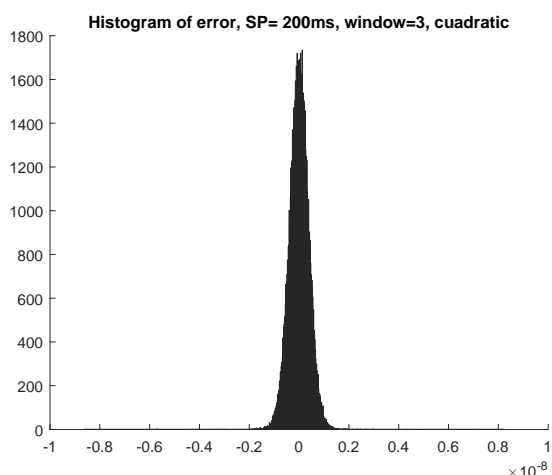


Figura 11: Histograma del error utilizando regresión cuadrática con un SP de 1s y una ventana de 3.

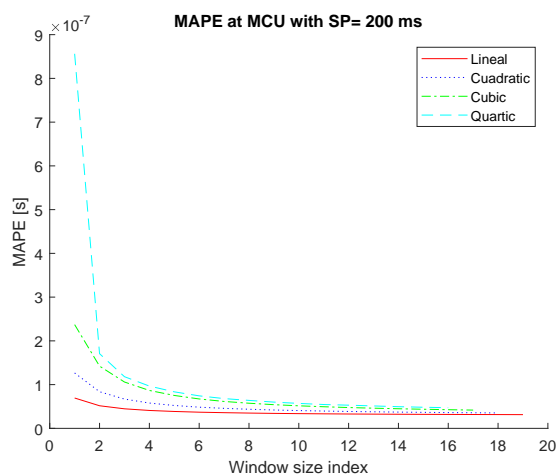


Figura 12: MAPE para un SP de 200 ms utilizando diferentes regresiones en la MCU.

Además, los contadores internos de la MCU funcionan a una velocidad inferior a la del transceptor UWB. Mientras que el transceptor UWB trabaja a 64 GHz, el reloj de la MCU trabaja a 78,698 MHz y no 96 MHz, como se explica en la Sección V, porque si utilizamos una frecuencia más alta, el MCU empieza a perder paquetes.

Cabe mencionar, que como la MCU está utilizando una frecuencia de reloj más baja en comparación con UWB, el comportamiento de RW es diferente, como podemos ver en las Fig. 12-13. En este caso, el tamaño de ventana óptimo viene dado por la mayor ventana posible del rango utilizado (20), tendiendo a infinito, con un *MAPE* de 31,35 con un SP de 0,2 s y empleando una regresión lineal y de 38,51 ns con un SP 1 s y empleando una regresión cuadrática. Por lo tanto, el efecto RW es menor en comparación con el ruido. Sin embargo, al aumentar el SP a 10, 30 y 60 s, el efecto RW aparece y el tamaño de ventana óptimo se encuentra por debajo de 5. Además, para los diferentes SP, las regresiones óptimas vienen dadas por la cuadrática, excepto con un SP de 0,2 s donde la regresión óptima es lineal. En la Tabla III, resumimos para las marcas de tiempo en la MCU los valores óptimos, indicando el SP, *MAPE*, *D* y *WS* utilizados.

Cuadro III: Resumen de los valores óptimos, con el valor mínimo de *MAPE* [ns], según *SP*, grado (*D*) y tamaño de ventana (*WS*) con las marcas temporales de MCU.

SP [s]	0,2	1	10	30	60
MAPE [ns]	31,35	38,51	101,6	337,5	761,1
D	1	2	2	2	2
WS	∞	∞	2	4	4

En la práctica, hay que tener en cuenta que si no se conociera la posición exacta de los nodos, podríamos calcularla fácilmente con el mismo transceptor, midiendo el TOF entre ellos. El TOF afectará a la precisión del tiempo como una compensación constante. En el transceptor, el TOF se puede calcular fácilmente utilizando la expresión propuesta en [21] por el equipo Decawave, basado en el

