

Universidad de Cantabria
Departamento de Ingeniería
Informática y Electrónica



Análisis de planificabilidad basado
en offsets en sistemas distribuidos
de tiempo real con prioridades
dinámicas

Tesis doctoral

Unai Díaz de Cerio Urain

Santander, Mayo 2015

Universidad de Cantabria
Departamento de Ingeniería
Informática y Electrónica



Análisis de planificabilidad basado
en offsets en sistemas distribuidos
de tiempo real con prioridades
dinámicas

Memoria

presentada para optar al grado de

DOCTOR por la Universidad de

Cantabria por

Unai Díaz de Cerio Urain

Ingeniero en Informática

Universidad de Cantabria
Departamento de Ingeniería Informática y Electrónica

Análisis de planificabilidad basado en offsets en sistemas distribuidos de tiempo real con prioridades dinámicas

Memoria

presentada para optar al grado de Doctor por la Universidad de Cantabria, dentro del programa de doctorado Ciencias, Tecnología y Computación, por el Ingeniero en Informática

Unai Díaz de Cerio Urain

El Director,

Dr. Michael González Harbour
Catedrático de Universidad

Declaro:

Que el presente trabajo ha sido realizado en el Departamento de Ingeniería Informática y Electrónica de la universidad de Cantabria, bajo mi dirección y reúne las condiciones exigidas para la defensa de tesis doctorales.

Santander, Mayo 2015

Fdo. Unai Díaz de Cerio Urain

Fdo. Michael González Harbour

Agradecimientos

Con estas pocas palabras quiero agradecer a todos los que han ayudado a que este trabajo llegue a buen puerto.

Gracias a Ikerlan por la oportunidad que me ha dado para realizar esta tesis, y en especial a Juanpe por su confianza y apoyo durante estos años.

Gracias a Michael por toda la ayuda que me ha dado para poder llevar a cabo esta tesis. Su irrupción en la dirección de la tesis supuso un bote salvavidas en un barco que se hundía rápidamente.

Gracias a Carlos por su ayuda para descifrar las formulas interminables con las que me he tenido que pelear.

Gracias a todos los compañeros de Ikerlan por los buenos momentos que hemos pasado. En especial a Aitor porque ha sido un gran apoyo moral.

Gracias también a los compañeros del Grupo de Ingeniería Software y Tiempo Real por hacerme tan agradables mis estancias en Santander. Todavía echo de menos las comidas de los viernes y los partidos de futbol sala.

Y por último, pero para nada menos importante, gracias a mi familia y amigos y en especial a mis padres que son los que más apoyo me han dado y más han soportado los malos momentos de la tesis.

Resumen

Palabras clave Tiempo-Real, Sistemas Distribuidos, Análisis de Planificabilidad, Análisis de Tiempo de Respuesta, Prioridades Dinámicas, EDF.

La tesis doctoral presentada en esta memoria está dedicada al análisis de planificabilidad en sistemas de tiempo real. Este análisis se realiza mediante técnicas de cálculo de tiempo de respuesta de peor caso. El trabajo está centrado en sistemas multiprocesador o distribuidos gobernados por eventos.

El objetivo principal de la tesis es potenciar el uso de planificadores por prioridades dinámicas, concretamente planificadores EDF (Earliest Deadline First, Plazo más Cercano Primero). Habiéndose demostrado que la planificación EDF permite alcanzar cotas de utilización superiores a las de las técnicas de prioridades fijas en sistemas monoprocesador, el estudio y desarrollo de técnicas de análisis en sistemas distribuidos o multiprocesador es claramente inferior a los de la planificación por prioridades fijas. Para este tipo de sistemas las técnicas de análisis exactas son muy costosas y, por ello, las técnicas de análisis usuales son aproximadas. En esta tesis, se pretende mejorar las técnicas de análisis para planificación EDF reduciendo así el pesimismo de las técnicas actuales.

Para conseguir este objetivo, en primer lugar se valida el método utilizado previamente y, posteriormente, se aplican mejoras sobre este para reducir el pesimismo del cálculo. Estas mejoras se obtienen adaptando métodos desarrollados para prioridades fijas a sistemas EDF. En concreto, los métodos utilizados para dicha adaptación son el modelo basado en intervalos de desfase (*offsets*) y la técnica que explota las relaciones de precedencia entre las tareas de un mismo flujo de tareas. El análisis de los desfases entre activaciones de las tareas (*offsets*) y de las relaciones de precedencia entre las tareas de un mismo flujo de tareas permiten reducir el pesimismo del análisis y con ello aumentar la utilización del sistema.

Por último, se introduce una modificación al modelo de flujo de tareas o transacciones lineales (*end-to-end flows*) que permite ampliar el tipo de aplicaciones representables por el modelo de tareas. Esta modificación consiste en introducir secciones atómicas de tareas que permiten la representación de threads que se autosuspenden en una operación de espera o llamadas bloqueantes (típicamente llamadas bloqueantes a procedimientos remotos). Se han adaptado las técnicas de análisis EDF para este nuevo modelo.

Abstract

Keywords Real-Time, Distributed Systems, Schedulability Analysis, Response Time Analysis, Dynamic Priorities, EDF.

The thesis presented in this report is dedicated to schedulability analysis in real-time systems. This analysis is carried out through worst case response time techniques and the targets are event-driven multiprocessor and distributed systems.

The main purpose of this thesis is to enhance the use of dynamic priorities schedulers, in particular those using EDF (Earliest Deadline First) scheduling algorithm, for distributed systems. Although it has been demonstrated that EDF scheduling can reach higher levels of utilization than fixed-priority scheduling techniques in uniprocessor systems, studies and techniques for distributed or multiprocessor systems in EDF than in fixed-priority scheduling. In these systems exact analysis techniques are too expensive and, thus, the usual techniques are approximate. Thus, the aim of this thesis is to improve analysis techniques for EDF scheduled systems decreasing the pessimism of the current techniques.

To achieve this aim, the first step is to validate previously used methods. Then, some improvements have been implemented to reduce the pessimism of the calculation. These improvements have been achieved by adapting to EDF methods developed for fixed priorities. In particular, the methods used are the offset-based model and the technique that exploits precedence relations between tasks of the same end-to-end flow. The study of the offsets between releases of the different tasks and of the precedence relations between tasks of the same end-to-end flow allows reducing the pessimism of the analysis and, consequently, to increasing the overall utilization of the system.

Finally, a modification in the end-to-end flow model has been introduced to represent atomic task sections. This modification allows us to expand

the type of real-time applications representable by the task model. This new model allow us to represent threads that suspend themselves on delay operations or blocking calls (usually remote procedure calls). The EDF analysis techniques have been extended to cover this new model.

Índice

Agradecimientos	VII
Resumen	IX
Abstract	XI
Lista de acrónimos	XXIV
1. Sistemas distribuidos de tiempo real	1
1.1. Sistemas de tiempo real	1
1.2. Modelo del sistema	4
1.3. Sistemas distribuidos y multiprocesador	8
1.3.1. Redes de comunicación	9
1.3.2. Modelo de flujo de tareas con <i>offsets</i>	11
1.4. Planificación de sistemas de tiempo real	12
1.4.1. Planificación estática	14
1.4.2. Planificación dinámica	15

1.5. Exclusión mutua	19
1.6. Problemática en sistemas distribuidos o multiprocesador	22
1.7. Objetivos	25
1.8. Organización de la memoria	28
2. Técnicas de análisis temporal	31
2.1. Definición de técnicas de análisis	31
2.2. Sistemas monoprocesadores	32
2.2.1. Tareas con recursos compartidos	35
2.2.2. Plazos arbitrarios	38
2.2.3. Plazos arbitrarios para planificaciones EDF	40
2.3. Sistemas multiprocesadores y distribuidos	42
2.3.1. Método holístico para FP (<i>Fixed Priorities</i> , Prioridades Fijas)	42
2.3.2. Método holístico para EDF	44
2.3.3. Método con <i>offsets</i> dinámicos para FP	45
2.3.4. Método con <i>offsets</i> dinámicos para EDF	47
2.3.5. Método con <i>offsets</i> dinámicos y relaciones de precedencia para FP	50
2.3.6. Método <i>slanted</i> con <i>offsets</i> dinámicos para FP	59
3. Demostración de la convergencia del análisis holístico en sistemas distribuidos con planificación EDF	61
3.1. Extensión al algoritmo de Spuri	62

3.2. Posible problema de no convergencia	62
3.3. Demostración de la convergencia del análisis	63
3.3.1. Sistema sin recursos compartidos	64
3.3.2. Sistema con recursos compartidos	86
3.3.3. Sistema con Periodos Finitos	102
3.4. Aplicación en un ejemplo industrial	104
3.5. Conclusiones	107
4. Relaciones de precedencia en el análisis de planificabilidad para sistemas distribuidos de tiempo real con planificación EDF	109
4.1. Aplicabilidad de las relaciones de precedencia a la planificación EDF	110
4.2. Técnica basada en offsets para sistemas LC-EDF	114
4.3. Relaciones de precedencia en sistemas LC-EDF	121
4.3.1. Interferencia de peor caso de un flujo de tareas	122
4.3.2. Interferencia del flujo de tareas bajo análisis	126
4.4. Resultados de simulación	131
4.5. Conclusiones	136
5. Análisis de planificabilidad de flujos de tareas con secciones atómicas	139
5.1. Modelo del sistema con flujo de tareas con secciones atómicas	140
5.2. Alternativas al modelo lineal	143

5.3. Análisis de planificabilidad	144
5.3.1. Calculando <i>offset</i> y <i>jitter</i> de las tareas	144
5.3.2. Interferencia de peor caso	149
5.3.3. Aplicación al flujo de tareas bajo análisis	154
5.4. Evaluación del análisis con secciones atómicas	157
5.5. Conclusiones	159
6. Conclusiones y trabajo futuro	161
6.1. Revisión de objetivos	161
6.2. Contribuciones de este trabajo	162
6.3. Trabajo futuro	164
Bibliografía	167

Índice de figuras

1.1. Modelo lineal de flujo de tareas.	6
1.2. Modelo lineal de flujo de tareas en sistema distribuido.	9
1.3. Flujo de tareas con <i>offsets</i>	11
1.4. Llamada a procedimiento remoto.	24
1.5. Construcciones de programación de llamada a procedimiento remoto.	25
2.1. Posible escenario de ejecución de un flujo de tareas que interfiere en la ejecución de la tarea bajo análisis en un sistema con FP.	51
3.1. Dependencia circular en sistemas distribuidos con uso de recursos compartidos.	63
3.2. Cambio en el orden de expulsión de las tareas debido al aumento del <i>jitter</i> de la tarea τ_3	65
3.3. Primer escenario de análisis de la tarea τ_1 . (a) Activación síncrona. (b) Plazo síncrono con la tarea τ_2 . (c) Plazo síncrono con la tarea τ_3	67
3.4. Segundo escenario de análisis de la tarea τ_1 . (a) Activación síncrona. (b) Plazo síncrono con la tarea τ_2 . (c) Plazo síncrono con la tarea τ_3	69

3.5.	Primer escenario de análisis de la tarea τ_2 . (a) Activación síncrona. (b) Plazo síncrono con la tarea τ_3	71
3.6.	Segundo escenario de análisis de la tarea τ_2 , activación síncrona.	72
3.7.	Primer escenario de análisis de la tarea τ_3 , activación síncrona.	73
3.8.	Segundo escenario de análisis de la tarea τ_3 . (a) Activación síncrona. (b) Plazo síncrono con la tarea τ_2	74
3.9.	Sistema con cualquier número de tareas (a) Características del sistema antes de incrementar el <i>jitter</i> de la tarea τ_{n+m+1} . (b) Características del sistema después de incrementar el <i>jitter</i> de la tarea τ_{n+m+1}	76
3.10.	Modelo de tareas del sistema del operador de puertas.	104
4.1.	Posibles escenarios de ejecución de un flujo de tareas en un sistema GC-EDF (<i>Global Clock - Earliest Deadline First</i> , Plazo más Cercano Primero con Reloj Global)	111
4.2.	Posible escenario de ejecución de un flujo de tareas en un sistema LC-EDF (<i>Local Clock - Earliest Deadline First</i> , Plazo más Cercano Primero con Reloj Local)	112
4.3.	Escenarios para calcular la contribución de la tarea τ_{ij} al periodo de ocupación.	115
4.4.	Efecto de los plazos locales para el cálculo de la contribución de la tarea τ_{ij} al periodo de ocupación.	117
4.5.	Escenarios para el cálculo de la contribución de la tarea τ_{ij} al periodo de ocupación con relaciones de precedencia	123
4.6.	Ejemplo de las reglas de reducción.	129
4.7.	Comparación de tiempos de respuesta para las diferentes técnicas en un procesador.	132
4.8.	Comparación de tiempos de respuesta para las diferentes técnicas en cuatro procesadores.	133

ÍNDICE DE FIGURAS

4.9. Utilización máxima planificable, 65 tareas.	135
4.10. Utilización máxima planificable, 40 tareas.	135
5.1. Nueva relación de precedencia en el flujo de tareas debido al efecto de la sección atómica de tareas	142
5.2. Patrón cíclico	143
5.3. Modelo genérico de flujo de tareas con una sección atómica de tareas	145
5.4. Representación de tiempos del <i>offset</i> y <i>jitter</i> de la tarea τ_{if} . .	147
5.5. Ejemplo de flujo de tareas con sección atómica	149
5.6. Tabla de interferencia del flujo de tareas Γ_i para el cálculo del tiempo de respuesta de la tarea τ_{ab} con las <i>H-sections</i> marcadas (a) y las <i>AH-sections</i> marcadas (b).	150
5.7. Ejemplo de aplicación de un flujo de tareas con sección atómica: Composición de flujos de tarea y distribución en procesadores	157
5.8. Flujo de tareas con sección atómica, e_1	158

Índice de Tablas

1.1. Notación del modelo de tareas.	8
1.2. Notación adicional del modelo de tareas con <i>offsets</i>	12
1.3. Técnicas para cálculo de tiempos de respuesta de peor caso. . .	28
1.4. Técnicas para cálculo de tiempos de respuesta de peor caso (II). .	28
3.1. Flujos de tareas del sistema.	105
3.2. Tareas del sistema.	105
3.3. Mensajes del sistema.	106
3.4. Recursos compartidos en el sistema.	106
3.5. Utilización de los recursos compartidos por las tareas.	106
3.6. Utilización de los procesadores y la red.	107
3.7. Tiempos de respuesta de peor caso de tareas y mensajes. . . .	107
4.1. Técnicas para cálculo de tiempos de respuesta de peor caso. . .	110
4.2. Técnicas para cálculo de tiempos de respuesta de peor caso (II). .	137
5.1. Características temporales de los flujos de tareas.	158

5.2. Características temporales de las tareas del sistema. 159

5.3. Comparativa de tiempos de respuesta de las tareas del sistema
para las técnicas con y sin secciones atómicas. 159

Índice de Algoritmos

2.1. Algoritmo iterativo para cálculo de tiempos de respuesta. . . .	43
2.2. crear_tabla_de_conflictos_FP.	55
2.3. resolver_conflictos_FP.	55
2.4. crear_tabla_de_conflictos_en_Γ _a _FP.	57
3.1. Extensión del algoritmo holístico para análisis de tiempos de respuesta de peor caso para EDF (<i>Earliest Deadline First</i> , Plazo más Cercano Primero).	62
4.1. construir_tabla_conflictos_EDF	125
4.2. resolver_conflictos_EDF	126
4.3. construir_tabla_conflictos_en_Γ _a _EDF	128
4.4. resolver_conflictos_en_Γ _a _EDF	130
5.1. construir_tabla_confl._con_seccion_atomica_EDF	152
5.2. resolver_confl._con_sec._atom._EDF	153
5.3. construir_tabla_confl._con_sec._atom._en_Γ _a _EDF	155
5.4. resolver_confl._con_seccion_atomica_en_Γ _a _EDF	156

Lista de acrónimos

AFDX	<i>Avionics Full Duplex Switched Ethernet</i> , Ethernet Conmutado de Doble-Canal para Aviónica
BPI	<i>Basic Priority Inheritance</i> , Herencia de Prioridad Básica
CAN	<i>Controller Area Network</i> , Red de Área de Controlador
CPU	<i>Central Processing Unit</i> , Unidad de Procesamiento Central
DDM	<i>Dynamic Deadline Modification</i> , Protocolo de Modificación de Plazo Dinámico
DFP	<i>Deadline Floor Protocol</i> , Protocolo de Suelo de Plazo
DM	<i>Deadline Monotonic</i> , Orden de Plazo Creciente
DPI	<i>Dynamic Priority Inheritance</i> , Protocolo de Herencia de Prioridad Dinámica
EDF	<i>Earliest Deadline First</i> , Plazo más Cercano Primero
FDDI	<i>Fiber Distributed Data Interface</i> , Interfaz de Datos Distribuida por Fibra
FP	<i>Fixed Priorities</i> , Prioridades Fijas
FTT-ETHERNET	<i>Flexible Time-Triggered Ethernet</i> , Ethernet Activada por Tiempos Flexible
GC-EDF	<i>Global Clock - Earliest Deadline First</i> , Plazo más Cercano Primero con Reloj Global

HOPA	<i>Heuristic Optimized Priority Assignment</i>), Asignación de Prioridades con Heurístico Optimizado
HOSDA.....	<i>Heuristic Optimized Scheduling Deadline Assignment</i> , Planificación mediante Asignación de Plazos con Heurístico Optimizado
IPCP	<i>Immediate Priority Ceiling Protocol</i> , Protocolo de Techo de Prioridad Inmediato
ISTR.....	Ingeniería de Software y Tiempo Real
LC-EDF	<i>Local Clock - Earliest Deadline First</i> , Plazo más Cercano Primero con Reloj Local
LLF.....	<i>Least Laxity First</i> , Holgura más Pequeña Primero
MAST	<i>Modeling and Analysis Suite for Real-Time Applications</i> , Herramientas de Modelado y Análisis para Aplicaciones de Tiempo Real
MIT.....	<i>Minimum Inter-arrival Time</i> , Tiempo Mínimo entre Llegadas
NPD	<i>Normalized Proportional Deadline</i> , Plazo Proporcional Normalizado
PCP	<i>Priority Ceiling Protocol</i> , Protocolo de Techo de Prioridad
PD.....	<i>Proportional Deadline</i> , Plazo Proporcional
QoS.....	<i>Quality of Service</i> , Calidad de Servicio
RM	<i>Rate Monotonic</i> , Orden de Frecuencia Creciente
RMS	<i>Rate Monotonic Scheduling</i> , Planificación Ordenada por Frecuencia creciente
RPC	<i>Remote Procedure Call</i> , Llamada a Procedimiento Remoto
RT-EP	<i>Real-Time Ethernet Protocol</i> , Protocolo Ethernet para Tiempo Real

- SRP *Stack Resource Policy*, Política de Reserva de Recursos Basada en Pila
- WCDO *Worst Case Dynamic Offset*, Peor Caso con Intervalos de Desfase (Offsets) Dinámicos
- WCDO-EDF . *Worst Case Dynamic Offset for EDF*, Peor Caso con Intervalos de Desfase (Offsets) Dinámicos para EDF
- WCDOPS *Worst Case Dynamic Offset Priority Schemes*, Peor Caso con Intervalos de Desfase (Offsets) Dinámicos y con Esquemas de Prioridad
- WCDOPS-EDF *Worst Case Dynamic Offset Precedence Schemes for EDF*, Peor Caso con Intervalos de Desfase (Offsets) Dinámicos y con Esquemas de Precedencia para EDF

Capítulo 1

Sistemas distribuidos de tiempo real

RESUMEN: Este capítulo describe los conceptos básicos sobre los que se construye esta tesis. En la sección 1.1 se repasa brevemente el significado de sistema de tiempo real. En la sección 1.2 se describen los diferentes modelos de sistemas de tiempo real existentes en la literatura y se señalan los utilizados en esta tesis. A continuación, en la sección 1.3, se definen los sistemas distribuidos y multiprocesador. Estos sistemas pueden ser planificados mediante los métodos descritos en la sección 1.4. En la sección 1.5 se describe la exclusión mutua entre tareas debida al uso de recursos compartidos. Posteriormente, en la sección 1.6, se analiza la problemática en torno a los sistemas distribuidos. Por último, en las secciones 1.7 y 1.8 se reseñan los objetivos y la organización de esta tesis.

