

Contrastando teoría y práctica en sistemas distribuidos de tiempo real con planificación EDF

J. Javier Gutiérrez^(✉), Diego García Prieto y Héctor Pérez

Grupo de Ingeniería Software y Tiempo Real
Universidad de Cantabria, 39005 - Santander, España
{gutierjj, gprietod, perezh}@unican.es

Resumen¹. El comportamiento de los planificadores EDF (*Earliest Deadline First*) ha sido ampliamente estudiado en sistemas monoprocesadores, y también hay bastantes trabajos sobre la planificación y el análisis de planificabilidad de sistemas multiprocesadores homogéneos EDF. Sin embargo, si se consideran los sistemas distribuidos, solo se dispone de un pequeño número de técnicas de análisis de planificabilidad y hay poca información sobre la experiencia práctica con este tipo de sistemas. Hay trabajos teóricos en los que se muestra que la disponibilidad de un mecanismo de sincronización de relojes permite conseguir un mayor uso de los procesadores, manteniendo el sistema distribuido planificable (es decir, cumpliendo con todos los requisitos temporales). Este trabajo propone una comparación entre los resultados teóricos que provienen del análisis de planificabilidad y los resultados medidos en una implementación práctica en lenguaje C sobre el sistema operativo de tiempo real MaRTE OS. Se ha seleccionado un experimento para mostrar las diferencias que se pueden llegar a obtener entre los resultados del análisis de planificabilidad y la ejecución real del sistema cuando se planifica con EDF.

Palabras clave: sistemas distribuidos, tiempo real, EDF, análisis de planificabilidad, sincronización de relojes, aplicaciones en C.

1 Introducción

La política de planificación *Earliest Deadline First* (EDF) está presente en diferentes capas de software como sistemas operativos de tiempo real (SHaRK [1], ERIKA [2] u OSEK / VDX [3]), redes de comunicación de tiempo real (CAN Bus [4] o redes de propósito general [5] [6]), middleware de distribución de tiempo real (RT-CORBA [7]) o lenguajes de programación de tiempo real (Java RTSJ [8] o Ada [9]). Desde sus orígenes [10], la política de planificación EDF ha sido ampliamente estudiada para sistemas monoprocesadores, tanto desde el punto de vista teórico como práctico, incluso en comparación con otras políticas como la de prioridades fijas (*Fixed Priorities*, FP) [11]

¹ Este trabajo ha sido financiado parcialmente por el Gobierno de España y los fondos FEDER (AEI/FEDER, UE) en el proyecto TIN2017-86520-C3-3-R (PRECON-14).

[12]. También hay muchos trabajos relacionados con EDF y sistemas multiprocesadores [13], en los que los planificadores EDF están bien definidos y pueden clasificarse como globales (las tareas pueden migrar de un procesador a otro, por ejemplo, [14]) o particionados (asignación fija de tareas a los procesadores, por ejemplo, [15] [16]). En [17] se realiza una evaluación de estos dos tipos de planificadores en un sistema operativo de tiempo real.

En el caso de los sistemas distribuidos de tiempo real, no existe un cuerpo de conocimiento equivalente sobre el comportamiento de la planificación EDF. En [18] se muestra cómo la utilización alcanzable de los procesadores manteniendo el sistema planificable depende en gran medida de: (1) la disponibilidad de un reloj global (es decir, un mecanismo de sincronización de relojes); y (2) la técnica de asignación de plazos de planificación utilizada cuando las tareas se componen de una secuencia de subtareas que pueden ejecutarse en diferentes procesadores, y con relaciones de precedencia entre ellas. En [18] se consideran dos tipos de políticas EDF:

- EDF con reloj local (*Local-Clock EDF*, LC-EDF), en la que los plazos de planificación de cada subtarea están referidos a los instantes de activación en el propio procesador, por lo que no se requiere la sincronización de relojes entre los procesadores.
- EDF con reloj global (*Global-Clock EDF*, GC-EDF), en la que los plazos de planificación de cada subtarea están referidos al tiempo de activación de la tarea (primera subtarea), posiblemente en un procesador diferente y, por lo tanto, requiere la sincronización de relojes.

En general, en [18] se muestra cómo desde un punto de vista teórico GC-EDF funciona mejor que LC-EDF. También se muestra cómo, en LC-EDF, la asignación de plazos de planificación a las subtareas, cuya suma no cumpliría con el plazo (como requisito temporal) asignado a la tarea, produce tiempos de respuesta de peor caso más bajos, mejorando así la capacidad de planificación. En la explotación de este fenómeno, se propusieron y probaron en [18] diferentes técnicas para la asignación de plazos de planificación para LC-EDF. En un trabajo posterior [19], se estudió si estas observaciones respecto del comportamiento de la planificación LC-EDF en la teoría, se daban también en la práctica. El resultado fue que los experimentos no corroboraban el bajo rendimiento predicho por la teoría para algunas asignaciones de plazos de planificación en LC-EDF. Además, las medidas de tiempos de respuesta promedio eran bastante similares en todas las circunstancias, lo que no permitía sacar conclusiones sobre si una técnica de asignación de plazos de planificación funciona claramente mejor que las otras.