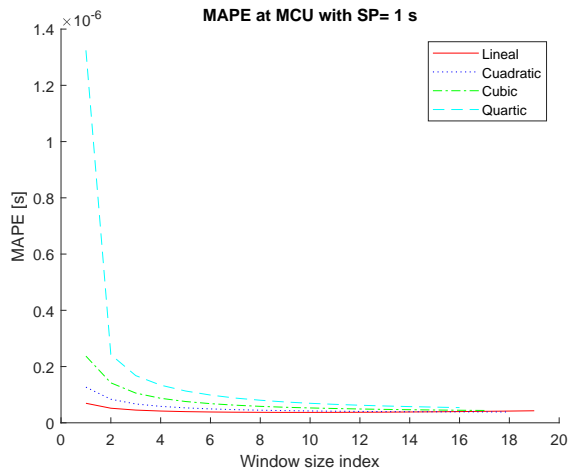


Figura 13: MAPE para un SP de 1 s utilizando diferentes regresiones en la MCU.

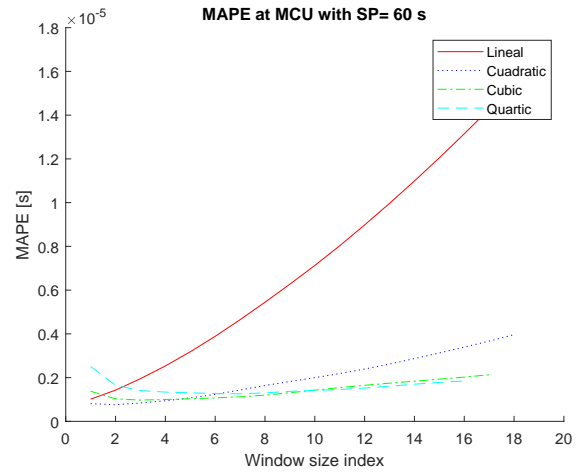


Figura 16: MAPE para un SP de 60 s utilizando diferentes regresiones en la MCU.

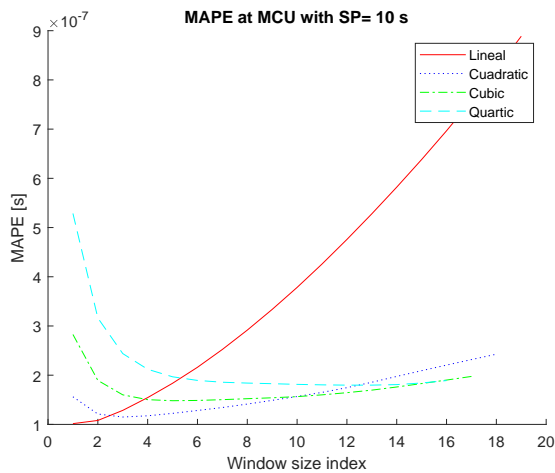


Figura 14: MAPE para un SP de 10 s utilizando diferentes regresiones en la MCU.

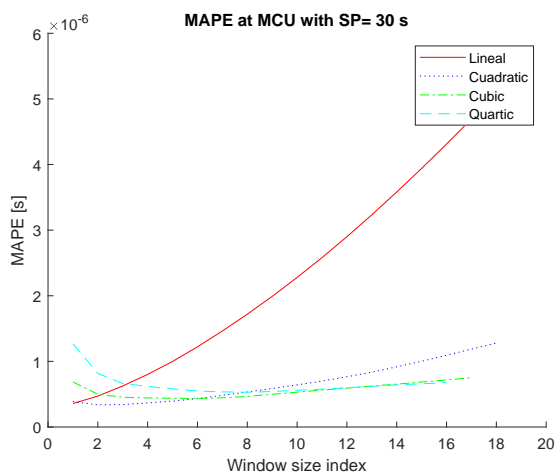


Figura 15: MAPE para un SP de 30 s utilizando diferentes regresiones en la MCU.

método *Poll-Response-Final* [22]. Este transceptor para aplicaciones de alcance tiene una inexactitud del orden de cm [2], que a la velocidad de la luz supone un desplazamiento constante de 33 ps por cm.