1.1. Sistemas de tiempo real

En los sistemas de computación tradicionales una solución se considera correcta cuando lo es desde un punto de vista lógico. En cambio, los sistemas de tiempo real deben cumplir también otros requisitos adicionales. Los

sistemas de tiempo real son sistemas informáticos que debido a su naturaleza o funcionalidad tienen una fuerte interacción con el entorno, que varía dinámicamente con el tiempo físico. Estos sistemas reaccionan a eventos del entorno y deben dar respuesta satisfaciendo unos requisitos o restricciones temporales. Debido a estas restricciones, un sistema se considera correcto cuando lo es desde el punto de vista lógico, y además, cumple con todas las restricciones temporales especificadas [SR88]. Generalmente, los sistemas de tiempo real suelen ser también sistemas empotrados y suelen estar dedicados a la monitorización y control de sistemas. Además, suelen estar compuestos por recursos tales como sensores, actuadores y unidades de cómputo, que se coordinan para cumplir los requerimientos temporales.

Un sistema se puede clasificar en tres categorías diferentes en función del tipo de restricciones temporales que se le imponen:

- Sistemas de tiempo real estricto (*hard real-time*): El incumplimiento de los plazos establecidos puede tener consecuencias fatales para el sistema. Por lo tanto, es imprescindible evitar estas situaciones y el sistema debe cumplir todas las restricciones establecidas, sin excepción.
- Sistema de tiempo real no-estricto o laxo (*soft real-time*): El incumplimiento genera pérdidas de prestaciones o calidad en la respuesta pero no genera un comportamiento incorrecto. Por este motivo, el sistema puede incumplir plazos ocasionalmente en alguno de los requerimientos. Estos sistemas suelen garantizar cierta calidad o QoS (*Quality of Service*, Calidad de Servicio).
- Sistema sin requisitos de tiempo real: El tiempo de respuesta del sistema no es importante.

Indudablemente, un mismo sistema puede tener tareas con requisitos temporales de distintos tipos, combinando requerimientos temporales estrictos y laxos con tareas sin ningún tipo de requerimiento temporal.

Los requerimientos temporales que se pueden imponer a un sistema son muy variados, como, plazo de una tarea (tiempo máximo que tiene una tarea para generar su respuesta), separación máxima o mínima entre dos respuestas, tiempo de cómputo, etc... Normalmente la restricción más utilizada es la de plazo de respuesta, sobre la que se centran la mayor parte de trabajos en torno al análisis de tiempo real. En esta tesis únicamente se estudian los plazos de respuesta en sistemas de tiempo real estricto.

Los requerimientos temporales deben ser tenidos en cuenta durante la fase de diseño del sistema y, para garantizar que el sistema cumple estos requerimientos, es necesario establecer una planificación de los recursos compartidos del sistema, tales como CPUs y redes de comunicación. Existen diferentes técnicas de planificación que se clasifican en dos tipos: planificación estática (el plan de ejecución de las tareas se define en tiempo de diseño) y planificación dinámica (no existe un plan preestablecido y el orden ejecución de las tareas se decide en tiempo de ejecución). En este último grupo los problemas de concurrencia se resuelven en función de la prioridad asignada a cada tarea. Estas prioridades pueden ser fijas (prioridades fijas) o pueden cambiar dinámicamente (prioridades dinámicas). Todos estos mecanismos de planificación se explican más en detalle en la sección 1.4. Se considera que un sistema es planificable cuando existe garantía absoluta de que cumplirá todos sus requerimientos temporales en toda circunstancia. Un test de planificabilidad comprueba si el sistema es o no planificable.

Para verificar el cumplimiento de los plazos existen dos tipos de técnicas de análisis:

- Técnicas *off-line*: el análisis del comportamiento del sistema y la verificación del cumplimiento de las restricciones se realiza previamente a la puesta en marcha del sistema. De forma que, una vez se pone en marcha el sistema, se conoce su comportamiento.
- Técnicas *on-line*: el análisis de la planificabilidad se realiza en tiempo de ejecución. Cuando una tarea nueva está lista para ejecutarse en el sistema, la planificabilidad se recalcula. Si el sistema sigue cumpliendo los plazos, la tarea se acepta; si no, se rechaza. Usualmente se rechazan solo tareas de tiempo real laxo mientras se garantiza el correcto funcionamiento del conjunto de tareas de tiempo real estricto.

Todo el análisis presentado en esta tesis se basa en técnicas *off-line*. Por lo que, previamente a la puesta en marcha del sistema, el análisis debe asegurar la planificabilidad del sistema y, por lo tanto cumplir los plazos. Para ello, tanto el *software* como el *hardware* deben de tener un comportamiento predecible y estar perfectamente especificados previamente.

1.2. Modelo del sistema

Los sistemas de tiempo real están compuestos por uno o varios procesadores conectados entre sí por redes de comunicación. Además, cuentan con sensores que monitorizan el estado del entorno y actuadores que permiten influir en el comportamiento del mismo. Dependiendo de la forma en la que se da esta interacción entre elementos del sistema, el sistema se puede clasificar como:

- Sistema gobernado por eventos: Las acciones de control o tareas del sistema se activan por la llegada de eventos.
- Sistema gobernado por paso de mensaje: Una variación del sistema anterior, donde el evento de activación es un mensaje enviado por un mecanismo interno al sistema o por una red de comunicación. Esta interacción es muy utilizada en sistemas distribuidos, en los que una tarea se activa cuando recibe un mensaje de otra que ya ha acabado su ejecución.
- Sistema gobernado por tiempo: las tareas del sistema son activadas por el mismo sistema en instantes de tiempo predeterminados. Normalmente estos instantes están marcados por periodos regulares.

En el modelo de sistema utilizado para esta tesis el software se organiza en forma de un conjunto de flujos de tareas que responden a eventos. El sistema está compuesto por una combinación de los dos primeros tipos de interacción entre las tareas, en donde la primera tarea del flujo de tareas o transacción se activa mediante un evento (pudiendo ser este generado por el reloj del sistema) y las siguientes tareas mediante paso de mensaje [KDK⁺89][KZF⁺91].

Por lo tanto, el sistema contará con 3 tipos de eventos:

- Eventos externos: se producen por un evento en el sistema físico y activan un flujo de tareas en el sistema como respuesta a dicho evento.
- Eventos temporizados: son los eventos creados por temporizadores o relojes del propio sistema. También activan un flujo de tareas.

- Eventos internos: son eventos generados por el sistema para la sincronización entre varias tareas de un mismo flujo.

Independientemente del tipo de evento, su llegada activa la ejecución de un grupo de tareas en el sistema. Por lo tanto, se debe analizar el patrón de llegadas de los eventos externos y temporizados para analizar el tiempo de respuesta que tendrán. El patrón de llegadas puede ser de varios tipos:

- Periódico: Los eventos se generan a intervalos de tiempos regulares. Este intervalo se denomina periodo de activación, T . Este valor es suficiente para caracterizar el patrón de llegada de eventos.
- Esporádico: Los eventos no se generan regularmente pero se establece un tiempo mínimo entre eventos consecutivos. Dado que buscamos el peor escenario, ese tiempo mínimo lo podemos establecer como T .
- Limitado o “a ráfagas”: Los eventos no se generan a intervalos regulares pero tienen limitado un número máximo de eventos n_{max} en un intervalo de tiempo T . Por lo que se puede modelar como eventos esporádicos, equivalentes a la activación de n_{max} instancias por evento, con tiempo mínimo entre llegadas igual a T . También se puede establecer un tiempo mínimo entre la activación de dos instancias.
- Aperiódico ilimitado: Los eventos se generan a intervalos no regulares y sin límite de número. Debido a esta falta de limitación, no se puede garantizar un tiempo de respuesta limitado para estos eventos. Sin embargo, es posible planificarlos de manera que se puedan garantizar los tiempos de respuesta de los restantes eventos periódicos, esporádicos o limitados [SSL89]. La planificación de estos eventos mediante servidores aperiódicos permite modelarlos como si fuesen periódicos a los efectos de análisis del resto de las tareas.

A pesar de los diferentes tipos de eventos, desde el punto de vista del análisis de tiempos de respuesta de peor caso, en muchos casos se puede ir a un modelo de tareas periódico.

Cada evento que se da en el sistema activa una acción o tarea que ejecuta un código asociado. Para poder garantizar un tiempo de respuesta máximo es necesario conocer el tiempo máximo de ejecución o tiempo de

peor caso de la tarea, C . Este tiempo es el tiempo que tardaría en ejecutarse la tarea suponiendo que no tiene ningún tipo de interferencia (generalmente de otras tareas), es decir, tiene el uso del procesador en exclusiva. Existen muchas técnicas y herramientas para el cálculo de este tiempo [BCP03][DP05][GB10][LBJ⁺95][PB00][YZ08]. A efectos de esta tesis supondremos que este valor es conocido para todas las tareas del sistema.

En los sistemas de tiempo real es habitual que un evento externo desencadene un flujo compuesto por varias tareas sincronizadas mediante eventos internos. Este flujo de tareas se define como un flujo de tareas o transacción. En sistemas en los que es necesario garantizar tiempos de respuesta mediante técnicas off-line esta interacción entre tareas debe ser previamente conocida. En esta tesis se consideran flujos fijos en cuanto a las tareas que los componen y los procesadores en que se ejecutan.

Además, solo consideramos un flujo lineal de tareas. Es decir, cada tarea solo puede ser activada por un único evento (ya sea interno, externo o temporizado) y solo puede generar un único evento interno a su conclusión. Dicho de otra forma, una tarea puede estar precedida por una o ninguna tarea y, a su vez, puede preceder a una o ninguna (Figura 1.1). En [Gut95] se plantean flujos multicamino alternativos que permiten mayor representación de los sistemas, pero en esta tesis no los tendremos en cuenta, ya que, como se plantea en [Gut95], pueden ser transformados a flujos lineales para su análisis. En el capítulo 5 se ampliará el modelo para dar cobertura a un mayor rango de aplicaciones.

Hasta este momento hemos supuesto que un flujo de tareas comienza su ejecución en el instante en el que se dispara el evento que la activa. Esto puede no ser así en todos los sistemas debido a un efecto llamado retraso de activación (*release jitter*). El *release jitter* es la variabilidad en el tiempo de activación de una tarea. Es decir, el retraso que se da entre el instante de activación nominal o teórico y el instante de activación real. Este retraso puede deberse a diversos motivos, como desfases en la sensorización o imprecisión en el reloj del sistema, así como a la variación de tiempo de respuesta de la tarea precedente en tareas pertenecientes a un flujo de tareas. Esta magnitud la identificaremos como J y, en técnicas off-line, deberemos garantizar un valor máximo.

Además, en sistemas con flujos de tareas en las que hay relaciones de precedencia entre tareas, es decir, en los que hay tareas que se activan por eventos internos generados a la finalización de otras tareas (Figura 1.1), se

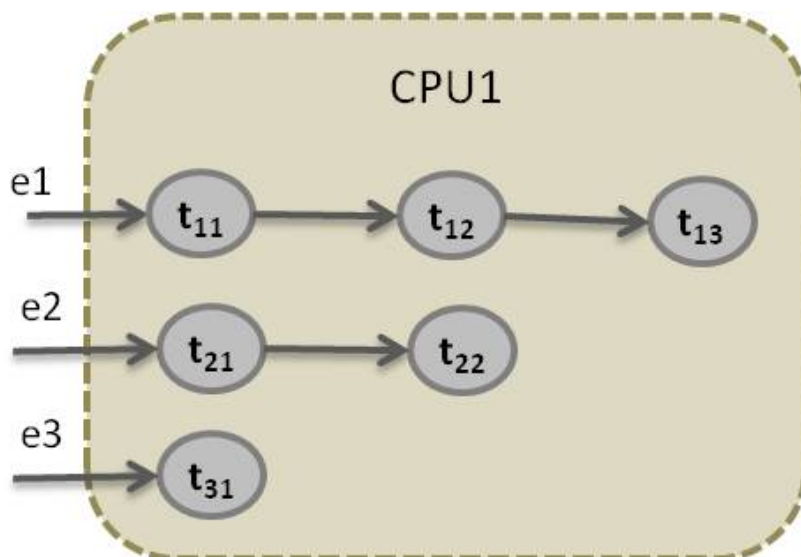


Figura 1.1: Modelo lineal de flujo de tareas.

genera otro efecto de retraso o *jitter*. En este caso, el instante en el que se activa la tarea respecto a la llegada del evento puede retrasarse en función del tiempo de respuesta de la tarea precedente. Por lo tanto, en estos casos, el valor máximo del *jitter* es la variación máxima de tiempo de respuesta de la tarea precedente (el peor caso menos el mejor caso) y se obtiene como parte del cálculo de los tiempos de respuesta de las tareas del sistema, y no como parte de la especificación del sistema.

Para simplicidad del análisis en esta tesis se asume que las sobrecargas por cambio de contexto, drivers de red o protocolos de comunicación son despreciables. El trabajo en [GHGD⁺13] demuestra que es fácil añadir estas sobrecargas cuando la técnica de análisis de tiempos de respuesta está disponible.

Resumiendo, el modelo del sistema está compuesto por flujos de tareas activados mediante eventos. Cada flujo de tareas Γ_i , a su vez, está formado por tareas enlazadas linealmente y está activado por un evento de periodo T_i . Cada tarea τ_{ij} está definida por un tiempo de ejecución máximo C_{ij} , un *jitter* máximo J_{ij} , un plazo de ejecución máximo D_{ij} y un tiempo de respuesta máximo R_{ij} . Como origen de tiempos, el plazo y el tiempo de respuesta se miden desde la llegada nominal del evento que activa la primera tarea del flujo, sin tener en cuenta el *jitter* que puede afectar a esta llegada. En la Tabla 1.3 se puede ver un resumen de los conceptos y notación introducida.

Tabla 1.1: Notación del modelo de tareas.

Concepto	Notación
Flujo de tareas	Γ_i
Tarea	τ_{ij}
Tiempo de ejecución máximo	C_{ij}
Periodo	T_i
Jitter	J_{ij}
Plazo máximo	D_{ij}

1.3. Sistemas distribuidos y multiprocesador

Cuando la respuesta a un mensaje externo se propaga por varios procesadores se deben analizar los mecanismos mediante los que se sincronizan e intercambian información dichos procesadores.

Los sistemas distribuidos y multiprocesador están compuestos por varios procesadores interconectados mediante recursos de comunicación (buses, redes locales, etc). Existen dos tipos de sistemas multiprocesador: los que utilizan memoria compartida y los que no. En el primer caso, de tratarse de un sistema de tiempo real, es necesario tener en cuenta la memoria para el análisis de planificabilidad, ya que se trata de un recurso compartido por los procesadores. Por otra parte, cabe mencionar los sistemas multinúcleo, en los que un procesador contiene varios núcleos de ejecución. Este caso es similar al sistema multiprocesador con memoria compartida, ya que los núcleos comparten la memoria del procesador. En esta tesis nos centramos en sistemas distribuidos o multiprocesador en los que no se comparte la memoria. Por tanto no se hará un análisis de la memoria compartida.

En un sistema de tiempo real en el que las tareas deben cumplir unos plazos, es necesario que la comunicación entre procesadores se realice en un tiempo acotado. En el siguiente apartado analizaremos diferentes redes y protocolos que permiten comunicación de tiempo real.

Los flujos de tareas en sistemas distribuidos además de estar compuestos por tareas, también contienen mensajes transmitidos mediante redes de comunicación. Esto se puede observar en la Figura 1.2

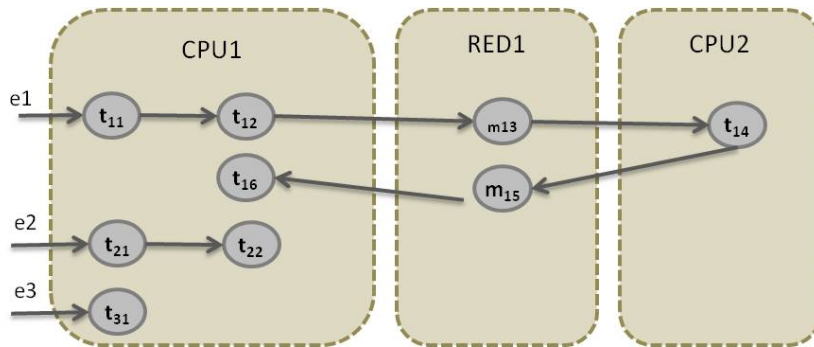


Figura 1.2: Modelo lineal de flujo de tareas en sistema distribuido.

1.3.1. Redes de comunicación

En general, la mayoría de las redes de comunicación no son compatibles con la comunicación de tiempo real y no soportan la planificación de los mensajes mediante prioridades. A pesar de esto, existen varias redes que sí lo permiten, como *token-bus* [SML88], FDDI (*Fiber Distributed Data Interface*, Interfaz de Datos Distribuida por Fibra) [ABCD91], el bus CAN (*Controller Area Network*, Red de Área de Controlador) [THW94] [TBW95] [FRB05] [DBBL07] [SSH⁺09], FTT-Ethernet (*Flexible Time-Triggered Ethernet*, Ethernet Activada por Tiempos Flexible) [PAG02], RT-EP (*Real-Time Ethernet Protocol*, Protocolo Ethernet para Tiempo Real) [MGHG03], AFDX (*Avionics Full Duplex Switched Ethernet*, Ethernet Conmutado de Doble-Canal para Aviónica) [GPGH13], etc.