En este artículo, proponemos avanzar en la verificación del comportamiento de la planificación EDF en sistemas distribuidos mediante el estudio teórico y práctico de los tiempos de respuesta sobre un sistema que se ha seleccionado, al que se aplicarán diferentes asignaciones de plazos de planificación tanto para LC-EDF como para GC-EDF. La ejecución de los experimentos para GC-EDF es posible por la disponibilidad de una plataforma con un mecanismo de sincronización de relojes que sigue la especificación IEEE 802.1AS [20]. Esta especificación es parte del conjunto de estándares TSN (*Time-Sensitive Networking*) [21], un esfuerzo continuo para permitir la transferencia de datos determinista en redes Ethernet.

El documento está organizado de la siguiente manera. En el apartado 2 realizamos una revisión del modelo que usamos para el sistema distribuido, así como una breve descripción de las técnicas de análisis de planificabilidad y de asignación de plazos de planificación para LC-EDF y GC-EDF que usamos en este trabajo. El sistema seleccionado como aplicación distribuida que se utilizará en nuestras pruebas se presenta en el apartado 3. En el apartado 4 se describe la plataforma utilizada en la ejecución real de las diferentes configuraciones del sistema. El apartado 5 muestra los resultados de aplicar el análisis de planificabilidad sobre el sistema para las diferentes técnicas de asignación de plazos de planificación consideradas y analiza los resultados teórico-prácticos obtenidos. Por último, en el apartado 6 se presentan las conclusiones de este trabajo.

2 Modelo de sistema y técnicas de análisis y asignación de plazos de planificación

En este apartado, se describe brevemente el modelo, las técnicas de análisis de planificabilidad y las técnicas de asignación de plazo de planificación que vamos a usar en este trabajo. Usaremos MAST (*Modeling and Analysis Suite for Real Time Applications*) [22], que integra tanto las técnicas de análisis usadas en este trabajo como el modelo [23][24], que está alineado con MARTE (*Modeling and Analysis of Real-Time Embedded systems*) [25], un estándar definido por OMG (*Object Management Group*) para el modelado y análisis de sistemas empotrados de tiempo real.

El modelo MAST considera un sistema compuesto por flujos e2e distribuidos (*end-to-end flows*) que se activan por eventos periódicos o esporádicos con un intervalo mínimo entre llegadas. Así, se puede considerar que cada flujo e2e Γ_i es activado por una secuencia periódica de eventos externos con periodo T_i , y contiene un conjunto de pasos (*steps*) que modelan las tareas y los mensajes. Cada activación periódica de un flujo e2e causa la ejecución de todos sus pasos, cada uno de ellos siendo activado cuando acaba la ejecución del precedente. En el contexto de este trabajo, asumimos una asignación estática de tareas y mensajes a procesadores y redes, y que las fases relativas de activación de los flujos e2e son arbitrarias.

La Figura 1 muestra el ejemplo del flujo e2e Γ_i con tres pasos, donde el evento e_i activa el flujo e2e con periodo T_i . Ningún paso puede activarse antes de que el precedente haya completado su ejecución. Suponemos que todos los eventos de activación son instantáneos y que cualquier actividad del sistema se modela como un paso. El paso j -ésimo de Γ_i se identifica como τ_{ij} y se caracteriza por su tiempo de ejecución de peor caso C_{ij} y su tiempo de ejecución de mejor caso C_{ij}^b . Los requisitos temporales que consideramos son los plazos e2e (*end-to-end deadlines*), D_i , medidos desde la activación del flujo e2e hasta la finalización del último paso del flujo e2e. Se permiten plazos mayores que los periodos. Como resultado de aplicar el análisis de planificabilidad, para cada paso τ_{ij} se obtiene su tiempo de respuesta de peor caso (o una cota superior del mismo) R_{ij} , y su tiempo de respuesta de mejor caso (o una cota inferior del mismo) R_{ij}^b . La estimación de los tiempos de respuesta de peor caso de los últimos pasos de los flujos e2e se puede comparar con los plazos para determinar la planificabilidad del sistema.