VII. CONCLUSIONES Y TRABAJO FUTURO

Las comunicaciones UWB pueden integrarse dentro de los nodos tradicionales de las WSN, manteniendo la característica de bajo coste de sus nodos. En este trabajo, utilizando este tipo de comunicaciones, hemos demostrado que podemos conseguir una sincronización en una escala de tiempo por debajo de los nanosegundos utilizando componentes comerciales disponibles, en particular utilizando un nodo inalámbrico sencillo basado en un transceptor UWB Decawave DWM1000 conectado a un MCU Arduino MKRZero. Cada uno de ellos con un oscilador y un sistema de reloj propios.

Hemos considerado el comportamiento de la deriva (o skew) de los relojes modelado como un proceso RW tanto entre los transceptores UWB, como entre el transceptor y la MCU. En base a ello y utilizando experimentos reales, hemos analizado el tamaño de ventana óptimo que minimiza el error de estimación para la sincronización temporal utilizando diferentes regresiones y el protocolo de sincronización *one way*. En el transceptor UWB, dado que utilizamos un reloj de 64 GHz, las derivas del RW tienen más influencia que el ruido intrínseco añadido en los intercambios de paquetes y sus marcas de tiempo comparado con la MCU. Uno de los principales inconvenientes del transceptor UWB DWM1000 es que no proporciona un pin de salida SFD cuando transmite tramas, sólo cuando recibe tramas. Debemos tener en cuenta que este transceptor fue diseñado principalmente para aplicaciones de localización y no para proporcionar una sincronización precisa entre relojes, que es el propósito de la investigación presentada en este trabajo.

En cuanto a los resultados, cuando el marcado de tiempo SFD se realiza en el propio transceiver tanto en la transmisión como en la recepción, logramos un error

de sincronización mínimo (*MAPE*) de 149 ps y 511,8 ps con un SP de 0,2 y 1 s, respectivamente. Sin embargo, cuando utilizamos las marcas de tiempo en la MCU a través del pin SFD, logramos un *MAPE* de 31,35 ns y 38,51 ns considerando SP de 0,2 s y 1 s, respectivamente.

Esto confirma que podemos explotar las características del transceptor DWM1000 para mejorar la sincronización temporal en WSN, utilizando protocolos de sincronización más complejos con el fin de conseguir una mayor precisión.

AGRADECIMIENTOS

Este trabajo ha sido posible a la financiación del Ministerio de Ciencia e Innovación con el proyecto RTI2018-098156-B-C55, la Generalitat Valenciana con el proyecto GV/2020/052 para grupos de investigación emergentes, la Conselleria de Innovacion, Universidades, Ciencia y Sociedad Digital de la Generalitat Valenciana a través de AEST/2021/16, BEST/2021/ y la Universitat de València con PIE

el Ministerio de Ciencia e Innovación con el proyecto RTI2018-098156-B-C55, la Generalitat Valenciana con el proyecto GV/2020/052 para grupos de investigación emergentes.