Como hemos dicho, los mensajes forman parte del flujo de tareas que dan respuesta a un evento y, por lo tanto, el sistema de comunicación debe ser predecible y el tiempo de transmisión de cada mensaje debe ser acotado. Para ello, cada mensaje debe tener asociada una longitud máxima, l_m .

Además, cada mensaje podría dividirse en diferentes paquetes de tamaño fijo para su transmisión. El proceso de transmisión se divide en los siguientes pasos:

- Generación del mensaje y encolado de paquetes: La tarea emisora debe generar el paquete y, si es necesario, dividirlo en paquetes. Además debe colocar los paquetes en la cola de transmisión.

- Acceso al medio de comunicación: Una vez en la cola de comunicación, el paquete debe esperar a que el medio de transmisión quede libre.
- Transmisión del mensaje: Tiempo que requieren los paquetes del mensaje para ser transmitidos desde el origen hasta el destino. Depende del tamaño del mensaje y de la velocidad de transmisión del medio.
- Recepción y recomposición del mensaje: Una vez se reciben todos los paquetes, se unen y se recompone el mensaje.

Para analizar el cumplimiento de requerimientos temporales del sistema, se deben analizar los efectos de cada una de las fases de la transmisión. En algunos sistemas el tiempo de generación y encolado de paquetes puede considerarse como parte de la tarea emisora y, por lo tanto, este tiempo se contempla en el tiempo de ejecución de peor caso de la misma. Por contra, en otros sistemas se utiliza otra tarea independiente, de forma que, esta tarea debe ser tenida en cuenta en el análisis. De igual manera, ocurre con el tiempo de recepción y recomposición del mensaje, pero en este caso con la tarea receptora. Por último, para el cálculo del impacto que tiene el tiempo de acceso al medio y transmisión del mensaje, se deben tener en cuenta todos los mensajes del sistema.

En una red de tiempo real el tiempo de transmisión de un mensaje por la red se puede calcular de manera similar al tiempo de respuesta de una tarea en un procesador. Un mensaje puede ser especificado mediante su longitud l_m o mediante su tiempo de transmisión de peor caso, C_m , tiempo máximo de transmisión del mensaje en caso de que tuviese uso exclusivo de la red de comunicación.

Por lo tanto, a efectos de análisis, un mensaje transmitido por la red se modelará como una tarea que se activa mediante un evento interno generado por la tarea emisora y generará, a su vez, otro evento interno que activará la tarea receptora. En consecuencia, el flujo de tareas estará compuesto por un grupo de tareas (ejecutadas en procesadores) y mensajes (transmitidos por las redes) activados entre sí mediante eventos internos. Cada tarea o mensaje τ_i estará definido por:

- C_i : Tiempo de ejecución de peor caso para una tarea o tiempo de transmisión de peor caso para un mensaje.

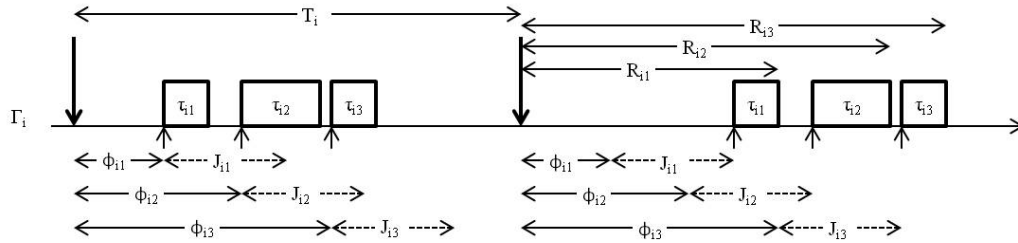


Figura 1.3: Flujo de tareas con *offsets*.

- T_i : Periodo de tiempo con el que se activa el evento de llegada que genera la ejecución del flujo de tareas al que pertenece la tarea o mensaje.
- J_i : Retraso o *jitter* máximo con el que se activa la tarea o mensaje.

1.3.2. Modelo de flujo de tareas con *offsets*

El modelo de tareas que se va a utilizar en esta tesis es el modelo transaccional presentado en [Tin94]. Una transacción o flujo de tareas es un grupo de tareas con relaciones de precedencia que se activan con igual periodo. Cada flujo de tareas Γ_i se activa mediante un evento externo de periodo T_i y *jitter* J_i y está compuesto por un grupo de m_i tareas o mensajes (recordamos que los mensajes en la red se analizan con las mismas técnicas que las tareas en los procesadores, por lo que en adelante nos referiremos a los mensajes como tareas). Cada llegada del evento externo generará una instancia del flujo de tareas. Una tarea τ_{ij} (donde j es la posición que ocupa en el flujo de tareas) del flujo de tareas se puede activar una vez transcurrido un intervalo de tiempo, llamado *offset* Φ_{ij} , desde la llegada nominal del evento externo, es decir, sin tener en cuenta el posible *jitter* de entrada.

En la Figura 1.3 se puede ver un ejemplo de un cronograma de la ejecución de un flujo de tareas. El eje horizontal representa el tiempo, mientras que cada flecha vertical hacia abajo indica la llegada nominal de una instancia del evento externo, separados por un periodo de tiempo T_i . Las flechas verticales hacia arriba indican la activación de la tarea. Como se puede observar en la figura a modo de ejemplo, las tareas de la primera instancia del flujo de tareas se han activado con *jitter* igual a cero, mientras que las de la segunda instancia se han activado con el máximo *jitter*. Cada tarea se identifica con dos subíndices: el primero hace referencia al flujo de tareas al que pertenece

y el segundo a la posición que ocupa dentro del conjunto de tareas del flujo de tareas. El rectángulo con el identificador de la tarea indica la ejecución de la misma. Por lo tanto, la tarea τ_{ij} será la j -ésima tarea del flujo de tareas Γ_i , con un tiempo de ejecución de peor caso C_{ij} . Esta tarea es nominalmente activada con un desfase de activación (offset) Φ_{ij} sobre la llegada del tiempo externo. Este tiempo se calcula como el máximo de dos tiempos: el tiempo mínimo de offset de la tarea, Φ_{ij} , y el tiempo de respuesta de mejor caso de la tarea anterior en el flujo de tareas, r_{ij-1}^b : $\phi_{ij} = \max(\Phi_{ij}, r_{ij-1}^b)$. El tiempo de activación nominal puede sufrir una variación de activación entre instancia y instancia limitada por el *jitter* máximo, J_{ij} . El máximo *jitter* que puede sufrir la tarea τ_{ij} se propaga del tiempo de respuesta de la tarea precedente, τ_{ij-1} y se calcula restando el *offset* de la tarea al tiempo de respuesta de peor caso de la tarea precedente: $J_{ij} = \max(r_{ij}, \phi_{ij}) - \phi_{ij}$. Consecuentemente, la activación de la tarea puede producirse en el intervalo $[t_e + \phi_{ij}, t_e + \phi_{ij} + J_{ij}]$, donde t_e es el instante de llegada nominal del evento externo. El tiempo de respuesta global de la tarea τ_{ij} se define por la diferencia entre el instante de finalización de la ejecución de la tarea y el instante de llegada nominal del evento externo correspondiente. El tiempo de respuesta de peor caso global se identifica con R_{ij} . En cambio, el tiempo de respuesta local se define como la diferencia entre el instante de finalización de la ejecución de la tarea y el instante de activación de la propia tarea. El tiempo de respuesta local de peor caso se identifica con r_{ij} .

En un sistemas de tareas como el descrito en esta tesis, en el que las tareas del flujo de tareas tienen relaciones de precedencia entre ellas, el *offset* de las tareas es dinámico, ya que éste depende de la respuesta de la tarea precedente en el flujo de tareas. Por lo tanto en los algoritmos generados en esta tesis y presentados en capítulos posteriores el offset se calculará en tiempo de análisis del sistema. En la Tabla 1.2 se resumen los conceptos adicionales añadidos en esta sección.

Tabla 1.2: Notación adicional del modelo de tareas con *offsets*.

Concepto	Notación
Offset	ϕ_{ij}
Offset mínimo	Φ_{ij}
Tiempo de respuesta local	r_{ij}
Tiempo de respuesta global	R_{ij}

1.4. Planificación de sistemas de tiempo real

En los sistemas de tiempo real es habitual que varias tareas tengan que compartir recursos como procesador, red de comunicación, zonas de memoria compartida, etc. Debido a que en el sistema puede haber varias tareas que simultáneamente necesiten estos recursos, es necesario realizar una política de planificación de estos recursos. Un sistema de tiempo real se considera planificable cuando, siguiendo las políticas de planificación elegidas, es capaz de cumplir las restricciones temporales establecidas.

Una política de planificación debe:

- ser predecible: Los efectos de interferencia de unas tareas sobre otras deben ser medibles con el objetivo de poder garantizar tiempos de respuesta máximos de las tareas que ejecutan en dicho planificador.
- proveer un algoritmo de planificación que establezca el orden de preferencia de las tareas sobre el uso de los recursos.
- garantizar el uso tanto de tareas periódicas como aperiódicas.
- proveer mecanismos para la ejecución de tareas sin requisitos temporales, procurando reducir sus tiempos de respuesta.
- ser lo suficientemente eficiente como para introducir una sobrecarga baja, aceptable para las aplicaciones.

Para determinar si la política de planificación elegida cumple las restricciones de tiempo establecidas es necesario realizar un test de planificabilidad. Un posible test de planificabilidad para sistemas de tiempo real consiste en calcular los tiempos de respuesta de peor caso de todas las tareas del sistema. Si todas las tareas tienen un tiempo de respuesta inferior al plazo establecido para cada una, el sistema se considera planificable. Un test de planificabilidad puede ser:

- Suficiente: Garantiza la planificabilidad del sistema. Aunque puede ser pesimista, en el sentido de que puede considerar un sistema no es planificable, aun cuando lo es. Esto se debe a que los tiempos de respuesta se han sobrestimado.

- **Exacto:** El test calcula tiempos de respuesta exactos, por lo que, en ningún caso es pesimista.

Existen test de planificabilidad exactos para sistemas monoprocesador; sin embargo, los test exactos para sistemas distribuidos o multiprocesador son infrecuentes y en raros casos son aplicables por su elevado coste computacional. En el caso de sistemas distribuidos gobernados por eventos los test exactos presentan tiempos de ejecución exponenciales, por lo que no son aplicables a sistemas prácticos complejos; por ello, los esfuerzos de la investigación se centran en mejorar o disminuir el pesimismo en los test suficientes.

Esta tesis se centra en el análisis de los recursos procesadores (como se ha mencionado antes, similar a las redes de comunicación); sin perder de vista la planificación de recursos compartidos, cuyas técnicas recogidas en la literatura se resumen en la sección 1.5. A continuación se presentan las diferentes técnicas de planificación de tareas en procesadores [Kop11][BW89].

1.4.1. Planificación estática

La planificación estática o ejecutivo cíclico [BS89] consiste en crear una ordenación cíclica o plan de ejecución de las tareas en tiempo de compilación. En este plan cada tarea tiene asignado su rodaja temporal para ejecutarse y el plan se repite cíclicamente durante la ejecución del sistema [Loc92][BW89].

El ejecutivo cíclico ofrece grandes ventajas:

- **Sencillez y eficiencia:** La sencillez que supone la implementación del planificador, que únicamente tiene que seguir una tabla con el plan preestablecido, conlleva muy poca carga sobre la ejecución de las tareas y, por tanto, una gran eficiencia.
- **Determinismo temporal:** El preestablecimiento de un plan y su simple implementación reduce los puntos de indeterminismo del sistema.
- **No hay necesidad de exclusión mutua:** No hay concurrencia en la ejecución, por lo que, las tareas pueden compartir recursos sin necesidad de establecer políticas de exclusión mutua.

1.4. Planificación de sistemas de tiempo real

- **Certificación sencilla:** La simplicidad del planificador redundante en un proceso sencillo de certificación.
- **Planificable por construcción:** Si es posible construir un plan de ejecución, no es necesario enfrentarle a un test de planificabilidad, ya que es planificable por construcción.

Por otra parte, las desventajas que conllevan su utilización también son importantes:

- **Dificultad en la generación del plan:** La necesidad de crear un plan de ejecución supone construir una tabla con todos los posibles estados de ejecución de las tareas. Esto conlleva hacer un plan para un intervalo temporal igual al mínimo común múltiplo de los periodos de todas las tareas. Esto puede resultar complicado si los periodos no son armónicos, con lo que habría que retocar dichos periodos. Además puede ser necesario partir las tareas en secciones de código para adecuarlas a las rodajas temporales. Este proceso puede resultar muy costoso y no siempre es posible.
- **Difícil mantenimiento:** Un cambio en el código de una tarea, supone tener que construir nuevamente el plan de ejecución e, incluso tener que rehacer la partición de tareas. Por lo tanto, no se respeta el principio de independencia entre la estructura lógica del sistema y su planificación.
- **Difícil tratamiento de tareas esporádicas:** Aunque se han propuesto soluciones para permitir el uso de tareas esporádicas, como servidores de consulta, el tratamiento de estas resulta complicado.

Este tipo de planificación ha resultado muy útil en infinidad de sistemas industriales, pero con el crecimiento de la complejidad en estos sistemas ha ido perdiendo protagonismo frente a los planificadores dinámicos con prioridades.

1.4.2. Planificación dinámica

Una solución más adecuada para sistemas complejos de tiempo real es la planificación dinámica o en tiempo de ejecución. En esta técnica, los problemas de concurrencia entre tareas se resuelven en el instante en el que se dan,

en función de las prioridades preasignadas. Esto hace que la implementación del planificador sea más compleja que en el caso de la planificación estática, pero permite un sistema más mantenible. El código no tiene que ser particionado, por lo que mantiene la independencia entre la estructura lógica y la planificación.

En esta política de planificación, al contrario que en la planificación estática, las tareas mantienen su concepto de tarea como hilo de ejecución (thread). Estas tareas se pueden encontrar en uno de los siguientes estados:

- En ejecución: Tarea que se encuentra ejecutándose en el procesador.
- Ejecutable: Tareas preparadas para ejecutarse a la espera de que se les asigne el procesador.
- Suspendida: A la espera de un evento (temporizado o no) o de un recurso para pasar al estado ejecutable.

La política de planificación se encarga de dar el procesador a la tarea con más prioridad de las que están en estado ejecutable.

Dentro de la planificación dinámica por prioridades se pueden encontrar políticas diferentes. Si la prioridad se mantiene fija durante toda la ejecución, estaremos hablando de planificación dinámica por prioridades fijas (FP). En cambio, si la prioridad varía en función del tiempo o del estado del sistema, hablaremos de planificación dinámica por prioridades dinámicas. La asignación dinámica de prioridades alcanza mejor utilización de los recursos de procesamiento [But05]. Por contra, su implementación es más compleja y por ello su implantación en ciertos entornos industriales está siendo más lenta.

Otro punto a tener en cuenta es la política de expulsión del planificador, pudiendo ser expulsor o no expulsor. En el planificador no expulsor, cuando una tarea empieza a ejecutarse en un procesador no es expulsada por ninguna otra tarea, a pesar de que tenga mayor prioridad, sino que continúa en ejecución hasta que se suspende programadamente. Por contra, en la política expulsora, una tarea es expulsada del procesador cuando una de mayor prioridad se activa, de modo que la CPU se asigna siempre a la tarea más prioritaria. Aunque el planificador no expulsor es más sencillo de implementar, provoca un retraso o bloqueo sobre las tareas de prioridad superior, lo que resulta en tiempos de respuesta significativamente más altos.

Aunque la elección más natural es el planificador expulsor, existen trabajos en torno al planificador no expulsor [ABB96][BB97][BB98]. En esta tesis únicamente se van a contemplar políticas expulsoras. Aun así, el efecto de bloqueo provocado por la política de no expulsión es sencillo de modelar (pesimistamente) mediante un término de bloqueo igual al mayor tiempo de ejecución de peor caso de las tareas con menor prioridad que la tarea bajo análisis.

1.4.2.1. Planificación dinámica por FP

Como hemos comentado antes en la política FP la asignación de prioridades es fija, es decir, no cambia durante el tiempo de ejecución, a no ser que sea cambiada de forma explícita. La asignación de estas prioridades de forma que se garantice la planificabilidad es un tema sobre el que se ha trabajado mucho y aún sigue abierto. La principal aportación a este campo fue la asignación de prioridades RM (*Rate Monotonic*, Orden de Frecuencia Creciente) [LL73] para sistemas monoprocesador, en donde las prioridades se asignan por periodo decreciente. Es decir, cuanto mayor plazo, menor prioridad. Esta técnica es óptima cuando los plazos son iguales a los periodos. Posteriormente este método fue mejorado para contemplar plazos menores que los periodos. Esta asignación, llamada DM (*Deadline Monotonic*, Orden de Plazo Creciente) [LW82], asigna mayor prioridad a las tareas con menor plazo. Más adelante, Audsley [Aud91] desarrolló un algoritmo heurístico para contemplar plazos mayores que el periodo que obtiene una asignación planificable siempre que exista una.

Estas técnicas de asignación son óptimas siempre que se trate de sistemas de tiempo real monoprocesador. En cambio, para sistemas distribuidos o multiprocesador, no existe una asignación de prioridades óptima. Por este motivo, se han desarrollado diversos algoritmos heurísticos buscando el mejor resultado para asignar las prioridades. Las técnicas más utilizadas han sido el templado simulado [TBW92] o el algoritmo HOPA (*Heuristic Optimized Priority Assignment*), Asignación de Prioridades con Heurístico Optimizado [GGH95]. También se han planteado soluciones mediante algoritmos genéticos [AUM⁺11].

1.4.2.2. Planificación dinámica por prioridades dinámicas

Como hemos comentado antes, en esta política, la asignación de las prioridades no es fija, sino que varía en función del estado de las tareas. El algoritmo más utilizado de este tipo es el EDF [LL73], que asigna prioridades en función del tiempo restante que queda para cumplirse los plazos de las tareas, es decir, el tiempo restante hasta el plazo absoluto (instante en el que termina el plazo para ejecutar la instancia actual de dicha tarea). El plazo absoluto se calcula sumando el plazo relativo (tiempo máximo asignado para la finalización de la tarea) al instante de activación. Es decir, la tarea que tiene el instante de cumplimiento de plazo más cercano tiene mayor prioridad. Otro algoritmo es el basado en holguras, llamado LLF (*Least Laxity First*, Holgura más Pequeña Primero) [DM89]. Se entiende holgura como el intervalo de tiempo entre instante actual y el instante de cumplimiento del plazo menos el tiempo de cómputo pendiente de ejecutar. Este algoritmo planifica en primer lugar la tarea con la menor holgura. A diferencia del EDF en el que el plazo de una instancia de una tarea no cambia (aunque sí cambie para su siguiente instancia), la holgura va cambiando con el paso del tiempo lo que introduce una mayor sobrecarga en la implementación del planificador. Otro algoritmo, más eficiente que el anterior, es el *Best-effort* (Mejor esfuerzo) [JLT85]. Este algoritmo se basa en combinar un valor asignado por el usuario en función de la “utilidad” de la tarea con el tiempo de ejecución restante de dicha tarea. Por lo que, la prioridad en tiempo de ejecución se obtiene mediante la ratio “utilidad”/tiempo_ejecución_restante.