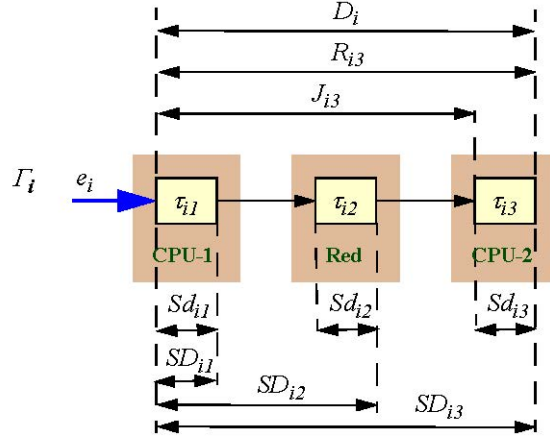


Fig. 1 Modelo del flujo $e_2e \Gamma_i$

Se permite que el evento externo que active el flujo $e_2e \Gamma_i$ tenga un *jitter* máximo de activación J_{i1} respecto de la propia activación periódica del flujo e_2e . Otros pasos τ_{ij} también podrían tener un *jitter* inicial J_{ij} . Con independencia de este *jitter*, los plazos y los tiempos de respuesta son relativos al comienzo teórico del período de la instancia concreta y no a la activación real del flujo e_2e . Permitimos que J_{ij} pueda ser mayor que el período del flujo e_2e , T_i , al que pertenece el paso. Para cada paso τ_{ij} planificado por la política EDF, se definen los plazos de planificación Sd_{ij} , para LC-EDF, y SD_{ij} , para GC-EDF, referidos respectivamente al instante de activación del propio paso en su procesador o red, o al instante de activación del flujo e_2e al que pertenece.

Este subconjunto del modelo MAST permite aplicar el análisis de tiempos de respuesta desarrollado por Spuri [26] para GC-EDF, y también el desarrollado por Rivas et al [27] para LC-EDF. Ambos análisis son equivalentes a la técnica holística desarrollada por Tindell y Clark en [28] para FP. Por otra parte, el cálculo de los tiempos de respuesta de mejor caso permite una estimación más precisa del *jitter*, por lo que se consigue una reducción de los tiempos de respuesta de peor caso calculados, mejorando la planificabilidad del sistema. De manera trivial, podemos calcular un límite inferior de estos tiempos de respuesta de mejor caso como la suma de los tiempos de ejecución de mejor caso del paso bajo análisis y los que le preceden en el flujo e_2e .

En este trabajo, vamos a utilizar cuatro métodos de asignación de plazos de planificación que evaluaremos de acuerdo con su interpretación según se use LC-EDF o GC-EDF tal como se describe en [18]:

- *Ultimate Deadline* (UD) [29]. Obtiene los plazos de planificación asignando el plazo e_2e (D_i) a todos los pasos del flujo e_2e .
- *Effective Deadline* (ED) [29]. Obtiene los plazos de planificación considerando que si un paso finaliza en el plazo asignado, los siguientes en el flujo e_2e deben completarse dentro de su tiempo de ejecución de peor caso.

Tabla 1. Características de la aplicación distribuida (tiempos en ms)

Flujo e2e	T_i	D_i	Configuración 1				Configuración 2			
			C_{i1}	C_{i2}	C_{i3}	C_{i4}	C_{i1}	C_{i2}	C_{i3}	C_{i4}
Γ_1	50	150	12	7	6	10	15	10	8	11
Γ_2	120	360	5	12	9	5	9	14	13	8
Γ_3	300	900	22	6	19	36	29	9	26	46
Γ_4	650	1950	14	41	83	26	21	59	98	38

- *Proportional Deadline* (PD) [29]. Calcula los plazos de planificación distribuyendo el plazo e2e (D_i) proporcionalmente al tiempo de ejecución de peor caso de cada paso (C_{ij}).
- *Proportional Deadline with Global Scheduling Deadline* (PD-GSD) [18]. Transforma los plazos obtenidos por PD como si estuvieran referidos al evento externo que activa el flujo. Así, el plazo de planificación del paso τ_{ij} será la suma de los plazos de planificación, asignados por PD, de los pasos precedentes en el flujo e2e, incluido él mismo.

Más detalles sobre la descripción de estos métodos, así como de las capacidades de planificación de cada uno de ellos, se puede encontrar en [18].

3 Aplicación distribuida

En este trabajo, consideramos una aplicación sintética distribuida que consta de dos procesadores (CPU1 y CPU2) conectados a través de una red de comunicaciones y ejecutando cuatro flujos e2e con la misma estructura: cuatro tareas que van ejecutándose alternativamente en cada procesador; la primera se activa periódicamente y las siguientes por la llegada del correspondiente mensaje a través de la red. Sin pérdida de generalidad, vamos a considerar los mensajes como eventos instantáneos y no los modelaremos. El efecto de estos mensajes no influirá en los resultados que se quieren mostrar, ya que el tiempo de envío de un mensaje es tres órdenes de magnitud inferior al menor de los periodos propuestos. Por el mismo motivo, también consideraremos despreciables los efectos de planificación del sistema operativo y del mecanismo de sincronización de relojes que están presentes en la implementación práctica. La Figura 2 muestra la arquitectura de la aplicación.