REFERENCIAS

- [1] E. Navarro-Camba, S. Felici-Castell, J. Segura-García, M. García-Pineda, and J. Pérez-Solano, "Feasibility of a stochastic collaborative beamforming for long range communications in wireless sensor networks," *Electronics*, vol. 7, no. 12, p. 417, Dec 2018. [Online]. Available: <http://dx.doi.org/10.3390/electronics7120417>
- [2] Decawave, "DWM1001 System Overview And Performance," <https://www.Decawave.com/content/dwm1001-system-overview-andperformance>, 2015, accessed: 26/05/2021.
- [3] J. J. Pérez-Solano, S. Ezpeleta, and J. M. Claver, "Indoor localization using time difference of arrival with uwb signals and unsynchronized devices," *Ad Hoc Networks*, vol. 99, p. 102067, 2020. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S157087051930215X>
- [4] S. Ganeriwal, I. Tsigogiannis, H. Shim, V. Tsitsis, M. Srivastava, and D. Ganesan, "Estimating Clock Uncertainty for Efficient Duty-Cycling in Sensor Networks," *Networking, IEEE/ACM Transactions on*, vol. 17, no. 3, pp. 843–856, 2009. [Online]. Available: <http://dx.doi.org/10.1109/TNET.2008.2001953>
- [5] M. Maróti, B. Kusy, G. Simon, and A. Lédeczi, "The flooding time synchronization protocol," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*, ser. SenSys '04. New York, NY, USA: ACM, 2004, pp. 39–49. [Online]. Available: <http://doi.acm.org/10.1145/1031495.1031501>
- [6] S. Yoon, C. Veerarittiphan, and M. L. Sichitiu, "Tiny-sync: Tight time synchronization for wireless sensor networks," *ACM Trans. Sen. Netw.*, vol. 3, no. 2, Jun. 2007. [Online]. Available: <http://doi.acm.org/10.1145/1240226.1240228>
- [7] T. Schmid, R. Shea, Z. Charbiwala, J. Friedman, M. B. Srivastava, and Y. H. Cho, "On the interaction of clocks, power, and synchronization in duty-cycled embedded sensor nodes," *ACM Trans. Sen. Netw.*, vol. 7, no. 3, pp. 24:1–24:19, Oct. 2010. [Online]. Available: <http://doi.acm.org/10.1145/1807048.1807053>
- [8] Z. Zhong, P. Chen, and T. He, "On-demand time synchronization with predictable accuracy," in *INFOCOM, 2011 Proceedings IEEE*, April 2011, pp. 2480–2488. [Online]. Available: <http://dx.doi.org/10.1109/INFCOM.2011.5935071>
- [9] Seiko Instruments Inc, "Product List," <http://speed.sii.co.jp/pub/compo/quartz/productListEN.jsp>, 2014, accessed: 06/02/2021.
- [10] J. J. Pérez-Solano and S. Felici-Castell, "Adaptive time window linear regression algorithm for accurate time synchronization in wireless sensor networks," *Ad Hoc Networks*, vol. 24, pp. 92–108, 2015. [Online]. Available: <http://dx.doi.org/10.1016/j.adhoc.2014.08.002>
- [11] F. Spitzer, *Principles of Random Walk*, ser. Graduate Text in Mathematics. Springer, 2001.
- [12] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, ser. SenSys '03. New York, NY, USA: ACM, 2003, pp. 138–149. [Online]. Available: <http://doi.acm.org/10.1145/958491.958508>
- [13] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, april 2005, pp. 364 – 369. [Online]. Available: <http://dx.doi.org/10.1109/IPSIN.2005.1440950>
- [14] F. Bonafini, P. Ferrari, A. Flammini, S. Rinaldi, and E. Sisinni, "Exploiting time synchronization as side effect in uwb real-time localization devices," in *2018 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS)*, 2018, pp. 1–6.
- [15] A. Dongare, P. Lazik, N. Rajagopal, and A. Rowe, "Pulsar: A wireless propagation-aware clock synchronization platform," in *2017 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2017, pp. 283–292.
- [16] S. Niranjayan and A. F. Molisch, "Ultra-wide bandwidth timing networks," in *2012 IEEE International Conference on Ultra-Wideband*, 2012, pp. 51–56.
- [17] Y. Xie, G. J. M. Janssen, and A. van der Veen, "A practical clock synchronization algorithm for uwb positioning systems," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 3891–3895.
- [18] J. J. Pérez-Solano and S. Felici-Castell, "Improving time synchronization in wireless sensor networks using bayesian inference," *Journal of Network and Computer Applications*, vol. 82, pp. 47 – 55, 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1084804517300164>
- [19] Arduino, "MKR ZERO," <https://store.arduino.cc/arduino-mkr-zero-i2s-bus-sd-for-sound-music-digital-audio-data>, 2018, accessed: 26/07/2021.
- [20] Thomas Trojer, "Github: Library that offers basic functionality to use Decawave's DW1000 chips/modules with Arduino," <https://github.com/thotro/arduino-dw1000>, 2018, accessed: 28/11/2021.
- [21] D. Neiryck, E. Luk, and M. McLaughlin, "An alternative double-sided two-way ranging method," in *2016 13th Workshop on Positioning, Navigation and Communications (WPNC)*, 2016, pp. 1–4.
- [22] Decawave, "The implementation of two-way ranging with the DW1000," https://www.decawave.com/wp-content/uploads/2018/10/APS013_The-Implementation-of-Two-Way-Ranging-with-the-DW1000_v2.3.pdf, 2018, accessed: 26/11/2021.