En sistemas basados en flujos de tareas de principio a fin habitualmente los únicos requisitos impuestos son los plazos de principio a fin, es decir, el plazo que debe cumplir desde que se activa el evento externo hasta que termina la ejecución de la última tarea del flujo. Por lo tanto, no existe un plazo para cada tarea individual, lo que es necesario en una planificación EDF. Consecuentemente, necesitamos una asignación de plazos artificial.

Para sistemas distribuidos o multiprocesador existen diversos algoritmos de asignación de plazos, como PD (*Proportional Deadline*, Plazo Proporcional) o NPD (*Normalized Proportional Deadline*, Plazo Proporcional Normalizado) que asignan los plazos de planificación de las tareas en función del plazo de principio a fin del flujo de tareas [Liu00]. También existen algoritmos heurísticos de asignación de plazos, como HOSDA (*Heuristic Optimized Scheduling Deadline Assignment*, Planificación mediante Asignación de Plazos con Heurístico Optimizado) [RGPGH10], actualmente integrado en el

1.5. Exclusión mutua

más complejo HOSDA [RGPGH11]. Además en sistemas distribuidos o multiprocesador podemos contar con dos tipos de planificadores, dependiendo del reloj del sistema:

- GC-EDF(Global Clock-EDF): El sistema cuenta con un reloj común o los relojes de los diferentes procesadores están sincronizados. Los plazos de las tareas son globales, es decir, se establecen con respecto a la llegada del evento que activa el flujo de tareas, aunque éste se pueda dar en un procesador diferente.
- LC-EDF(Local Clock-EDF): El sistema no cuenta con un reloj común y no hay sincronización entre los relojes de los diferentes procesadores. Los plazos de las tareas son locales, es decir, se establecen con respecto a la activación de la tarea en su procesador.

Aunque los planificadores dinámicos son más complejos de implementar, existen en el mercado sistemas operativos como *MaRTE OS* [ARGH01] [Mar], *SHaRK* [GAGB01] [Sha], *ERIKa* [Eri] u *OSEK/VDX* [DMSW08] que los implementan y lenguajes como Ada [TDB⁺07] o Java (RTSJ [Jav]) que lo soportan. Además, podemos encontrar también un planificador dinámico implementado en las versiones más recientes de *Linux* (a partir de la versión 3.14) [Lin]. También podemos encontrar planificadores EDF en redes de comunicación [PAG02] [DNM01] [DN00] y en *middlewares* de distribución, como *RT-CORBA* [OMG].

1.5. Exclusión mutua

En el punto anterior se ha analizado la planificación de los recursos procesadores, pero no son los únicos recursos compartidos que existen en los sistemas de tiempo real. En estos sistemas son comunes también recursos de acceso mutuamente exclusivo. Es decir, mientras una tarea está utilizando un recurso exclusivo, otra no puede utilizarlo, por lo tanto, no se puede utilizar políticas expulsoras respecto a estos recursos.

Los recursos de exclusión mutua provocan un efecto de inversión de prioridad. Esto se debe a que una tarea de prioridad alta puede ser bloqueada por una segunda tarea de prioridad inferior que está utilizando un recurso de

exclusión mutua que la primera necesita. Estos efectos de inversión se pueden deber a varios motivos:

- Secciones de código no expulsable que durante su ejecución pueden bloquear tareas de prioridad superior.
- Peticiones de uso de dispositivos que se resuelven en orden de llegada (FIFO). Una tarea debe esperar a que las peticiones de tareas realizadas con anterioridad, incluidas las de menor prioridad.
- Sincronización de recursos compartidos de uso exclusivo. Dependiendo de la política utilizada para resolver el acceso mutuamente exclusivo esta inversión de prioridad puede ser acotada o no acotada, como se explica a continuación.

La sección de código que se ejecuta haciendo uso de un recurso compartido de uso exclusivo se llama sección crítica. La inversión de prioridad no acotada se da cuando una tarea de baja prioridad se encuentra ejecutando una sección crítica que bloquea un recurso compartido requerido también por una tarea de alta prioridad, a la vez que la tarea de prioridad baja es expulsada por una tarea de prioridad intermedia. Esa expulsión puede provocar una inversión de prioridad igual al tiempo de ejecución de todas las tareas de prioridad intermedia que puedan expulsar a la tarea de baja prioridad. Esa inversión puede ser muy larga y no acotable a la duración de las secciones críticas.

Otro efecto que puede provocar el uso de recursos compartidos es el bloqueo mutuo o interbloqueo (deadlock). Se produce un bloqueo mutuo cuando un conjunto de tareas necesita para su progreso recursos poseídos por un segundo conjunto de tareas y a su vez este segundo conjunto necesita para progresar recursos poseídos por el primer conjunto. Este efecto genera que ambos conjuntos de tareas se queden bloqueados perpetuamente.

El objetivo de los protocolos de sincronización de tiempo real es evitar esas inversiones de prioridad o por lo menos reducirlas y acotarlas a la duración de una o varias secciones críticas. Los principales protocolos para monoprocesador con prioridades fijas son:

- Protocolo de secciones no expulsables [Mok83]: Cuando una tarea entra en una sección crítica, no puede ser expulsada por ninguna otra.

1.5. Exclusión mutua

Es equivalente a ejecutar las secciones críticas con la prioridad más alta. Este protocolo evita tanto el bloqueo mutuo como la inversión de prioridad no acotada, pero genera tiempos de bloqueo muy grandes. Esto se debe a que cada sección crítica interfiere en todas las tareas independientemente de su nivel de prioridad y de que usen el recurso compartido o no.

- BPI (*Basic Priority Inheritance*, Herencia de Prioridad Básica) [SRL90]: En este protocolo la tarea que toma el recurso ejecuta a la prioridad más alta de las tareas que bloquea en cada instante de tiempo. Es decir, la prioridad de la tarea durante la sección crítica varía en función de las tareas a las que está bloqueando. Si durante la ejecución de la sección crítica una tarea de mayor prioridad intenta tomar el recurso, es bloqueada y por lo tanto la tarea que está ejecutando la sección crítica aumenta su prioridad automáticamente. Este protocolo elimina la inversión de prioridad, pero produce tiempos de bloqueo que pueden ser suma de varias secciones críticas. Además no evita el bloqueo mutuo.
- IPCP (*Immediate Priority Ceiling Protocol*, Protocolo de Techo de Prioridad Inmediato) [LR80]: Este protocolo introduce una variación del PCP (*Priority Ceiling Protocol*, Protocolo de Techo de Prioridad) [SRL90]. En este caso, la herencia de prioridad sucede al inicio de la sección crítica, sin esperar a que se bloquee a una tarea de mayor prioridad. Por lo tanto, las secciones críticas se ejecutan a la prioridad techo del recurso, que corresponde a la de la tarea de más alta prioridad que hace uso de dicho recurso. Su funcionamiento es similar al de las secciones no expulsables, pero permite que tareas con prioridad más alta que el techo del recurso expulsen a la tarea que está ejecutando la sección crítica. Este protocolo evita las inversiones de prioridad no acotadas y en sistemas monoprocesadores también evita el bloqueo mutuo producido por el uso de recursos mutuamente exclusivos. Además se reduce el efecto de bloqueo respecto al protocolo BPI pues como máximo se sufre bloqueo por parte de una única sección crítica.

Por otra parte, los protocolos para sistemas monoprocesador con prioridades dinámicas son:

- DPI (*Dynamic Priority Inheritance*, Protocolo de Herencia de Prioridad Dinámica) [SSRB98]: Este protocolo extiende el protocolo IPCP a los sistemas planificados mediante prioridades dinámicas.
- DDM (*Dynamic Deadline Modification*, Protocolo de Modificación de Plazo Dinámico) [Jef92]: Este algoritmo impone plazos artificiales con

el objetivo de limitar el tiempo de bloqueo que sufren las tareas.

- SRP (*Stack Resource Policy*, Política de Reserva de Recursos Basada en Pila) [Bak90] [Bak91]: Este protocolo extiende el protocolo IPCP para adaptarse a la planificación EDF y mejorar otros aspectos del protocolo.
- DFP (*Deadline Floor Protocol*, Protocolo de Suelo de Plazo) [Bur12] [GARBGH14] Este protocolo es de uso exclusivo en planificadores EDF. Mantiene las mismas propiedades que el protocolo SRP, pero simplifica la implementación y ofrece mayor eficiencia. Para ello, en vez de asignar prioridades a las tareas y calcular el techo de los recursos, utiliza los plazos ya asignados y calcula el suelo de plazo de los recursos.

Todos estos protocolos generan en sistemas monoprocesador efectos de bloqueo acotados a la duración de las secciones críticas y pueden ser modelados mediante términos de bloqueo. El bloqueo que sufre una tarea τ_{ij} es el retraso máximo causado por tareas de prioridad inferior con motivo del uso de recursos compartidos, se identifica como B_{ij} y será equivalente a la duración máxima de una o varias secciones críticas (dependiendo del protocolo).

Los protocolos anteriores son aplicables a sistemas monoprocesador. En sistemas multiprocesador, estos protocolos son aplicables a recursos locales, pero no a recursos globales (compartidos por varios procesadores). Para estos tipos de recursos son necesarios otros protocolos de sincronización que tengan en cuenta el bloqueo remoto [RSL88].

1.6. Problemática en sistemas distribuidos o multiprocesador

Como hemos comentado antes, el análisis y la planificación de sistemas de tiempo real en sistemas monoprocesadores está bien resuelto, con algoritmos de asignación de prioridades y de análisis óptimos. Por contra, en sistemas multiprocesador y distribuidos no existe solución óptima para resolver tanto el análisis como la planificación. Por lo que la investigación se centra en buscar una solución suficientemente buena.

Un problema que deben resolver las técnicas de análisis es el efecto de la activación retrasada (*release jitter*). Este retraso genera una desviación en la activación periódica de las tareas ya que cada tarea se activa mediante un evento interno generado a la finalización de la tarea precedente y, por lo tanto, la activación depende del tiempo de respuesta de la tarea anterior. Esto significa que para hacer el análisis el tiempo de respuesta de las tareas debe ser conocido a priori, lo que genera una dependencia cíclica entre tiempo de respuesta y retraso máximo. El tiempo de respuesta depende del retraso máximo y viceversa.

En [TC94] Tindell y Clark propusieron un primer test de planificabilidad suficiente que proporcionaba cotas superiores para los tiempos de respuesta de tareas en sistemas distribuidos con FP. Este test, denominado análisis holístico, resolvía el problema del *release jitter* con un algoritmo iterativo en el que se trataba las tareas como independientes. Se partía de unos valores iniciales de *jitter* iguales a cero y después de realizar el análisis se actualizaban los valores de *jitter* usando los tiempos de respuesta obtenidos. Este proceso se repetía hasta alcanzar una solución estable. Esta técnica es pesimista debido a que no tiene en cuenta las relaciones de precedencia entre las tareas. Posteriormente, Tindell [Tin94] introdujo una técnica que analiza los desfases (*offsets*) entre activaciones de tareas de un mismo flujo de tareas para reducir el pesimismo del análisis Palencia y González Harbour [PGH98] adaptaron esta técnica de análisis con *offsets* a los sistemas distribuidos y, en un trabajo posterior, Palencia y González Harbour [PGH99] volvieron a reducir el pesimismo del análisis, esta vez analizando las relaciones de precedencia entre las tareas de un mismo flujo de tareas.

En [Spu96a] y [Spu96b] Spuri adaptó la técnica holística de Tindell y Clark a sistemas distribuidos de tiempo real con prioridades dinámicas, concretamente EDF. Como el test de Tindell y Clark, se trata de un test suficiente pero con un alto pesimismo. Uno de los problemas que aborda esta tesis es que nunca se ha publicado una demostración matemática del algoritmo de Spuri, siendo este la base de muchas técnicas posteriores. Por otra parte, aunque hay trabajos que han reducido el pesimismo del análisis de Spuri, como por ejemplo [PGH05], en el que se adapta el análisis con *offsets* [PGH98] a sistemas planificados con EDF, el número de estos, en comparación con las técnicas para FP, es muy reducido. Por lo tanto, otro de los problemas que se acomete es la reducción del pesimismo en sistemas distribuidos planificados mediante EDF.

Un tercer problema que se observa en estos sistemas está relacionado con

las limitaciones que impone el modelo de flujo de tareas lineal. Este modelo presupone que, cuando los plazos son más largos que los periodos, más de una ejecución del mismo flujo de tareas pueden estar activas simultáneamente. Si el flujo de tareas tiene un tiempo de respuesta más largo que el periodo, antes de que la última tarea pueda completar su ejecución, una nueva instancia de la primera tarea habrá sido lanzada por el evento correspondiente al siguiente periodo. Esto se tiene en cuenta en las técnicas de análisis de tiempos de respuesta, pero existen algunos casos en los que el modelo no concuerda correctamente con el patrón de ejecución del sistema. Ello ocurre cuando varias tareas en el flujo de tareas comparten el mismo *thread*. Considerar el caso en el que un *thread* se auto-suspende en medio de su ejecución en una operación de espera. A veces esto se hace para adaptar la ejecución a los requisitos temporales del sistema o para controlar su *jitter* de salida [BW09]. En el modelo de análisis se dividiría ese *thread* en dos tareas de análisis y se modela la espera mediante un *offset* entre ambas tareas. En el análisis convencional es posible que esas dos tareas se activen simultáneamente, y por lo tanto, el análisis tendrá en cuenta un posible efecto de expulsión entre ellas. Sin embargo, sabemos que en la práctica esto no puede ocurrir ya que estas tareas son solo un artefacto usado para modelar un único *thread*. Por lo general, el *thread* no podrá empezar a ejecutar hasta que la instancia anterior haya completado su ejecución. Por lo tanto, el modelo de análisis no refleja la realidad y el análisis generará resultados incorrectos.

Otro ejemplo muy común de este problema es una llamada bloqueante (RPC (*Remote Procedure Call*, Llamada a Procedimiento Remoto)) en un sistema distribuido. Supongamos una tarea con un *thread* interno que realiza una llamada a un procedimiento remoto para invocar un servicio en un procesador remoto. El *thread* ejecuta una parte inicial antes de la llamada, luego hace la llamada suspendiéndose a la espera de la respuesta del nodo remoto y, cuando se recibe la respuesta, continúa con la ejecución de la parte final. Esta tarea será modelada como en la Figura 1.4, usando 5 elementos: una tarea para la parte inicial del *thread*, un mensaje de salida, una tarea en el procesador remoto, un mensaje de entrada y otra tarea para el final del *thread*. El problema es que las dos tareas ejecutadas en el procesador 1 podrían estar activas simultáneamente y los análisis convencionales tendrían en cuenta posibles efectos de expulsión entre ellas. Esto genera que el análisis sea incorrecto porque en la práctica la tarea inicial no se puede ejecutar hasta que la última tarea de la instancia anterior haya finalizado.

Es posible usar estructuras de programación que permiten a las tareas con esperas o llamadas a procedimientos remotos comportarse como el mo-

1.7. Objetivos

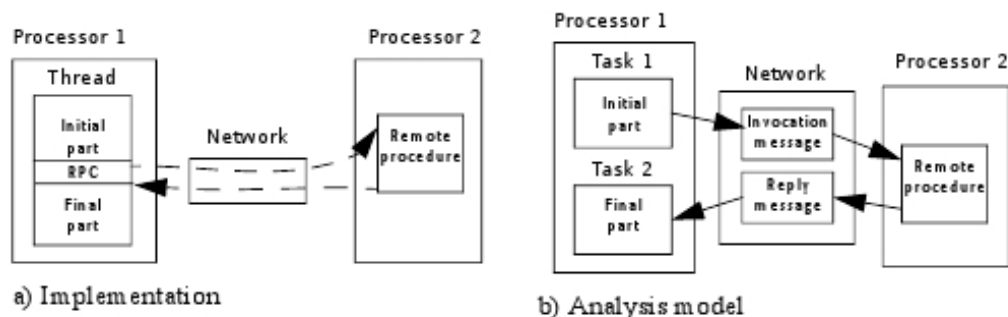


Figura 1.4: Llamada a procedimiento remoto.

delo usado en el análisis. Por lo tanto, el modelo no es incorrecto pero sí incompleto. Es necesario que el modelo de análisis permita definir dos tareas pertenecientes al mismo flujo de tareas que en la práctica se ejecutan en el mismo *thread* y son atómicas en el sentido de que la primera tarea no puede ejecutar una nueva instancia hasta que la segunda haya finalizado la ejecución de su instancia actual.

La Figura 1.5 muestra dos maneras alternativas de programar una llamada a un procedimiento remoto en un *thread* de tiempo real. La estructura de programación en la Figura 1.5-a se analizaría correctamente con las técnicas de análisis convencionales, pero la construcción más simple de la Figura 1.5-b generaría unos resultados de análisis incorrectos.

1.7. Objetivos

Como se ha señalado en la sección anterior, el desarrollo de técnicas y herramientas para el análisis de sistemas distribuidos de tiempo real planificados mediante EDF es inferior al que se ha dado en otros métodos de planificación, como FP; así como en sistemas monoprocesador. Por lo tanto el objetivo principal de esta tesis es potenciar la utilización de las planificaciones EDF mediante el desarrollo de nuevas técnicas y herramientas. Para la consecución de este objetivo se plantean tres objetivos o fases parciales.

El primer objetivo será estudiar la validez de las técnicas existentes para análisis de sistemas distribuidos de tiempo real planificados mediante EDF. Como se ha comentado anteriormente el método de Spuri [Spu96a] [Spu96b],



El Thread A espera selectivamente un `Activation_Event` o un `Reply_Message`; hace una llamada a procedimiento remoto asíncrona (no bloqueante).

El Thread B espera el `Action_Event`; hace una llamada remota bloqueante que implícitamente espera el `Reply_Message`.

Figura 1.5: Construcciones de programación de llamada a procedimiento remoto.

base de muchos de los métodos propuestos en planificaciones EDF en sistemas distribuidos, no ha sido demostrado matemáticamente. Por lo tanto, este primer paso consistirá en analizar los posibles problemas de este método y demostrar su corrección.

El segundo objetivo consiste en extender las técnicas existentes reduciendo el pesimismo de las mismas. Las técnicas desarrolladas permiten testear la planificabilidad de sistemas de tiempo real con las siguientes características:

- El sistema está distribuido por un conjunto de procesadores de propósito general, conectados entre sí por sistemas de comunicación de tiempo real acotados. Estos nodos de procesamiento pueden ser multiprocesador. Además el sistema puede contar con sensores y actuadores que permiten la interacción con el entorno físico.
- El *software* está compuesto por un conjunto de tareas distribuidas en los procesadores del sistema. Las tareas se sincronizan entre sí, mediante mecanismos locales en el mismo procesador o mediante mensajes

1.7. Objetivos

enviados a través de redes de comunicación si están en procesadores diferentes. Además estas tareas pueden compartir recursos mutuamente exclusivos locales (todas las tareas que comparten el recurso se alojan en el mismo procesador). No se soportan recursos globales compartidos por tareas en diferentes procesadores.