Para esta aplicación, proponemos estudiar dos configuraciones cuyas características se muestran en la Tabla 1. Ambas tienen los mismos periodos de activación de los flujos e2e y también los mismos plazos que se han fijado como tres veces el periodo del flujo correspondiente. Los tiempos de ejecución han sido seleccionados para mostrar las diferencias entre los tiempos de respuesta obtenidos por el análisis cuando se utilizan los diferentes métodos de asignación de plazos de planificación. Los tiempos de ejecución de la Tabla 1 son fijos, es decir, los tiempos de ejecución de mejor caso de las tareas son iguales a sus tiempos de ejecución de peor caso. Las utilizaciones de las CPUs son: 74.14% para la CPU1 y 74.59% para la CPU2 en la Configuración 1, y 96.92% for CPU1 and 96.97% para la CPU2 en la Configuración 2.

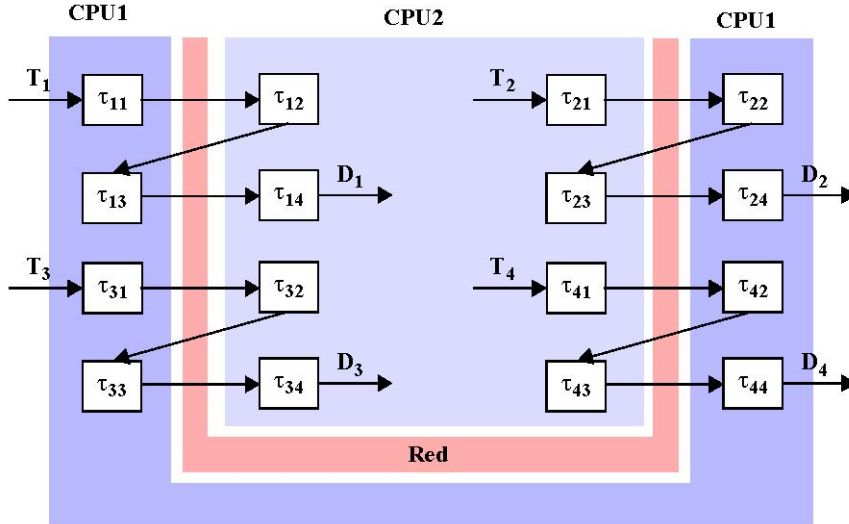


Fig. 2 Arquitectura de la aplicación distribuida

4 Plataforma distribuida con sincronización de relojes

La plataforma distribuida utilizada es la presentada en [30] y cuya arquitectura se muestra en la Figura 3. Los procesadores disponen de varios núcleos y están conectados a través de una red Ethernet que dispone del mecanismo de sincronización de relojes IEEE 802.1AS [20]. Como sistema operativo principal se utiliza Linux, que proporciona acceso a la red y ejecuta el software que permite la sincronización de relojes entre diferentes procesadores. Las aplicaciones de tiempo real ejecutan sobre el sistema operativo MaRTE OS [31][32], que se configura seleccionando la arquitectura *linux_lib*. Con esta arquitectura, una aplicación MaRTE OS se puede ejecutar como un proceso estándar Linux, pero la concurrencia y la planificación de las tareas de la aplicación la proporciona MaRTE OS. Dado que una aplicación en MaRTE OS se ejecuta como un proceso de Linux, ésta se puede ver interrumpida por las actividades del núcleo de Linux. Para minimizar este efecto y mejorar el determinismo de las aplicaciones en MaRTE OS, se aplican las utilidades de Linux que permiten aislar núcleos del procesador. Así, el objetivo es tener la aplicación de MaRTE OS ejecutando en un núcleo aislado de Linux. MaRTE OS accede al reloj global a través de su reloj REALTIME de manera convencional (más detalles en [30]).

Por su parte, MaRTE OS proporciona los servicios de planificación que permiten ejecutar tareas con planificación tanto FP como EDF. La política que implementa para EDF es definida como *EDF_Across_Priorities* en el anexo de sistemas de tiempo real del lenguaje de programación Ada [9]. Esta política permite usar tanto LC-EDF como GC-EDF teniendo en cuenta lo siguiente:

- El instante de activación del flujo e2e en cada instancia debe ser registrado por la primera tarea y enviado en todos los mensajes que forman parte del mismo.

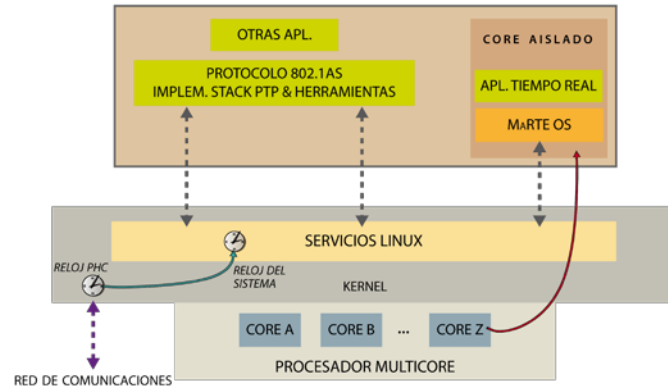


Fig. 3 Arquitectura de la plataforma distribuida

Por un lado, la disponibilidad de un reloj global permite medir tiempos de respuesta de manera sencilla en todas las tareas del flujo e2e con independencia del procesador en el que ejecutan; por otro, esta marca temporal es esencial para la planificación GC-EDF, que toma como referencia el instante en el que se activó el flujo e2e, de manera que tras la recepción de un mensaje, la tarea puede establecer su nuevo plazo de planificación relativo a ese instante.