- Las tareas y los mensajes se agrupan en cadenas o flujos de tareas. La activación del flujo de tareas, y por lo tanto, de la primera tarea, viene dada por la llegada de un evento externo. Cada llegada del evento externo generará una instancia del flujo de tareas. La llegada de eventos debe seguir un patrón periódico o esporádico. Las siguientes tareas/-mensajes se activan por eventos internos generados a la finalización de la tarea/mensaje precedente.

La planificación del sistema a analizar debe seguir las siguientes políticas:

- Las tareas son asignadas a procesadores estáticamente de manera previa al análisis. El uso de los recursos compartidos por parte de las tareas es conocido. De forma análoga, los mensajes son asignados a las redes de forma estática.
- La ejecución de las tareas se planifica mediante un modelo de planificador dinámico expulsor basado en la planificación EDF, más concretamente, se utiliza una política LC-EDF, en la que los plazos utilizados para la planificación son locales (referenciados a la activación de la propia tarea en su procesador).
- La planificación de los recursos compartidos sigue una de las siguientes técnicas: DPI [SSRB98], SRP [Bak90][Bak91] o DFP [Bur12][GARBGH14]. Es indiferente, a efectos de análisis, cuál de ellas se utilice, ya que todas generan los mismos términos de bloqueo.
- Además se permite un modelo heterogéneo, en el que las redes o los procesadores se planifiquen y analicen mediante otras técnicas, siempre que éstas nos permitan obtener cotas inferiores al tiempo de respuesta de mejor caso y cotas superiores a los tiempos de respuesta de peor caso. Mediante la técnica presentada por Rivas et al. [RGPGH11] se pueden analizar sistemas heterogéneos con procesadores y redes con diferentes técnicas de planificación.

Los requerimientos temporales impuestos al sistema son únicamente tiempos de respuesta máximos y pueden ser de los siguientes tipos:

- Plazo máximo de respuesta local d_i . Tiempo máximo permitido desde que una tarea se activa hasta que finaliza su ejecución.
- Plazo máximo de respuesta global D_i . Tiempo máximo permitido desde que nominalmente llega el evento externo asociado al flujo de tareas de la tarea analizada hasta que dicha tarea termina su ejecución. Habitualmente se imponen plazos de principio a fin sobre el flujo de tareas, que corresponden con el plazo global de la última tarea del flujo.

En la siguiente tabla se pueden ver las técnicas disponibles previamente a la realización de esta tesis:

Tabla 1.3: Técnicas para cálculo de tiempos de respuesta de peor caso.

Técnica	FP	GC-EDF	LC-EDF
Holística	[THW94]	[Spu96b]	[RGPGH10]
WCDO	[PGH98]	[PGH03]	
WCDOPS	[PGH99]		

Como resultado de este trabajo se ha demostrado que la aplicación de la técnica WCDOPS (*Worst Case Dynamic Offset Priority Schemes*, Peor Caso con Intervalos de Desfase (Offsets) Dinámicos y con Esquemas de Prioridad) a planificaciones GC-EDF no ofrece ventajas sobre la técnica WCDO (*Worst Case Dynamic Offset*, Peor Caso con Intervalos de Desfase (Offsets) Dinámicos). Además se han desarrollado las técnicas WCDO y WCDOPS para planificaciones LC-EDF, teniendo como resultado el siguiente mapa de técnicas:

Tabla 1.4: Técnicas para cálculo de tiempos de respuesta de peor caso (II).

Técnica	FP	GC-EDF	LC-EDF
Holística	[THW94]	[Spu96b]	[RGPGH10]
WCDO	[PGH98]	[PGH03]	[DdCUGHP14a]
WCDOPS	[PGH99]	No aplicable	[DdCUGHP14a]

Por último, se plantea otro objetivo, que consiste en ampliar el modelo de tareas de forma que el test de planificación dé mayor cobertura a los sistemas

existentes en la industria. Con ello, se pretende permitir el modelado y análisis de sistemas de tiempo real no soportados por los modelos anteriores o que tenían que ser adaptados para realizar dicho análisis. Esta extensión consiste en introducir el concepto de *thread* atómico, en el cual no puede haber varias instancias (de una o varias tareas diferentes) activas en el mismo instante. Además se pretende extender el análisis este nuevo modelo, reduciendo al máximo el pesimismo.

1.8. Organización de la memoria

En este apartado se recoge la organización de los capítulos de esta memoria.

El capítulo 2 se inicia con la introducción del concepto de técnicas de análisis de planificabilidad. Además, se introducen las diferentes técnicas, tanto para sistemas monoprocesador, como para sistemas distribuidos, que sirven como base para esta tesis. Empezando desde las más antiguas, como los primeros test de planificabilidad de Liu y Layland, hasta las que sirven como base directa para las desarrolladas en esta tesis, como pueden ser las desarrolladas por Palencia y González Harbour o Spuri.

En el capítulo 3 se presenta un análisis y validación de la técnica que sirve como base para las que se desarrollarán en posteriores capítulos. Se trata de la técnica holística de Spuri para sistemas distribuidos con planificación EDF. En primera instancia se presenta el posible problema de convergencia que puede tener dicho algoritmo y posteriormente se demuestra que ese problema no puede ocurrir durante la ejecución de dicho algoritmo. Además se ha introducido un cambio al algoritmo cuando el sistema cuenta con recursos compartidos que permite la convergencia del mismo. Por último, se aplica este análisis de planificabilidad a un ejemplo industrial. Las publicaciones derivadas de este trabajo son: [DdCUGHP14b], [DdCUUGHPG14] y [DdCUGHPa].

En el capítulo 4 se presentan dos nuevos métodos que mejoran el análisis presentado en el capítulo 3. Estas dos nuevas mejoras consisten en aplicar el análisis con *offsets* y las relaciones de precedencia de FP a sistemas planificados mediante EDF. El análisis con *offsets* permite reducir el pesimismo del análisis debido al estudio de los desfases entre las activaciones de las tareas.

Por otra parte, la aplicación del análisis de precedencias, permite reducir aún más el pesimismo teniendo en cuenta las relaciones de precedencia entre tareas de un mismo flujo de tareas. Por último, se realiza una comparativa entre el método anterior y los dos nuevos algoritmos. La publicación derivada de este trabajo es: [DdCUGHP14a].

El capítulo 5 introduce una modificación en el modelo de flujos de tareas. Esta modificación pretende ampliar las posibilidades a la hora de modelar sistemas de tiempo real. Al principio de este capítulo se introduce el nuevo concepto "sección atómica de tareas", que representa un grupo de tareas dentro del flujo de tareas que debe ser ejecutado atómicamente. Es decir, dos instancias de dichas tareas no pueden estar activas simultáneamente. Después, se introducen los cambios necesarios que hay que realizar a los algoritmos del capítulo anterior para adaptarlos a este nuevo modelo. En este capítulo, además, se analizan diferentes alternativas a este modelo que existen en la literatura. La publicación derivada de este trabajo es: [DdCUGHPb].

Por último, el capítulo 6 se dedica a resumir las conclusiones obtenidas durante la tesis, así como a listar las posibles líneas de trabajo futuro.

En la parte final de esta memoria se puede encontrar una lista de referencias bibliográficas citadas a lo largo de ella.

Capítulo 2

Técnicas de análisis temporal

RESUMEN: En este capítulo se analizan los trabajos previos realizados en torno a las técnicas de análisis temporal. En primer lugar, en la sección 2.1, se introducen los test de planificabilidad para sistemas de tiempo real. En segundo lugar, en la sección 2.2, se presentan los trabajos más relevantes en sistemas compuestos por un único procesador o sistemas monoprocesador. Por último, en la sección 2.3, se analizan aportaciones en técnicas de análisis temporal para sistemas con varios procesadores, sistemas multiprocesador o distribuidos. Algunos de estos trabajos son la base sobre la que se sustenta esta tesis.

2.1. Definición de técnicas de análisis

Las técnicas de análisis temporal verifican el cumplimiento de los requerimientos temporales del sistema mediante un test de planificabilidad en tiempo de compilación. Un test puede ser:

- Suficiente: si el test resuelve que el sistema es planificable, éste lo es en cualquier circunstancia.
- Necesario: si el test resuelve que no es planificable, entonces existe al menos una condición en la que el sistema no es planificable.

- Exacto: Cuando se cumplen las dos condiciones anteriores: suficiencia y necesidad.

En sistemas de tiempo real estricto el test de planificabilidad debe ser siempre suficiente. Además, un test será más aceptable cuanto más se acerque a la condición de necesario, y por lo tanto, a la exactitud. Existen tres familias principales de test que se aplican a tiempo real, cada una basada en un concepto diferente:

- Límite de utilización [LL73] [DMTW99] [GSYY10]: Se entiende por utilización el porcentaje de ocupación del procesador. Cada tarea periódica τ_{ij} genera una utilización de:

$$U_{ij} = \frac{C_{ij}}{T_i} \quad (2.1)$$

Dependiendo de la técnica de planificación que se esté utilizando y de las características del sistema, existirá una utilización límite para que el sistema sea planificable.

- Demanda de procesador [Bar06][ZB09]: En este test se verifica que en un periodo de tiempo la demanda del procesador (la cantidad de trabajo que se ha demandado o activado) no es superior a dicho periodo.
- Tiempo de respuesta [JP86]: Este test calcula el tiempo de respuesta de peor caso de todas las tareas del sistema, o una cota superior a estos tiempos, y los compara con sus respectivos plazos. El sistema es planificable si todos se cumplen.

2.2. Sistemas monoprocesadores

En 1973, Liu y Layland presentaron el primer test de planificabilidad basado en límite de utilización [LL73]. Este test era aplicable a sistemas con tareas independientes y plazos iguales al periodo. El artículo presentaba una asignación de prioridades óptima, RMS (*Rate Monotonic Scheduling*, Planificación Ordenada por Frecuencia creciente), en la que se asigna la prioridad en función del periodo: cuanto mayor periodo menor prioridad. Esta asignación es óptima en el sentido de que, si un sistema con esta asignación no

2.2. Sistemas monoprocesadores

es planificable, no existe ninguna otra asignación con la que el sistema sea planificable. El test de planificabilidad establece que si la utilización total del procesador es igual o inferior al límite de utilización del procesador, el sistema es planificable. Es decir, si cumple:

$$U = \sum_{i=1}^N \frac{C_i}{T_i} \leq U(N) = N \left(2^{\frac{1}{N}} - 1 \right) \quad (2.2)$$

donde U es la utilización total del procesador o la suma de utilizations del conjunto de tareas del sistema y $U(N)$ es el límite de utilización máximo del procesador para N tareas.

También estableció que en estos sistemas con plazos iguales a los periodos y tareas independientes, con planificación EDF el límite de utilización es el 100%. Es decir, el sistema es planificable si cumple:

$$U = \sum_{i=1}^N \frac{C_i}{T_i} \leq 1 \quad (2.3)$$

Este test es suficiente, pero no exacto, para prioridades fijas, ya que puede determinar que un sistema no es planificable porque supera el límite de utilización, cuando en realidad sí lo es. Para planificaciones EDF sí es exacto, ya que la utilización no puede superar el 100

En 1986 Joseph y Pandya [JP86] y en 1987 Harter [HJ87] presentaron independientemente un test de planificación basado en el cálculo de tiempos de respuesta en sistemas de prioridades fijas. Este método es más potente ya que permiten analizar sistemas cuyas tareas tienen plazos inferiores o iguales al periodo y es exacto. Además permite asignaciones arbitrarias de prioridades.

El test de Joseph y Pandya se basa en el cálculo del tiempo de peor caso (R_i) de todas las tareas. Este tiempo depende de la propia ejecución de la tarea (C_i) y de la interferencia de las demás tareas (I_i):

$$R_i = C_i + I_i \quad (2.4)$$

donde la interferencia viene dado por la suma de interferencias de las tareas de mayor prioridad:

$$I_i = \sum_{j \in hp(i)} \left\lceil \frac{R_i}{T_j} \right\rceil C_j \quad (2.5)$$

donde $\lceil x \rceil$ es la función techo, que se define como el menor entero mayor o igual que x ; y $hp(i)$ es el conjunto de tareas de mayor o igual prioridad que τ_i . Como se puede comprobar en las ecuaciones (2.4) y (2.5) existe una interdependencia entre el tiempo de respuesta de peor caso (R_i) y la interferencia (I_i). En ese mismo artículo, Joseph y Pandya demostraron que esto se puede solucionar mediante una expresión iterativa:

$$R_i^{(n+1)} = C_i + \sum_{j \in hp(i)} \left\lceil \frac{R_i^{(n)}}{T_j} \right\rceil C_j \quad (2.6)$$

Por lo tanto, se van calculando nuevos valores de R_i en función de los valores anteriores. La dependencia monótona respecto al término R_i asegura la convergencia del algoritmo cuando la utilización sea menor del 100%. Se inicia el algoritmo con un valor inicial $R_i^{(0)} = C_i$ y se finaliza cuando dos valores consecutivos son iguales, $R_i^{(n+1)} = R_i^{(n)}$. El sistema es planificable siempre que todas las tareas tengan un tiempo de peor caso menor que su plazo: $R_i < D_i$. En este test el peor caso de tiempos de respuesta se obtiene cuando todas las tareas del sistema se activan simultáneamente, lo que llamamos instante crítico. Además el peor caso de cada tarea se da siempre en la primera instancia después de un instante crítico.

Posteriormente, Lehoczky [Leh90] presentó el concepto de periodo de ocupación. La ejecución de un recurso de procesamiento está compuesta por periodos de ocupación (periodo con ejecución continuada de una o varias tareas) y periodos ociosos (periodos en los que no hay tareas en ejecución). El tiempo de respuesta de peor caso se da siempre en el periodo de ocupación más largo.

En cuanto a los sistemas planificados mediante EDF, en [Liu00] Liu modificó el test presentado en la Ecuación (2.3) para adaptarla a tareas con periodo inferior al periodo:

$$U = \sum_{i=1}^N \frac{C_i}{\min(D_i, T_i)} \leq 1 \quad (2.7)$$

En 1980, Leung y Merrill [LM80] demostraron que un grupo de tareas periódicas es planificable si y solo si todos los plazos absolutos en el intervalo $[0, \max(s_i) + 2H]$ se cumplen, siendo s_i el tiempo de inicio de la tarea τ_i , $\min(s_i) = 0$ y H es el mínimo común múltiplo de los periodos de las tareas.

En 1990, Baruah et al. [BMR90] [BHR90] extendieron esta condición para sistemas con tareas esporádicas, y demostraron que un grupo de tareas es

2.2. Sistemas monoprocesadores

planificable si y solo si $\forall t > 0, h(t) \leq t$, siendo $h(t)$ la función de demanda de procesador que calcula el máximo tiempo de ejecución requerido por todas las instancias de las tareas con tiempo de activación y plazo absoluto en un intervalo contiguo de longitud t y $h(t)$ se calcula mediante:

$$h(t) = \sum_{i=1}^N \max \left\{ 0, 1 + \left\lfloor \frac{t - D_i}{T_i} \right\rfloor \right\} \quad (2.8)$$

Demostraron que usando este test de planificabilidad suficiente el valor de t puede ser limitado fácilmente. Un grupo de tareas es planificable si y solo si $U \leq 1$ y $\forall t < L_a, h(t) \leq t$, donde L_a se define como:

$$L_a = \max \left\{ D_1, \dots, D_n, \max_{1 \leq i \leq n} \{T_i - D_i\} \frac{U}{1 - U} \right\} \quad (2.9)$$

En 1996, bajo la presunción de que el plazo de las tareas es inferior o igual al periodo, Ripoll et al. [RCM96] presentaron un límite superior diferente para el test de planificabilidad:

$$L_a^2 = \frac{\sum_{i=1}^n (T_i - D_i) U_i}{1 - U} \quad (2.10)$$

2.2.1. Tareas con recursos compartidos

El uso de recursos compartidos de acceso mutuamente excluyente es muy habitual en sistemas de tiempo real por lo que los test de planificabilidad deben tenerlos en cuenta. En esta sección revisaremos los trabajos que han desarrollado este aspecto.

Suponiendo que el planificador sigue las políticas IPCP, descritas en la sección 1.5, para evitar la inversión de prioridad no acotada, una tarea τ_i que comparta uno o más recursos, sufrirá un retraso o bloqueo máximo B_i en su ejecución. Este bloqueo será equivalente a la duración de la sección crítica más larga, con techo de prioridad mayor o igual que τ_i , y ejecutadas por tareas con prioridad menor que τ_i :

$$B_i = \max_{\forall j \in lp(i)} CS_j^i \quad (2.11)$$

donde $lp(i)$ es el conjunto de tareas con menor prioridad que τ_i y CS_j^i es la duración de la sección crítica ejecutada por una tarea τ_j bloqueando un

recurso con techo de prioridad igual o mayor que la prioridad de τ_i . En cambio, si la política utilizada es BPI, el bloqueo es mayor y más complejo de calcular.

Por otra parte, si estamos utilizando una política para EDF, como son DPI o SRP, cada tarea sufre un retraso calculado con la siguiente formula:

$$B_i = \max_{\forall j} (CS_j(s) : \lceil s \rceil \geq \pi_i \text{ and } \pi_j \leq \pi_i) \quad (2.12)$$

donde $CS_j(s)$ es la sección crítica de la tarea j bloqueando el recurso s y $\lceil s \rceil$ es el techo de expulsión del recurso s , calculado como el máximo nivel de expulsión de todas las tareas que utilizan dicho recurso:

$$\lceil s \rceil = \max(\pi_i : \tau_i \text{ may lock } s) \quad (2.13)$$

siendo π_i el nivel de expulsión de la tarea τ_i , calculado mediante el plazo efectivo de las tareas ($D_i - J_i$):

$$\pi_i > \pi_j \Leftrightarrow D_i - J_i < D_j - J_j \quad (2.14)$$

En 1990, Sha, Rajkumar y Lehoczky [SRL90] extendieron la propuesta de Liu y Layland [LL73] para contemplar el uso de recursos compartidos. Un sistema es planificable, utilizando asignación de prioridades RM, si cada tarea verifica:

$$U_i = \sum_{\forall j \in hp(i)} \frac{C_j}{T_j} + \frac{B_i}{T_i} \leq U(i) = i \left(2^{\frac{1}{i}} - 1 \right) \quad \forall i = 1..N \quad (2.15)$$

Por lo tanto, una tarea es planificable si la utilización de las tareas de mayor o igual prioridad es menor o igual que el límite establecido por el numero de dichas tareas, U_i . El sistema será planificable si todas las tareas lo son. Este método fue ampliado por Lehoczky et al. [LSST91] y Klein et al. [KRP⁺93] para contemplar cualquier tipo de asignación FP, sin limitarlo a RM. En este caso el límite de utilización se establece en:

$$U_i = \sum_{\forall j \in H(i)} \frac{C_j}{T_j} + \frac{C_i}{T_i} + \frac{B_i}{T_i} + \sum_{\forall j \in S(i)} \frac{C_j}{T_i} \leq U(i) = n_i \left(2^{\frac{1}{n_i}} - 1 \right) \quad \forall i = 1..N \quad (2.16)$$

donde $H(i)$ es el conjunto de tareas con prioridad mayor y periodo menor que τ_i ; $S(i)$ es el conjunto de tareas con prioridad y periodo mayor que τ_i ; y n_i es el número de tareas en el conjunto $H(i) + 1$.

Estas últimas propuestas están todas basadas en límites de utilización, pero Audsley [ABR⁺93] extendió la técnica basada en tiempos de respuesta de Joseph y Pandya para contemplar el tiempo de bloqueo. Este método, al igual que su predecesor, permite cualquier asignación FP y además contempla el uso de plazos inferiores al periodo. También, tiene en cuenta el efecto de activación retrasada o *release jitter*. Este test se basa también en la creación de un instante crítico pero la definición de éste varía ligeramente con el de Joseph y Pandya, debido al *release jitter*. El *teorema 2-2* de [ABR⁺93] dicta que el instante crítico se produce cuando la tarea bajo análisis se activa después de haber sufrido su máximo *jitter*, además de estar recién bloqueada en el recurso compartido con sección crítica más larga, y coincidiendo con la activación, también con máximo retraso, de todas las tareas de mayor o igual prioridad. Las activaciones posteriores se dan sin retraso.

Teniendo en cuenta la definición anterior, el tiempo de respuesta de peor caso de una tarea es igual a:

$$R_i = B_i + C_i + I_i + J_i \quad (2.17)$$

donde J_i es el máximo retraso que puede sufrir la tarea τ_i .

Debido al retraso en la activación retrasada, para el cálculo de la interferencia I_i , ya no se puede utilizar como referencia el intervalo del tiempo de respuesta R_i , como en la ecuación (2.5). Hay que definir un nuevo intervalo w_i que va desde la activación de la tarea hasta el instante de finalización de su ejecución. La interferencia máxima se define como:

$$I_i = \sum_{j \in hp(i)} \left\lceil \frac{w_i + J_i}{T_j} \right\rceil C_j \quad (2.18)$$

y el instante de finalización como:

$$w_i = B_i + C_i + I_i \quad (2.19)$$

y, por lo tanto, el tiempo de respuesta:

$$R_i = w_i + J_i \quad (2.20)$$

Como se puede observar, existe una interdependencia entre w_i y J_i que se puede solucionar mediante un método iterativo al igual que en (2.17). Combinando las ecuaciones (2.18) y (2.19) se obtiene:

$$w_i^{n+1} = B_i + C_i + \sum_{j \in hp(i)} \left\lceil \frac{w_i^{(n)} + J_i}{T_j} \right\rceil C_j \quad (2.21)$$

Si nos centramos en la planificación mediante EDF, Baker [Bak90] aplicó el algoritmo SRP para la sincronización de tareas al análisis de planificabilidad. Partiendo del test de utilización añadió el calculo del bloqueo al análisis. Mediante la siguiente formula se puede obtener si un sistema con N tareas periódicas con plazos menores o iguales al periodo planificado con EDF y con recursos compartidos es planificable:

$$\forall k_{k=1,\dots,N} \left(\sum_{i=1}^k \frac{C_i}{D_i} \right) + \frac{B_k}{D_k} \leq 1 \quad (2.22)$$

2.2.2. Plazos arbitrarios

En esta sección se repasan las propuestas que permiten el uso de plazos arbitrarios, es decir, el plazo de una tarea pueden ser menor, mayor o igual que el periodo. El hecho de que el plazo pueda ser mayor que el periodo, provoca que pueda haber más de una instancia de la misma tarea pendiente de ejecución. Como consecuencia, el análisis se complica, ya que hay que tener en cuenta la interferencia de activaciones anteriores de la misma tarea. En este caso, ninguna de las asignaciones anteriores (RM y DM) de tareas es óptima. Se han propuesto algoritmos heurísticos para la asignación de prioridades [Aud91].

Lehoczky [Leh90] presentó una generalización del test para plazos menores o iguales que el periodo basado en límite de utilización. En este trabajo demuestra que la creación de un instante crítico sigue siendo válida para plazos mayores que el periodo, pero hay que extender el análisis a todas las activaciones que se dan dentro del periodo de ocupación (*busy period*) de peor caso. Un periodo de ocupación para una tarea es un periodo de tiempo durante el cual el procesador está ocupado ejecutando tareas de igual o mayor prioridad.

Para calcular el tiempo de ejecución requerido por tareas de mayor o igual prioridad que la tarea τ_i más las activaciones anteriores a la activación x (activación bajo análisis) de dicha tarea, se define un nuevo término $W_i(x, t)$:

$$W_i(x, t) = xC_i + \sum_{j \in hp(i)} \left\lceil \frac{t}{T_j} \right\rceil C_j \quad (2.23)$$

y la utilización máxima como:

$$U_i(x, t) = \frac{W_i(x, t)}{t} \quad (2.24)$$

2.2. Sistemas monoprocesadores

Por lo tanto, el sistema es planificable, si para cada activación de la tarea en el periodo de ocupación, existe un instante anterior a su plazo en el que la utilización máxima es menor que el 100%. Es decir:

$$\min_{0 < t \leq (x-1)T_i + D_i} U_i(x, t) \leq 1 \quad \forall x \in 1..N_i, \quad \forall i \in 1..N \quad (2.25)$$

donde $(x-1)T_i + D_i$ es el plazo de finalización de la activación x de la tarea τ_i y N_i es el número de activaciones de la tarea τ_i en el periodo de ocupación. Es sencillo calcular la última activación en el periodo de ocupación ya que será la primera que finalice su ejecución sin que se haya activado la siguiente:

$$N_i = \min \left(x \mid \min_{0 < t \leq xT_i} U_i(x, t) \leq 1 \right) \quad (2.26)$$

También se presentaron otras técnicas con *deadlines* arbitrarios, como [BMR90, BHR90]. En cuanto a las técnicas basadas en tiempos de respuesta, Tindell, Burns y Wellings [TBW94] presentaron un test exacto que incorporaba el efecto del *jitter* y la compartición de recursos. La técnica se basa en buscar el tiempo de respuesta de peor caso entre todas las activaciones dentro del periodo de ocupación. Para ello define el tiempo de finalización de la activación q (siendo $q = 0$ la primera activación) como:

$$w_i(q) = (q+1)C_i + B_i + \sum_{j \in hp(i)} \left\lceil \frac{w_i(q) + J_i}{T_j} \right\rceil C_j \quad (2.27)$$

Esta ecuación puede ser resuelta mediante el método iterativo anteriormente definido, con un valor inicial $w_i^{(0)} = (q+1)C_i$. Teniendo en cuenta que el instante de activación de la instancia q de la tarea τ_i es $-J_i + qT_i$, el tiempo de respuesta de dicha activación es:

$$R_i(q) = w_i(q) + J_i - qT_i \quad (2.28)$$

Y por lo tanto, el tiempo de respuesta de peor caso será igual al máximo del tiempo de respuesta de todas las activaciones en el periodo de ocupación:

$$R_i = \max_{q=0,1,2,\dots} R_i(q) \quad (2.29)$$

Además, en este trabajo demostraron que no es necesario analizar todas las activaciones dentro del periodo de ocupación, si no que se puede detener el análisis cuando se cumple la condición:

$$w_i(q) \leq (q+1)T_i \quad (2.30)$$

Como en trabajos anteriores, el sistema es planificable cuando todas las tareas tienen un tiempo de ejecución de peor caso menor que el plazo de ejecución asignado.

2.2.3. Plazos arbitrarios para planificaciones EDF

En cuanto a sistemas EDF con plazos arbitrarios en los que el plazo puede ser superior al periodo de las tareas, en los siguientes trabajos [B⁺06] [GRS96] [HBJK06] se estableció una condición necesaria y suficiente para la planificabilidad. Esta condición se da si y solo si $U \leq 1$ y $\forall t < L_a, h(t) \leq t$, siendo L_a :

$$L_a = \max \left\{ D_1, \dots, D_n, \frac{\sum_{i=1}^n (T_i - D_i) U_i}{1 - U} \right\} \quad (2.31)$$

Por otra parte, en 1996, Spuri [Spu96a] presentó un método de cálculo de tiempos de respuesta para analizar la planificación en sistemas monoprocesadores con planificador EDF.

De acuerdo con el Lemma 4.1 en [Spu96a], para calcular el tiempo de respuesta de peor caso es suficiente estudiar la planificación del patrón de llegada de más demanda en el primer periodo de ocupación. El tamaño L de este periodo de ocupación se puede calcular mediante una formula iterativa simple:

$$\begin{cases} L^{(0)} = \sum_{i=1}^n C_i \\ L^{(m+1)} = W(L^{(m)}), \end{cases} \quad (2.32)$$

donde $W(t)$ es la carga de trabajo acumulada en el instante de tiempo t , siendo esta la suma de tiempos de computación C de las instancias de las tareas que se han activado antes o en el instante t :

$$W(t) = \sum_{i=1}^n \left\lceil \frac{t + J_i}{T_i} \right\rceil C_i \quad (2.33)$$

Este mismo lema dice que el tiempo de respuesta de una tarea τ_i se da en un periodo de ocupación en el que el resto de tareas se activan síncronamente al inicio del periodo de ocupación (instante crítico) habiendo sufrido su mayor *jitter* y el resto de instancias de las tareas a su máximo ritmo, lo antes posible en el periodo de ocupación. Mencionar que, a diferencia del caso de prioridades fijas, la tarea bajo análisis no tiene que ser necesariamente

2.2. Sistemas monoprocesadores

activada al inicio del periodo de ocupación. Se establece $t = 0$ como el inicio del periodo de ocupación, y a como el instante en el que una instancia de la tarea bajo análisis se activa dentro del periodo de ocupación. La siguiente fórmula sirve para calcular la interferencia causada por otras tareas en un tiempo t sobre la ejecución de una instancia de la tarea τ_i activada en el instante a . Se debe tener en cuenta que solo las tareas con plazo absoluto más temprano que el plazo absoluto de la instancia bajo análisis D pueden interferir.

$$W_i(a, t) = \sum_{\tau_j \neq \tau_i, D_j \leq a + D_i + J_i} \min \left\{ \left\lceil \frac{t + J_i}{T_j} \right\rceil, 1 + \left\lfloor \frac{a + D_i + J_j - D_j}{T_j} \right\rfloor \right\} C_j \quad (2.34)$$

A partir de la ecuación anterior, se puede calcular la longitud $L_i(a)$ del periodo de ocupación a partir de la activación a , mediante la siguiente formula iterativa que tiene en cuenta la interferencia de las instancias de otras tareas con plazo más temprano, la ejecución de instancias de la tarea τ_i que se han activado antes o en el instante a , y el tiempo de bloqueo $B_{k(a+D_i)}$ causado por instancias con plazo posterior que comparten recursos con instancias dentro del periodo de ocupación:

$$\begin{cases} L_i^{(0)}(a) &= \sum_{j \neq i} C_j, \\ L_i^{(m+1)}(a) &= W_i(a, L_i^{(m)}(a)) + \left(1 + \left\lfloor \frac{a + J_i}{T_i} \right\rfloor\right) C_i + B_{k(a+D_i)}. \end{cases} \quad (2.35)$$

siendo $k(x) = \max(k : D_k - J_k \leq x)$.

De acuerdo al protocolo SRP [Bak90][Bak91], el tiempo B_i que una tarea τ_i puede sufrir está limitado a una única sección crítica cs . Por lo tanto, el tiempo de bloqueo es calculado con la sección crítica más larga con un nivel de expulsión π_j menor o igual que el nivel de expulsión π_i de la tarea τ_i y un techo de prioridad del recurso compartido $\lceil s \rceil$ mayor que el nivel de expulsión π_i :

$$B_i = \max \{ cs_j(s) : \lceil s \rceil \geq \pi_i \ \& \ \pi_j \leq \pi_i \} \quad (2.36)$$

donde el orden de niveles de expulsión se establece mediante el plazo efectivo (definido como la diferencia entre el plazo relativo y el *release jitter*: $D_i - J_i$) y el techo de prioridad de un recurso compartido s es el máximo de los niveles de expulsión de las tareas que pueden bloquear dicho recurso:

$$\begin{aligned} \pi_i > \pi_j &\iff D_i - J_i < D_j - J_j \\ \lceil s \rceil &= \max \{ \pi_i : i \text{ may lock } s \} \end{aligned} \quad (2.37)$$

Una vez determinado el valor $L_i(a)$, el tiempo de respuesta de peor caso relativo a la activación en el instante a es:

$$R_i(a) = \max \{J_i + C_i + B_i, L_i(a) - a\} \quad (2.38)$$

Por último, de acuerdo con el Lemma 4.1 en [Spu96a], el tiempo de respuesta de peor caso para la tarea τ_i es:

$$R_i = \max_{a \geq -J_i} \{R_i(a)\} \quad (2.39)$$

Afortunadamente, no es necesario calcular el valor $R_i(a)$ para todos los valores $a \geq -J_i$. L , la longitud del primer periodo de ocupación, es la máxima longitud de todos los periodos de ocupación y, por lo tanto, los valores significativos de a están en el intervalo $[-J_i, L - J_i - C_i - B_i]$. Es más, solo es necesario analizar los valores de a que, estando dentro de este intervalo, hacen que una instancia de la tarea τ_i se active en el instante $t = 0$ o los valores que hacen que el plazo absoluto de la instancia bajo análisis coincida con el plazo absoluto de la instancia de otra tarea.

2.3. Sistemas multiprocesadores y distribuidos

2.3.1. Método holístico para FP

En 1994 Tindell y Clark extendieron la formulación desarrollada para planificación por prioridades fijas en sistemas monoprocesadores [TBW92] [TBW94] a sistemas distribuidos [TC94]. El análisis holístico propuesto trataba las tareas como independientes, manteniendo las relaciones de precedencia mediante el *release jitter*. Es decir, el *release jitter* de una tarea depende del tiempo de respuesta de la tarea precedente:

$$J_{ij} = R_{ij-1} \quad (2.40)$$

donde R_{ij-1} es el tiempo de respuesta global de peor caso correspondiente a la tarea precedente a τ_{ij} en el flujo de tareas Γ_i que da respuesta al evento e_i . Si τ_{ij} es la primera tarea del flujo de tareas, se considera $R_{ij-1} = 0$.

Teniendo esto en consideración el tiempo de respuesta de una tarea con un *jitter* dado se puede calcular mediante las ecuaciones (2.27) - (2.30) citadas en la sección anterior y pertenecientes al trabajo [TBW94].

2.3. Sistemas multiprocesadores y distribuidos

Se puede observar que esta técnica tiene una dependencia cíclica entre el tiempo de respuesta y el *jitter*. Los tiempos de respuesta se calculan mediante la ecuación (2.28), utilizando los tiempos de finalización calculados con la ecuación (2.27), que requiere del *jitter* calculado en la ecuación (2.40), que a su vez, depende de los tiempos de respuesta obtenidos en la ecuación (2.28). Para resolver esta dependencia circular, Tindell y Clark propusieron un algoritmo iterativo para obtener los tiempos de peor caso de las tareas. Este algoritmo (Algoritmo 2.1) estima unos valores iniciales para los términos de *jitter* y calcula el tiempo de respuesta de las tareas para esos valores de *jitter* mediante las ecuaciones (2.27) y (2.28). Estos nuevos valores de los tiempos de respuesta se usan para volver a estimar los valores de *jitter*, mediante la ecuación (2.40). Este proceso se realiza repetitivamente hasta que en dos iteraciones consecutivas se obtienen los mismos valores de los tiempos de respuesta. El carácter monótono de las ecuaciones garantiza que el cálculo converja hacia una solución.

Algoritmo 2.1: Algoritmo iterativo para cálculo de tiempos de respuesta.

```
1 begin
2   Inicializar términos de jitter ( $J_i^{(0)} = 0$ );
3    $n = 0$  ;
4   repeat
5     Estimar tiempos de respuesta ( $R_i^{(n+1)}$ ) en función del jitter
      ( $J_i^{(n)}$ ) ;
6     Estimar nuevos términos de jitter ( $J_i^{(n+1)}$ ) en función de los
      nuevos tiempos de respuesta ( $R_i^{(n+1)}$ );
7      $n = n + 1$ ;
8   until  $R_i^{(n+1)} = R_i^{(n)}$ ;
9    $R_i = R_i^{(n)}$ ;
```

Para reducir el pesimismo del método holístico, en 1998 Palencia et al. [PGGH98] introdujeron el cálculo de tiempo de respuesta de mejor caso (R_i^b), lo que permitía reducir el efecto *release jitter* estimado que las tareas generan en sus sucesoras. El nuevo término de *jitter* se calcula mediante la ecuación:

$$J_i = R_{i-1} - R_{i-1}^b \quad (2.41)$$

Para el cálculo del tiempo de respuesta de mejor caso propusieron la

ecuación:

$$R_i^b = w_i^b + R_{i-1}^b \quad (2.42)$$

donde w_i^b es la contribución de mejor caso de su propio tiempo de ejecución más la contribución de mejor caso de tareas de prioridad más alta:

$$w_i^b = C_i^b + \sum_{\forall j \in hp(i)} \left[\frac{w_i^b - (T_j + R_{j-1} - R_j^b)}{T_j} \right]_0 C_j^b \quad (2.43)$$

En este trabajo se demostró que la mejora que ofrecía realizar el análisis completo de la Ecuación (2.43) no era muy superior a utilizar una formulación más sencilla basada en sumar los todos los tiempos de mejor caso:

$$R_i^b = C_i^b + \sum_{\forall j \in hp(i)} C_j^b \quad (2.44)$$

Más tarde, Redell y Sanfridson [RS02] y Bril et al. [BCGG09] [BLM13] mejoraron el análisis de mejor caso de Palencia et al. [PGGH98] para reducir el pesimismo. En ambos casos el tiempo de respuesta de mejor caso se calcula mediante la siguiente formula:

$$R_i^b = C_i^b + \sum_{\forall j \in hp(i)} \left[\frac{R_i^b - J_j - T_j}{T_j} \right]_0 * C_j^b \quad (2.45)$$

donde $[x]_0$ es igual a $\min([x], 0)$.

2.3.2. Método holístico para EDF

En 1996, Spuri aplicó el método holístico de Tindell y Clark, descrito en la sección anterior, a sistemas distribuidos con planificadores EDF [Spu96b] basándose en las ecuaciones para sistemas monoprocesadores presentada en la sección 2.2.3.