- Para el caso en el que se use planificación LC-EDF, es necesario registrar los instantes de activación de cada tarea de un flujo e2e después de la primera. Estos instantes coinciden con la llegada del mensaje que activa cada tarea, por lo que las comunicaciones se deben modificar para que con cada mensaje se entregue también el instante en el que ha sido recibido. De este modo, tras la recepción del mensaje la tarea puede establecer su nuevo plazo de planificación relativo a ese instante que puede ser medido con un reloj local.

El hardware que utilizamos consta de dos procesadores x86_64 con cuatro núcleos a 1.9 GHz conectados a través de una red Ethernet de 1 Gbps con un *switch* compatible con el estándar IEEE 802.1AS y ejecutando el protocolo de sincronización de relojes. Así, la arquitectura mostrada en la Figura 3 estaría configurada con: (1) Linux kernel v4.4.256-rt214 configurado para permitir el aislamiento de núcleos, (2) MaRTE OS v2017 usando la arquitectura *linux_lib*, (3) ptp4l v3.0 para implementar la pila del protocolo PTP (*Precision Time Protocol*), y (4) phc2sys v3.0 para sincronizar los relojes de sistema y PHC (*PTP Hardware Clock*).

5 Análisis de tiempos de respuesta y medidas de tiempos de ejecución

El objetivo de este apartado es contrastar los resultados de la aplicación de las técnicas holísticas de análisis de tiempos de respuesta para EDF, con la implementación práctica de la aplicación presentada en el apartado 3, usando la plataforma descrita en el apartado 4. En principio, PD es el único método de asignación de plazos de planificación que tendría sentido aplicar a LC-EDF, mientras que los otros tres (UD, ED y PD-GSD),

Tabla 2. Plazos de planificación asignados por cada uno de los métodos (tiempos en ms)

Tarea	Configuración 1				Configuración 2			
	UD	ED	PD	PD-GSD	UD	ED	PD	PD-GSD
τ_{11}	150.000	127.000	51.429	51.429	150.000	121.000	51.136	51.136
τ_{12}	150.000	134.000	30.000	81.429	150.000	131.000	34.091	85.227
τ_{13}	150.000	140.000	25.714	107.143	150.000	139.000	27.273	112.500
τ_{14}	150.000	150.000	42.857	150.000	150.000	150.000	37.500	150.000
τ_{21}	360.000	334.000	58.065	58.065	360.000	325.000	73.636	73.636
τ_{22}	360.000	346.000	139.355	197.419	360.000	339.000	114.545	188.182
τ_{23}	360.000	355.000	104.516	301.935	360.000	352.000	106.364	294.545
τ_{24}	360.000	360.000	58.065	360.000	360.000	360.000	65.455	360.000
τ_{31}	900.000	839.000	238.554	238.554	900.000	819.000	225.000	225.000
τ_{32}	900.000	845.000	65.060	303.614	900.000	828.000	75.000	300.000
τ_{33}	900.000	864.000	206.024	509.639	900.000	854.000	216.667	516.667
τ_{34}	900.000	900.000	390.361	900.000	900.000	900.000	383.333	900.000
τ_{41}	1950.000	1800.000	166.463	166.463	1950.000	1755.000	189.583	189.583
τ_{42}	1950.000	1841.000	487.500	653.963	1950.000	1814.000	532.639	722.220
τ_{43}	1950.000	1924.000	986.890	1640.850	1950.000	1912.000	884.722	1606.940
τ_{44}	1950.000	1950.000	309.146	1950.000	1950.000	1950.000	343.056	1950.000

presentados en el apartado 2, son los que tendrían sentido para GC-EDF. Sin embargo, en función de los resultados mostrados en [18] y [19], vamos a evaluar los cuatro métodos de asignación de plazos de planificación tanto para LC-EDF como para GC-EDF.

Tras modelar la aplicación con MAST, se aplican los métodos de asignación de plazos de planificación descritos en el apartado 2. La Tabla 2 muestra el resultado de esta asignación para los cuatro métodos, las dos configuraciones y todas las tareas (pasos). Se puede observar que UD realiza la misma asignación para ambas configuraciones, y PD es la única técnica en la que la suma de los plazos de planificación de las tareas perteneciente a un flujo e2e coincide con el plazo e2e.