Al basarse en el método holístico el cálculo de tiempos de respuesta se realiza mediante el mismo algoritmo iterativo (Algoritmo 2.1) y el *jitter* de las tareas se calcula mediante la ecuación (2.40) o la ecuación (2.41), en caso de utilizar tiempos de respuesta de mejor caso. Por otra parte, el cálculo del tiempo de respuesta para un *jitter* dado se calcula mediante las ecuaciones (2.32)-(2.39) de la sección 2.2.3.

Esta técnica de análisis solo contemplaba planificadores GC-EDF, es decir, planificadores para sistemas con planificadores EDF y reloj global. Posteriormente, Rivas et al. [RGPGH10] extendieron el análisis a LC-EDF, lo que permite aplicar este análisis a sistemas sin sincronización de relojes.

2.3.3. Método con *offsets* dinámicos para FP

En 1998, Palencia y González Harbour [PGH99] presentaron una nueva técnica de análisis para FP que se basaba en la técnica holística de Tindell y Clark [TC94] y que reduce significativamente el pesimismo de esta. Al basarse en la técnica holística, utiliza el mismo algoritmo (Algoritmo 2.1) para el cálculo de los tiempos de respuesta. En [Tin94] Tindell, introdujo un concepto nuevo llamado *offset*, que permite modelar desfases entre las tareas de un mismo flujo de tareas, para así aprovechar estos intervalos ociosos para dar la oportunidad a otras tareas de prioridad inferior a ejecutar. Palencia y González Harbour extendieron el uso de los *offsets* a sistemas distribuidos para representar el retraso mínimo que puede sufrir una tarea debido a la ejecución de sus tareas precedentes y con ello reducir el tiempo de respuesta. Por lo tanto, el instante de activación de una instancia de la tarea τ_{ij} ocurrirá en el intervalo de tiempo entre $t_0 + \phi_{ij}$ y $t_0 + \phi_{ij} + J_{ij}$, siendo t_0 el instante de llegada del evento externo. Con lo que el cálculo del *jitter* cambiaría:

$$\begin{aligned}\phi_{ij} &= \max(\Phi_{ij}, r_{ij-1}^b) \\ J_{ij} &= R_{ij-1} - \phi_{ij} + J_{clock}\end{aligned}\tag{2.46}$$

donde Φ_{ij} es el *offset* mínimo de la tarea y J_{clock} representa el efecto de retraso debido a la mala resolución o imprecisión en la sincronización del reloj.

Para el cálculo del tiempo de respuesta de una tarea τ_{ab} de peor caso es necesario crear el escenario de peor caso para dicha tarea. Para ello, se debe crear un instante crítico que inicie el periodo de ocupación de peor caso. Con la introducción de los *offsets*, hay que tener en cuenta que, al contrario del caso en el que las tareas eran consideradas independientes, ahora todas las tareas de mayor prioridad no pueden activarse simultáneamente en el instante crítico. El análisis calcula la interferencia de peor caso de la transacción Γ_i en la ejecución de la tarea τ_{ab} en el instante crítico creado a partir de la instancia de la tarea τ_{ik} con mayor retraso:

$$W_{ik}(\tau_{ab}, t) = \sum_{\forall j \in hp_i(\tau_{ab})} \left(\left\lfloor \frac{J_{ij} + \varphi_{ijk}}{T_i} \right\rfloor + \left\lceil \frac{t - \varphi_{ijk}}{T_i} \right\rceil \right) C_{ij}\tag{2.47}$$

donde $hp_i(\tau_{ab})$ representa el conjunto de tareas perteneciente al flujo de tareas Γ_i y que se ejecutan en el mismo procesador y con prioridad igual o mayor que la tarea τ_{ab} . φ_{ijk} es el periodo de tiempo entre el instante crítico y la activación de la primera instancia de la tarea τ_{ij} en el periodo de ocupación:

$$\varphi_{ijk} = T_i - (\phi_{ik} + J_{ik} - \phi_{ij}) \text{ mód } T_i \quad (2.48)$$

El problema de este análisis es que no se conoce qué τ_{ik} se debe utilizar para crear el periodo de ocupación de peor caso y el analizar todas las posibles combinaciones resultaría intratable, debido a la explosión combinatorial en sistemas complejos. Por ello Tindell introduce un método aproximado en el que se obtiene un límite superior de la interferencia de las tareas del flujo de tareas Γ_i en un periodo de ocupación de longitud w , como el máximo de todas las posibles interferencias que pueden ser causadas considerando cada una de las tareas de Γ_i como la creadora del periodo de ocupación:

$$W_i^*(\tau_{ab}, w) = \max_{\forall k \in hp_i(\tau_{ab})} W_{ik}(\tau_{ab}, w) \quad (2.49)$$

Para reducir el pesimismo en el cálculo, en el flujo de tareas Γ_a al que pertenece la tarea bajo análisis, no se utilizará esta función, si no que se usará la original (2.47). Consecuentemente, deberemos considerar los instantes críticos creados a partir de todas las tareas τ_{ac} en el conjunto $hp_a(\tau_{ab})$, incluyendo la tarea τ_{ab} . El tiempo de finalización de cada instancia en el periodo de ocupación, $w_{abc}(p)$, se calcula mediante:

$$w_{abc}(p) = B_{ab} - (p - p_{0,abc} + 1)C_{ab} + W_{ac}(\tau_{ab}, w_{abc}(p)) + \sum_{\forall i \neq a} W_i^*(\tau_{ab}, w_{abc}(p)) \quad (2.50)$$

donde el parámetro $p_{0,abc}$ corresponde a la primera activación de τ_{ab} en el periodo de ocupación creado con τ_{ac} , y por lo tanto la primera que debe ser analizada:

$$p_{0,abc} = - \left\lfloor \frac{J_{ab} + \varphi_{abc}}{T_a} \right\rfloor + 1 \quad (2.51)$$

Y $p_{L,abc}$ es la última activación dentro del periodo de ocupación y, por lo tanto, la última que debe ser analizada:

$$p_{L,abc} = - \left\lceil \frac{L_{abc} - \varphi_{abc}}{T_a} \right\rceil \quad (2.52)$$

siendo L_{abc} la longitud del periodo de ocupación:

$$L_{abc} = B_{ab} + \left(\left\lceil \frac{L_{abc} - \varphi_{abc}}{T_a} \right\rceil - p_{0,abc} + 1 \right) C_{ab} + W_{ac}(\tau_{ab}, L_{abc}) + \sum_{\forall i \neq a} W_i^*(\tau_{ab}, L_{abc}) \quad (2.53)$$

El tiempo de respuesta de peor caso de una instancia p se calcula restando el instante de llegada nominal del evento asociado al instante de finalización:

$$R_{abc}(p) = w_{abc}(p) - \varphi_{abc} - (p - 1)T_a + \phi_{ab} \quad (2.54)$$

Y el tiempo de respuesta de peor caso de la tarea se obtiene mediante el máximo de todas las activaciones:

$$R_{ab} = \max_{\forall c \in hp_a(\tau_{ab}) \cup b} \left[\max_{p=p_{0,abc} \dots p_{L,abc}} (R_{abc}(p)) \right] \quad (2.55)$$

2.3.4. Método con *offsets* dinámicos para EDF

En [PGH03] y [PGH05], Palencia y González Harbour extendieron el análisis basado en *offsets* de la sección 2.3.3 a sistemas EDF. La técnica, a su vez, se basa en la técnica holística de Spuri resumida en la sección 2.3.2 y reduce el pesimismo de esta al aplicar el uso de *offsets*.

En este método, el cálculo de *offsets* y *jitters* se realiza de la misma forma que para FP, mediante la ecuación (2.46). La mayor diferencia de este método radica en el cálculo de la interferencia que ejercen las tareas del sistema sobre la tarea bajo análisis. La contribución que genera una tarea τ_{ij} al periodo de ocupación de longitud t y plazo D cuando el inicio del periodo de ocupación coincide con la activación de la tarea τ_{ik} es:

$$W_{ijk}(t, D) = \left(\left\lceil \frac{J_{ij} + \varphi_{ijk}}{T_i} \right\rceil + \min \left(\left\lceil \frac{t - \varphi_{ijk}}{T_i} \right\rceil, \left\lceil \frac{D - \varphi_{ijk} - d_{ij}}{T_i} \right\rceil + 1 \right) \right)_0 C_{ij} \quad (2.56)$$

donde, $(x)_0$ indica que, si x es negativo, el resultado es 0. φ_{ijk} se puede calcular mediante la ecuación (2.48).

La contribución total de un flujo de tareas se calcula sumando la contribución de todas las tareas:

$$W_{ik}(t, D) = \sum W_{ijk}(t, D), \forall j \in \Gamma_i \quad (2.57)$$

Para calcular el tiempo de respuesta de peor caso de la tarea τ_{ab} es necesario analizar todos los posibles periodos de ocupación construidos eligiendo una tarea de cada flujo de tareas. Como en la sección anterior, esto es intratable y se puede utilizar un aproximación mediante un límite superior. Por lo tanto, la máxima interferencia que el flujo de tareas Γ_i genera es:

$$W_i^*(t, D) = \max(W_{ik}(t, D)), \forall k \in \Gamma_i \quad (2.58)$$

Igual que en la técnica para FP, para reducir el pesimismo en el cálculo, en el flujo de tareas Γ_a al que pertenece la tarea bajo análisis no se utilizará esta ecuación, sino que se usará la original (2.57). Por ello, se deben considerar los instantes críticos creados a partir de todas las tareas del flujo de tareas Γ_a .

No se conoce cuál es el instante crítico dentro del periodo de ocupación, pero se sabe que puede estar al principio del periodo de ocupación o cuando el plazo de la tarea τ_{ab} coincide con el plazo de otra tarea. Como una tarea puede tener varias activaciones en el periodo de ocupación, hay que analizar los instantes críticos creados a partir de todas. Las activaciones de una tarea en el periodo de ocupación están en el rango $[p_{0,ijk}, p_{L,ijk}]$, que se puede calcular con las ecuaciones (2.51) y (2.52) de la sección anterior. Por lo tanto, el conjunto de valores a analizar, Ψ , viene dado por los instantes en los que finalizan los plazos de dichas activaciones:

$$\Psi = \cup\{\varphi_{ijk} + (p - 1)T_i + d_{ij}\}, \forall p = p_{0,ijk}..p_{L,ijk}, \forall i, j, k \quad (2.59)$$

Como puede haber también varias activaciones de τ_{ab} dentro del periodo de ocupación, se deben calcular todas ellas. Es decir, se deben analizar las activaciones en el rango $[p_{0,abc}, p_{L,abc}]$. Si la primera activación de la tarea τ_{ab} ocurre en el instante A después del inicio del periodo de ocupación, el tiempo de finalización de la activación p , $w_{abc}^A(p)$, se calcula añadiendo la contribución de peor caso de todos los flujos de tareas:

$$w_{abc}^A(p) = B_{ab} + (p - p_{0,abc} + 1)C_{ab} + W_{ac}^-(w_{abc}^A(p), D_{abc}^A(p)) + \sum_{\forall i \neq a} W_i^*(w_{abc}^A(p), D_{abc}^A(p)) \quad (2.60)$$

donde W_{ac}^- es el resultado de la ecuación (2.57) para el flujo de tareas Γ_a sin considerar la contribución de la tarea τ_{ab} , y $D_{abc}^A(p)$ es el plazo de la activación p cuando la primera activación ocurre en el instante A :

$$D_{abc}^A(p) = A + \varphi_{ijk} + (p - 1)T_a + d_{ab} \quad (2.61)$$

2.3. Sistemas multiprocesadores y distribuidos

El tiempo de respuesta global de peor caso se obtiene restando al tiempo de finalización el instante de llegada del evento externo:

$$R_{abc}^A(p) = w_{abc}^A(p) - A - \varphi_{abc} - (p-1)T_a + \phi_{ab} \quad (2.62)$$

Para la activación p solo es necesario analizar los valores de A entre 0 y T_a que hagan que el plazo de dicha activación coincida con el de otra tarea. Si analizásemos valores mayores que T_a estaríamos analizando una activación p diferente. Por lo tanto, solo es necesario analizar los valores de Ψ en el subconjunto:

$$\Psi^* = \{\Psi_x \in \Psi \mid \varphi_{ijk} + (p-1)T_a + d_{ab} \leq \Psi_x < \varphi_{ijk} + pT_a + d_{ab}\} \quad (2.63)$$

Por cada elemento Ψ_x en Ψ^* , el valor de A es igual a:

$$A = \Psi_x - [\varphi_{ijk} + (p-1)T_a + d_{ab}] \quad (2.64)$$

El periodo de ocupación se puede calcular considerando la contribución sin la limitación del plazo:

$$L_{abc} = W_{ac}(L_{abc}, \infty) + \sum_{\forall i \neq a} W_i^*(L_{abc}, \infty) \quad (2.65)$$

Para calcular el tiempo de respuesta global de peor caso de la tarea τ_{ab} hay que determinar el valor máximo de todos los potenciales instantes críticos analizados:

$$R_{ab} = \max \left(R_{abc}^A(p) \right) \forall p = p_{0,abc} \cdot p_{L,abc}, \forall c \in \Gamma_a, \forall A \in \Psi^* \quad (2.66)$$

Esta técnica de análisis solo contemplaba sistemas GC-EDF, es decir, sistemas con planificador EDF y reloj global. En la sección 4.2, extendemos esta técnica a sistemas LC-EDF, en los que los plazos de las tareas son locales [DdCUGHP14a]. Esto permite aplicar este análisis a sistemas sin sincronización de relojes.

Con el objetivo de reducir el pesimismo de la técnica de Palencia y González Harbour, Pellizzoni y Lipari propusieron una modificación en el modelo [PL05]. El cambio en el modelo consiste en eliminar el *jitter* que provoca

la tarea precedente considerándolo como *offset*. Por lo tanto, el cálculo del *offset* pasa a ser:

$$\phi_{ij} = R_{ij-1} \quad (2.67)$$

Con lo que el *jitter* debido al tiempo de respuesta de la tarea precedente se anula y solo influiría el *jitter* provocado por la imprecisión del reloj:

$$J_{ij} = J_{clock} \quad (2.68)$$

Al eliminar el *jitter* se elimina también gran parte del pesimismo del análisis. La única desventaja de la modificación propuesta es que se debe modificar el sistema para que las tareas respeten el *offset* artificial que se ha establecido. Es decir, las tareas deben esperar para activarse hasta el instante en el que la tarea precedente finalizaría en su peor caso de ejecución, aunque en la activación actual no haya ocurrido dicho peor caso. Por lo tanto, el diseño del sistema estaría influenciado por el análisis del tiempo de respuesta, lo que no ocurre en la técnica de Palencia y González Harbour.

2.3.5. Método con *offsets* dinámicos y relaciones de precedencia para FP

Este método presentado por Palencia y González Harbour [PGH99] se basaba en el trabajo anterior [GHKL91], en el que se estudiaban las relaciones de precedencia entre tareas. Aunque los métodos basados en *offsets* de las secciones anteriores consideran las relaciones de precedencia entre las tareas de una forma indirecta, pueden ser pesimistas en ciertas situaciones, como la que se presenta en la Figura 2.1. La Figura 2.1-a muestra la interferencia de un flujo de tareas periódico Γ_i en el tiempo de ejecución de peor caso de la tarea bajo análisis τ_{ab} , posiblemente perteneciente a otro flujo de tareas. Las flechas descendientes indican el instante de llegada del evento externo, mientras que, las flechas ascendentes señalizan la activación de cada tarea, retrasadas respecto a la llegada del evento por el *offset* de cada tarea ϕ_{ij} , que es representado mediante flechas horizontales. El *jitter* máximo de las tareas está también representado por flechas horizontales indicando el periodo de tiempo hasta el cual se podría retrasar la activación correspondiente. Cada caja representa la ejecución de una tarea. Dos instancias del flujo de tareas están representadas con diferentes sombreados (gris el primero, blanco el segundo). Las tareas usan FP y la prioridad asignada a cada tarea es

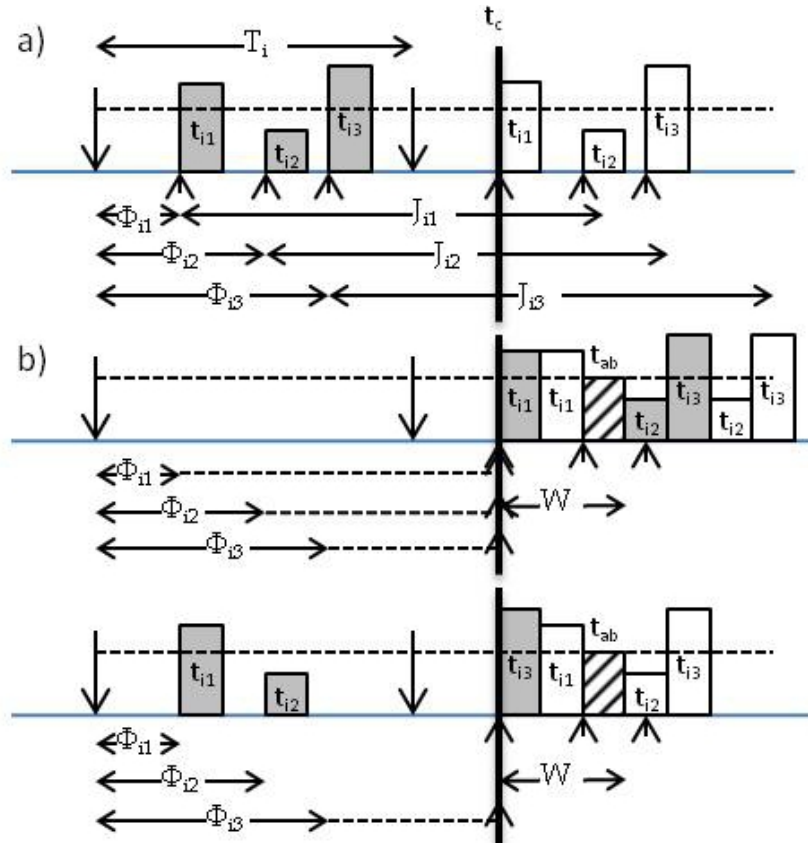


Figura 2.1: Posible escenario de ejecución de un flujo de tareas que interfiere en la ejecución de la tarea bajo análisis en un sistema con FP.

proporcional a la altura de las cajas que representa la ejecución de dicha tarea. La prioridad de la tarea bajo análisis τ_{ab} está representada por una línea discontinua horizontal.