La Tabla 3 muestra los resultados de aplicar el análisis de planificabilidad holístico [26][27] a los casos mostrados en la Tabla 2, considerando que se usa tanto LC-EDF como GC-EDF. Sólo se muestran los tiempos de respuesta de peor caso de la última tarea de cada flujo e2e, que son los que se puede comparar con los plazos e2e. El caso de asignación PD no se ha estudiado para GC-EDF porque en principio carece de sentido. A partir de estos resultados analíticos podemos hacer las siguientes observaciones:

- Los tiempos de respuesta de peor caso obtenidos dependen fuertemente de asignación de plazos de planificación. Así, para la configuración con mayor carga (Configuración 2), las asignaciones UD, ED y PD para LC-EDF no convergen (los tiempos se van a infinito); sin embargo, UD y ED sí que convergen para GC-EDF, aunque con tiempos muy altos.
- Para la Configuración 1, la menos cargada, se puede decir que los resultados que se obtienen con LC-EDF y GC-EDF son muy similares salvo para UD, siendo la asignación PD-GSD la que obtiene los menores tiempos de respuesta.
- Para la Configuración 2, los mejores resultados también se obtienen con la asignación PD-GSD, aunque en este caso el uso de planificación con reloj global

Tabla 3. Tiempos de respuesta de peor caso de los flujos e2e obtenidos por el análisis de planificabilidad (tiempos en ms)

Flujo e2e	Configuración 1 – LC-EDF				Configuración 2 – LC-EDF			
	UD	ED	PD	PD-GSD	UD	ED	PD	PD-GSD
Γ_1	70.000	54.000	48.000	64.000	No conv.	No conv.	No conv.	87.000
Γ_2	164.000	118.000	134.208	176.321	No conv.	No conv.	No conv.	326.505
Γ_3	609.000	356.000	433.286	400.000	No conv.	No conv.	No conv.	814.000
Γ_4	2392.000	1466.000	1025.100	893.000	No conv.	No conv.	No conv.	2786.940
	Configuración 1 – GC-EDF				Configuración 2 – GC-EDF			
	UD	ED	PD	PD-GSD	UD	ED	PD	PD-GSD
Γ_1	51.000	64.000	-	64.000	1114.000	884.000	-	88.318
Γ_2	121.000	141.000	-	178.000	1352.000	1084.000	-	316.818
Γ_3	530.000	481.000	-	391.000	1923.000	1663.000	-	761.485
Γ_4	1508.000	1457.000	-	893.000	3037.000	2793.000	-	2048.820

GC-EDF es mejor: reduce bastante los tiempos de respuesta de tres flujos e2e a costa de aumentar ligeramente el otro.

Estos resultados están en la línea de los estudios previos [18][19] en los que GC-EDF obtiene mayor rendimiento, aunque en esta aplicación no se aprecia un dominio tan claro sobre LC-EDF.

Para comparar estos resultados teóricos con la ejecución real de la aplicación, se ha realizado una implementación de la misma en C para ejecutarla sobre MaRTE OS en la plataforma descrita en el apartado 4. Se han ejecutado las dos configuraciones con las asignaciones de plazos de planificación posibles durante una hora, y se han medido los tiempos de ejecución peores, mejores y promedio de cada flujo e2e. Los mensajes intercambiados por las tareas tenían una longitud de 64 bytes. Teniendo en cuenta estos resultados y los de la Tabla 3, en las gráficas de la Figura 4 se representan valores de tiempos de respuesta de los flujos e2e relativos al plazo e2e del flujo en porcentaje (un valor por debajo de 100 indica que el tiempo de respuesta es menor que el plazo). Los valores que se representan se refieren a los tiempos de respuesta de peor caso medidos (W_{crt}), los teóricos que provienen del análisis (W_{crt-T} , no se representan los casos que no convergen en la Configuración 2) y los tiempos de respuesta promedio; todo ello para los cuatro flujos e2e, las dos configuraciones y las diferentes asignaciones de plazos de planificación para LC-EDF y GC-EDF. Para la Configuración 2 se usa una escala logarítmica con objeto de apreciar mejor las diferencias, ya que algunos resultados teóricos superan en mucho los plazos e2e. Teniendo en cuenta estos resultados, podemos hacer las siguientes observaciones:

- Las asignaciones que no convergían en el análisis de la Configuración 2 para LC-EDF obtienen en la práctica tiempos de peor caso muy similares e incluso mejores en algunos flujos e2e que las asignaciones que sí convergían. En cualquier caso, para esta configuración con mayor carga las asignaciones que mejor van son PD y PD-GSD. La asignación ED para GC-EDF es especialmente mala en la práctica.
- En la Configuración 1 los resultados prácticos también son muy similares para todas las asignaciones, apreciándose pequeñas diferencias entre ellas, de manera

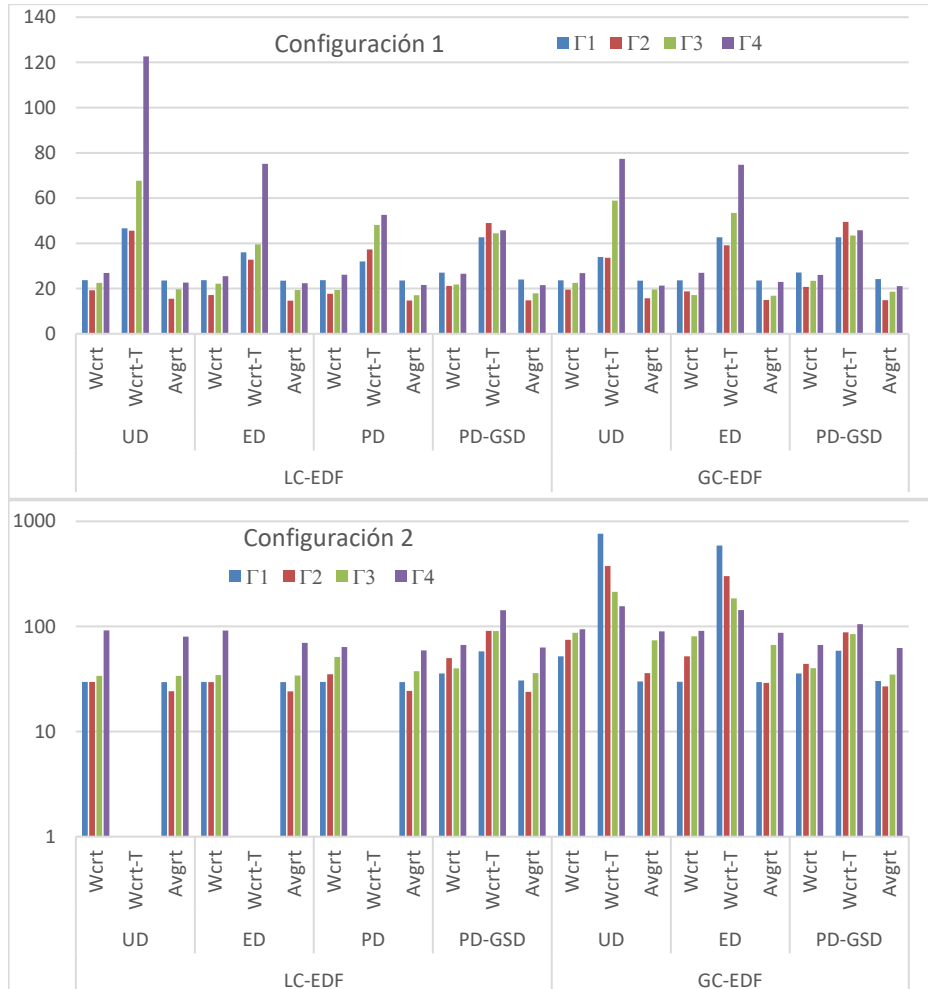


Fig. 4 Tiempos de respuesta de los flujos e2e relativos a los plazos e2e (en %)

que, si un flujo e2e va un poco mejor, otros empeoran ligeramente. En esta configuración los resultados teóricos se acercan más a los prácticos para la asignación PD-GSD por un lado, y para los casos planificados con GC-EDF.

- Los tiempos promedio medidos siguen el mismo comportamiento que los de peor caso. Las asignaciones que producen tiempos de respuesta de peor caso menores también tienen menores tiempos promedio.

6 Conclusiones

En [19] se mostraba cómo para la planificación LC-EDF, PD-GSD era la técnica más precisa y la única que permitía cumplir los plazos en la teoría y en la práctica.

Siguiendo esta línea, este trabajo ha buscado comprobar si el dominio que la planificación GC-EDF parece tener sobre LC-EDF en la teoría, se da también en la práctica.

Aunque en las pruebas realizadas es posible que no se hayan encontrado los tiempos de respuesta de peor caso, las medidas realizadas sí que muestran la tendencia del comportamiento de las diferentes asignaciones de prioridad y se puede concluir: (1) que ese dominio teórico no se da en la práctica, (2) la precisión de la técnica de análisis es muy dependiente de la asignación de plazos de planificación, y (3) salvo para sistemas sencillos, el comportamiento teórico-práctico de la planificación EDF en sistemas distribuidos es muy difícil de comprender y requiere más estudio.

Bibliografía

1. S.Ha.R.K. (Soft Hard Real-Time Kernel) home page, <http://shark.sssup.it/>
2. ERIKA Enterprise, Evidence home page, <http://www.evidence.eu.com/>
3. Diederichs, C., Margull, U., Slomka, F., Wurrer, G.: An application-based EDF scheduler for OSEK/VDX. In: Design, Automation and Test in Europe, DATE '08, pp. 1045-1050 (2008)
4. Pedreiras, P., Almeida, L.: EDF message scheduling on controller area network. *Computing & Control Engineering Journal* 13(4), 163-170 (2002)
5. Di Natale, M., Meschi, A.: Scheduling Messages with Earliest Deadline Techniques. *Real-Time Systems* 20(3), 255-285 (2001)
6. Qian, T., Mueller, F., Xin, Y.: Hybrid EDF Packet Scheduling for Real-Time Distributed Systems. In: Proc. of the 27th Euromicro Conference on Real-Time Systems, Lund (Sweden), 37-46 (2015)
7. OMG (Object Management Group), Realtime Corba Specification. v1.2, <http://www.omg.org/spec/RT/1.2/> (2005)
8. RTSJ (Real-Time Specification for Java) home page, <http://www.rtsj.org>
9. ISO/IEC, 2012. Ada 2012 Reference Manual. Language and Standard Libraries - International Standard ISO/IEC 8652:2012(E) (2012)
10. Liu, C.L., Layland, J.W.: Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment. *Journal of the ACM* 20 (1), 46-61 (1973)
11. Buttazzo, G.: Rate Monotonic vs. EDF: Judgment Day. *Real-Time Systems* 29(1), 5-26 (2005)
12. Davis, R.I., Burns, A., Baruah, S., Rothvoß, T., George, L., Gettings, O.: Exact comparison of fixed priority and EDF scheduling based on speedup factors for both pre-emptive and non-pre-emptive paradigms. *Real-Time Systems* 51 (5), 561-601 (2015)
13. Davis, R.I., Burns, A.: A Survey of Hard Real-Time Scheduling for Multiprocessor Systems. *ACM Computing Surveys* 43(4), Article 35 (2011)
14. Bertogna, M., Cirinei, M., Lipari, G.: Schedulability analysis of global scheduling algorithms on multiprocessor platforms. *IEEE Transactions on Parallel and Distributed Systems* 20(4), 553-566 (2009)
15. Baruah, S., Fisher, N.: Non-migratory feasibility and migratory schedulability analysis of multiprocessor real-time systems. *Real-Time Systems* 39(1-3), 97-122 (2008)
16. Baruah, S.: Partitioned EDF scheduling: a closer look. *Real-Time Systems* 49 (6), 715-729 (2013)
17. Gracioli, G., Fröhlich, A.A., Pellizzoni, R., Fischmeister, S.: Implementation and evaluation of global and partitioned scheduling in a real-time OS. *Real-Time Systems* 49(6), 669-714 (2013)

18. Rivas, J.M., Gutiérrez, J.J., Palencia, J.C., González Harbour, M.: Deadline Assignment in EDF Schedulers for Real-Time Distributed Systems. *IEEE Transactions on Parallel and Distributed Systems* 26(10), 2671-2684 (2015)
19. Gutiérrez, J.J., and Pérez, H.: Theory and Practice of EDF Scheduling in Distributed Real-Time Systems. *Proc. of the 23rd International Conference on Reliable Software Technologies, Ada-Europe 2018, Lisbon (Portugal)*, in *Lecture Notes in Computer Science, LNCS Vol. 10873*, 123-137 (2018)
20. IEEE 802.1AS - "Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks" (2020)
21. IEEE 802.1 Time-Sensitive Networking (TSN) Task Group. Available online: <https://1.ieee802.org/tsn/>
22. MAST home page: <http://mast.unican.es/>
23. González Harbour, M., Gutiérrez, J.J., Palencia, J.C., Drake, P.: MAST: Modeling and Analysis Suite for Real Time Applications. *Proc. of the 13th Euromicro Conference on Real-Time Systems, Delft, The Netherlands*, IEEE Computer Society Press, 125-134 (2001)
24. González Harbour, M., Gutiérrez, J.J., Drake, J.M., López, P., Palencia, J.C.: Modeling distributed real-time systems with MAST 2. *Journal of Systems Architecture* 59(6), 331-340 (2013)
25. Object Management Group. UML Profile for MARTE: Modeling and Analysis of Real-Time Embedded systems. *OMG Document*, v1.2 formal/19-04-01 (2019)
26. Spuri, M.: Holistic Analysis for Deadline Scheduled Real-Time Distributed Systems. *Tech. Rep. RR-2873, INRIA, France* (1996)
27. Rivas, J.M., Gutiérrez, J.J., Palencia, J.C., González Harbour, M.: Optimized Deadline Assignment and Schedulability Analysis for Distributed Real-Time Systems with Local EDF Scheduling. In: *Proc. of the 8th International Conference on Embedded Systems and Applications, ESA'2010, Las Vegas (Nevada), USA*, 150-156 (2010)
28. Tindell, K.W., Clark, J.: Holistic Schedulability Analysis for Distributed Hard Real-Time Systems. *Microprocessing and Microprogramming*, vol. 40, no. 2-3, 117-134 (1994)
29. Liu, J.: *Real-Time Systems*. Prentice Hall, (2000)
30. Pérez Tijero, H., García Prieto, D., Gutiérrez, J.J.: First steps towards an IEEE 802.1AS clock for EDF scheduling in distributed real-time systems. *WiP of the 25th Ada-Europe International Conference on Reliable Software Technologies* (2021)
31. MaRTE OS home page: <http://marte.unican.es>
32. Aldea, M., González Harbour, M.: MaRTE OS: An Ada Kernel for Real-Time Embedded Applications. In: *Proc. of the International Conference on Reliable Software Technologies, Ada-Europe 2001, Leuven (Belgium)*, in *Lecture Notes in Computer Science, LNCS 2043*, 305-316 (2001)