La Figura 2.1-b muestra dos diferentes escenarios de ejecución dependiendo del retraso considerado para la activación de cada instancia, que es indicado mediante una línea discontinua horizontal en la zona inferior de la gráfica temporal. En el primer escenario se establecen los *jitters* como se haría en el método basado en *offsets* para crear la situación de peor caso. Esto es, retrasando todas las activaciones de las instancias anteriores al instante crítico de forma que coincidan con dicho instante crítico, y haciendo que el resto de activaciones se activen lo antes posible. Si se analiza el escenario sin tener en cuenta las relaciones de precedencia, todas las instancias con prioridad superior o igual que la de la tarea τ_{ab} interfieren en su ejecución. Por

lo tanto, el tiempo de finalización sería $W = 2C_{i1} + 2C_{i3} + C_{ab}$. No obstante, las relaciones de precedencia establecen que la tarea τ_{i3} no puede ejecutarse antes de que la tarea τ_{i2} haya completado su ejecución, como se puede ver en la figura. Por otra parte, la prioridad de la tarea τ_{i2} es inferior a la de la tarea τ_{ab} , y por lo tanto no se puede ejecutar antes que la tarea τ_{ab} (Ver primer escenario de la Figura 2.1-b). Consecuentemente, el tiempo de finalización de peor caso de la tarea τ_{ab} tendría que ser $W = 2C_{i1} + C_{ab}$ de acuerdo con el primer escenario. Sin embargo, como se puede ver en el segundo escenario de la Figura 2.1-b, puede haber otras situaciones cuando algunas de las instancias anteriores al instante crítico no se retrasan. Algunas instancias que en el primer escenario no pueden interferir con τ_{ab} , en el segundo escenario si pueden; y viceversa. En este segundo escenario el tiempo de finalización es igual a $W = C_{i1} + C_{i3} + C_{ab}$.

Para encontrar el peor escenario, Palencia y González Harbour [PGH99] propusieron un método para resolver los conflictos de activación entre las tareas.

Un conflicto de activación se define como:

Definición 2.1. (en [PGH99]) Dos tareas están en conflicto cuando ejecuciones de la misma instancia del flujo de tareas son incompatibles en el periodo de ocupación creado para el análisis de la tarea τ_{ab} . Supongamos que τ_{ij} y τ_{ik} son dos tareas con prioridad igual o mayor que la tarea bajo análisis τ_{ab} y τ_{ij} precede a τ_{ik} , $j < k$. Si hay otra tarea intermedia τ_{il} ($j < l < k$) en el flujo de tareas que se ejecute en el mismo procesador y con prioridad menor que τ_{ab} , entonces, en el periodo de ocupación de τ_{ab} las tareas τ_{ij} y τ_{ik} , activadas en la misma instancia del flujo de tareas, no puede interferir simultáneamente. Por lo tanto, las activaciones de las tareas τ_{ij} y τ_{ik} están en conflicto.

En el caso de la Figura 2.1 la tarea de prioridad menor, τ_{i2} , está entre las tareas de prioridad mayor τ_{i1} y τ_{i3} . Por lo tanto, hay un conflicto entre las dos últimas para el análisis de τ_{ab} . Como consecuencia, se deduce que el tiempo de finalización de peor caso de la tarea τ_{ab} es el máximo de las dos situaciones:

$$W = \max(2C_{i1} + C_{ab}, C_{i1} + C_{i3} + C_{ab}) \quad (2.69)$$

Con objeto de detectar y resolver dichos conflictos de activación, Palencia y González Harbour propusieron clasificar las tareas de un flujo de tareas Γ_i en diferentes secciones, llamadas *H-sections*. Una *H-section* es un conjunto

2.3. Sistemas multiprocesadores y distribuidos

de tareas contiguas (alojadas en el mismo procesador que τ_{ab}) con prioridad mayor o igual que la prioridad de la tarea τ_{ab} .

Definición 2.2. (en [PGH99]) Dos tareas τ_{ij} y τ_{ik} pertenecen a la misma *H-section*, para el análisis de τ_{ab} , si ambas se ejecutan en el mismo procesador que τ_{ab} , con prioridad mayor o igual que τ_{ab} y no hay tareas intermedias entre ellas, en el mismo procesador, con prioridad menor que τ_{ab} . $H_{ij}(\tau_{ab})$ identifica la *H-section* a la que pertenece τ_{ij} y está compuesta por tareas del flujo de tareas Γ_i que verifican:

$$H_{ij}(\tau_{ab}) = \{l \in \Gamma_i \mid (\exists [j \leq x \leq l \vee l \leq x \leq j] \mid \text{proc}(\tau_{ix}) = \text{proc}(\tau_{ab}) \wedge \text{prio}(\tau_{ix}) < \text{prio}(\tau_{ab}))\} \quad (2.70)$$

donde $\text{proc}(\tau_{ij})$ es el procesador donde se ejecuta la tarea τ_{ij} , y $\text{prio}(\tau_{ij})$ es la prioridad de τ_{ij} .

Las tareas τ_{ij} y τ_{ik} pertenecen a la misma *H-section* si $H_{ij}(\tau_{ab}) = H_{ik}(\tau_{ab})$.

Para calcular la máxima interferencia que el flujo de tareas Γ_i puede provocar en la ejecución de la tarea τ_{ab} , es necesario analizar todos los posibles instantes críticos creados a partir de tareas del flujo de tareas Γ_i con mayor o igual prioridad que τ_{ab} . Centrándonos en el instante crítico creado por la tarea τ_{ik} , vamos a distinguir las instancias iniciadas antes y después del instante crítico. Todas las instancias posteriores al instante crítico, con índices $p' \geq 1$, deben empezar su ejecución en orden de precedencia empezando por la primera tarea; y por lo tanto, solo la primera *H-section* puede contribuir al periodo de ocupación. La primera *H-section* se identifica como:

$$MP_i(\tau_{ab}) = \{l \in \Gamma_i \mid (\exists x < l \mid \text{proc}(\tau_{ix}) = \text{proc}(\tau_{ab}) \wedge \text{prio}(\tau_{ix}) < \text{prio}(\tau_{ab}))\} \quad (2.71)$$

La interferencia de dichas activaciones se puede calcular mediante la ecuación:

$$W_{ik}(\tau_{ab}, t) \mid_{p' > 0} = \sum_{\forall j \in MP_i(\tau_{ab})} \left[\frac{t - \varphi'_{ijk}}{T_i} \right]_0 C_{ij} \quad (2.72)$$

donde φ'_{ijk} es el instante en el que se activa la instancia $p' = 1$ (correspondiente al primer evento dentro del periodo de ocupación) de la tarea τ_{ij} , calculado mediante la ecuación:

$$\varphi'_{ijk} = T_i - (\phi_{ik} + J_{ik}) \text{ mód } T_i + \phi_{ij} \quad (2.73)$$

Como φ'_{ijk} puede ser mayor que T_i y la contribución no puede ser negativa, se debe considerar un límite de 0, y por tanto, $\lceil x \rceil_0 = \max(\lceil x \rceil, 0)$.

Para la contribución de instancias previas al instante crítico creado con la tarea τ_{ik} , se deben encontrar los conflictos entre las tareas del flujo de tareas y resolverlos. Para ello, se propuso la construcción de una tabla de conflictos para un periodo de ocupación y una tarea bajo análisis τ_{ab} . Cada fila de la tabla representa un vector de tareas pertenecientes a Γ_i y cada columna representa las diferentes instancias de una tarea (instancias con valores $p' \leq 0$). Cada celda (j, p') puede tener dos valores diferentes: 0 si la activación p' de la tarea τ_{ij} no puede darse dentro del periodo de ocupación, o C_{ij} si la activación puede darse dentro del periodo de ocupación.

Se debe construir una tabla con N columnas y n'_{iNk} filas, siendo N el número de tareas en el conjunto $hp_i(\tau_{ab})$ y n'_{iNk} es el número de activaciones pendientes de la última tarea en $hp_i(\tau_{ab})$. El proceso de creación de la tabla de conflictos para el flujo de tareas Γ_i correspondiente al instante crítico iniciado con τ_{ik} para el análisis de la tarea τ_{ab} se puede realizar mediante el Algoritmo 2.2. Durante la construcción de la tabla se puede reducir el número de conflictos, debido a que al estar analizando el instante crítico creado con la tarea τ_{ik} , estamos implícitamente eliminando la posibilidad de algunos conflictos. Las tareas precedidas por τ_{ik} y en conflicto con ella no pueden ejecutarse, ya que hemos elegido la τ_{ik} para ejecutarse, por lo menos en la activación que inicia el instante crítico, con el índice $p'_{0,ikk}$, calculado mediante:

$$p'_{0,ijk} = - \left\lceil \frac{J_{ij} + \varphi'_{ijk}}{T_i} \right\rceil + 1 \quad (2.74)$$

Por lo tanto, en las instancias con índice mayor o igual que $p'_{0,ikk}$, se pueden eliminar activaciones de tareas precedidas por τ_{ik} y no pertenecientes a la misma *H-section* (Algoritmo 2.2 línea 9).

Para resolver los conflictos de activación hay que recorrer la tabla fila a fila (instancia a instancia), sumando en cada fila el máximo valor de tiempo de ejecución de una única *H-section*. El tiempo de ejecución de una *H-section* en una instancia p' se calcula sumando las celdas pertenecientes a las tareas de dicha *H-section*. Por lo tanto, la interferencia del flujo de tareas Γ_i antes del instante crítico ($p' \leq 0$) se calcula mediante el Algoritmo 2.3.

Teniendo en cuenta el Algoritmo 2.3 para la interferencia antes del instante crítico y la Ecuación (2.72) para la interferencia después del instante

Algoritmo 2.2: crear_tabla_de_conflictos_FP.

Input: $\tau_{ab}, t, \Gamma_i, \tau_{ik}$
Result: Tabla

```

1 begin
2   Inicializar_Tabla;
3   for  $\tau_{ij} \in hp_i(\tau_{ab})$  do
4     for  $p' \in p'_{0,ijk}, 0$  do
5       if  $t \geq \varphi'_{ijk} - (p' - 1)T_i$  then
6         Tabla[j, p'] :=  $C_{ij}$ ;
7       else
8         Tabla[j, p'] := 0;
9       if  $p' \geq p'_{0,ikk}$  and  $j > k$  and  $H_{ij}(\tau_{ab}) \neq H_{ik}(\tau_{ab})$  then
10        Tabla[j, p'] := 0;
```

Algoritmo 2.3: resolver_conflictos_FP.

Input: $\tau_{ab}, t, \Gamma_i, \tau_{ik}$
Result: tiempo_interferencia

```

1 begin
2   crear_tabla_de_conflictos_FP( $\tau_{ab}, t, \Gamma_i, \tau_{ik}, Tabla$ );
3   Total := 0;
4   for  $p' \in p'_{0,iNk}, 0$  do
5     max_section := 0;
6     sum := 0 ;
7     for  $\tau_{ij} \in \Gamma_i$  do
8       if  $prio(\tau_{ij}) < prio(\tau_{ik})$  and  $proc(\tau_{ij}) = proc(\tau_{ab})$  then
9         sum := 0;
10      else
11        sum := sum + Tabla[j, p'];
12      if sum > max_section then
13        max_section := sum;
14    Total := Total + max_section ;
```

crítico, la interferencia total se calcula sumando ambas interferencias:

$$W_{ik}(\tau_{ab}, t) = \text{resolver_conflictos_FP}(\tau_{ab}, t, \Gamma_i, \tau_{ik}) + \sum_{\forall j \in MP_i(\tau_{ab})} \left\lceil \frac{t - \phi'_{ijk}}{T_i} \right\rceil_0 C_{ij} \quad (2.75)$$

En esta técnica, al igual que en la técnica de *offsets*, se puede obtener un límite superior de la interferencia de Γ_i :

$$W_i^*(\tau_{ab}, t) = \max_{\forall k \in XP_i(\tau_{ab})} W_{ik}(\tau_{ab}, t) \quad (2.76)$$

Debido a que dos tareas consecutivas con prioridad mayor que τ_{ab} y que se ejecutan en el mismo procesador no pueden pertenecer a periodos de ocupación diferentes, los casos de análisis de la Ecuación (2.76) se reducen al conjunto:

$$XP_i(\tau_{ab}) = \left\{ l \in \Gamma_i \mid \begin{array}{l} \text{proc}(\tau_{il}) = \text{proc}(\tau_{ab}) \wedge \text{prio}(\tau_{il}) \geq \text{prio}(\tau_{ab}) \wedge \\ \text{proc}(\tau_{il-1}) \neq \text{proc}(\tau_{ab}) \vee \text{prio}(\tau_{il-1}) \geq \text{prio}(\tau_{ab}) \end{array} \right\} \quad (2.77)$$

En cuanto al análisis de la interferencia de las instancias del flujo de tareas Γ_a en el que se encuentra la tarea bajo análisis τ_{ab} , se utiliza el mismo algoritmo para la creación de la tabla de conflictos (Algoritmo 2.2), pero en este caso se le pueden aplicar las siguientes reglas de reducción:

1ª regla de reducción (Debido a la creación del instante crítico). Debido a que el instante crítico se ha creado con la tarea τ_{ac} , la activación $p'_{0,acc}$ de dicha tarea debe estar en el periodo de ocupación, y eso provoca que las activaciones iguales o posteriores a $p'_{0,acc}$ de otras tareas en conflicto con τ_{ac} pueden ser eliminadas. Esto significa que, se pueden eliminar las celdas (j, p') que verifican la condición:

$$p' \geq p'_{0,acc} \wedge j > c \wedge H_{aj}(\tau_{ab}) \neq H_{ac}(\tau_{ab}) \quad (2.78)$$

2ª regla de reducción (Debido a la ejecución de la instancia p'_{ab} de τ_{ab} en el periodo de ocupación). Debido a que se está analizando el tiempo de respuesta de la instancia p'_{ab} de τ_{ab} , se pueden eliminar instancias con índice igual o anterior a p'_{ab} de tareas en conflicto con τ_{ab} . Esto es, las celdas (j, p') que verifican la condición:

$$p' \leq p'_{ab} \wedge j < b \wedge H_{aj}(\tau_{ab}) \neq H_{ab}(\tau_{ab}) \quad (2.79)$$

3ª regla de reducción (Debido al análisis de la instancia p'_{ab} de τ_{ab}). Al analizar la instancia p'_{ab} de τ_{ab} , tareas posteriores en esa misma instancia o en instancias posteriores no pueden contribuir a la interferencia de la instancia p'_{ab} de τ_{ab} . Por lo tanto, se pueden eliminar las celdas (j, p') que verifican la condición:

$$(p' \geq p'_{ab} \wedge j > b) \vee (p' > p'_{ab} \wedge j = b) \quad (2.80)$$

Por lo tanto, el algoritmo de creación de la tabla de conflictos para el cálculo de interferencia de las instancias anteriores al instante crítico del flujo de tareas Γ_a se muestra en el Algoritmo 2.4.

Algoritmo 2.4: crear_tabla_de_conflictos_en_ Γ_a _FP.

Input: $\tau_{ab}, t, \Gamma_a, \tau_{ac}$
Result: Tabla

```

1 begin
2   Inicializar_Tabla;
3   for  $\tau_{aj} \in hp_a(\tau_{ab})$  do
4     for  $p' \in p'_{0,ajc}, 0$  do
5       if  $t \geq \varphi'_{ajc} - (p' - 1)T_a$  then
6          $Tabla[j, p'] := C_{aj}$ ;
7       else
8          $Tabla[j, p'] := 0$ ;
9       if  $p' \geq p'_{0,acc}$  and  $j > c$  and  $H_{aj}(\tau_{ab}) \neq H_{ac}(\tau_{ab})$  then
10         $Tabla[j, p'] := 0$ ;
11       if  $p' \leq p'_{ab}$  and  $j < b$  and  $H_{aj}(\tau_{ab}) \neq H_{ab}(\tau_{ab})$  then
12         $Tabla[j, p'] := 0$ ;
13       if  $(p' \geq p'_{ab} \wedge j > b)$  or  $(p' > p'_{ab} \wedge j = b)$  then
14         $Tabla[j, p'] := 0$ ;

```

El proceso de resolución de conflictos es el mismo que el mostrado en el Algoritmo 2.3 pero usando el Algoritmo 2.4. Por otra parte, para calcular la interferencia de las instancias posteriores al instante crítico hay que tener en cuenta que solo las tareas pertenecientes a la primera *H-section* pueden contribuir, es decir, las tareas en el conjunto MP_a , obtenido mediante la ecuación (2.71). Tampoco pueden interferir instancias de tareas posteriores a τ_{ab} en activaciones igual o posteriores a p'_{ab} . Por todo ello la interferencia

después del instante crítico es:

$$W_{ac}(\tau_{ab}, t) |_{p'_{ab}>0} = \sum_{\substack{\forall j \in MP_a(\tau_{ab}) \\ j < b}} \left[\frac{t - \varphi'_{ajc}}{T_a} \right]_0 C_{aj} +$$

$$\max \left\{ 0, p'_{ab} C_{ab} + \sum_{\substack{\forall j \in MP_a(\tau_{ab}) \\ j > b}} \min \left(p'_{ab} - 1, \left[\frac{t - \varphi'_{ajc}}{T_a} \right]_0 \right) C_{aj} \right\} \quad (2.81)$$

Y por lo tanto la interferencia total generada por el flujo de tareas Γ_a al periodo de ocupación para la activación p'_{ab} de la tarea τ_{ab} es:

$$W_{ac}(\tau_{ab}, p'_{ab}, t) = \text{resolver_conflictos_en_}\Gamma_a_FP(\tau_{ab}, p'_{ab}, t, \Gamma_a, \tau_{ac}) +$$

$$+ W_{ac}(\tau_{ab}, t) |_{p'_{ab}>0} \quad (2.82)$$

A partir de estas ecuaciones, el tiempo de finalización $w_{abc}(p'_{ab})$ de la activación p'_{ab} se calcula mediante:

$$w_{abc}(p'_{ab}) = B_{ab} + W_{ac}(\tau_{ab}, p'_{ab}, w_{abc}(p'_{ab})) + \sum_{\forall i \neq a} W_i^*(\tau_{ab}, w_{abc}(p'_{ab})) \quad (2.83)$$

donde $W_i^*(\tau_{ab}, w_{abc}(p'_{ab}))$ se calcula mediante la ecuación (2.76). Teniendo en cuenta este resultado se puede calcular el tiempo de respuesta de peor caso de dicha activación:

$$R_{abc}(p'_{ab}) = w_{abc}(p'_{ab}) - \varphi_{abc} - (p'_{ab} - 1)T_a + \phi_{ab} \quad (2.84)$$

El análisis debe aplicarse a todas las activaciones que se dan en el periodo de ocupación L_{abc} , calculado mediante:

$$L_{abc} = B_{ab} + W_{ac}(\tau_{ab}, L_{abc}) + \sum_{\forall i \neq a} W_i^*(\tau_{ab}, L_{abc}) \quad (2.85)$$

Por lo que las activaciones a analizar van de $p'_{0,abc}$ (ecuación (2.74)) y $p'_{L,abc}$:

$$p'_{L,abc} = - \left[\frac{L_{abc} - \varphi'_{abc}}{T_a} \right]_0 \quad (2.86)$$

Con lo que el tiempo de respuesta de peor caso para la tarea τ_{ab} es:

$$R_{ab} = \max_{\forall c \in XP_a(\tau_{ab})} \left[\max_{p=p'_{0,abc} \cdot p'_{L,abc}} (R_{abc}(p)) \right] \quad (2.87)$$

Posteriormente, Redell [Red04] desarrolló una técnica que, además de ampliar el modelo lineal de flujos de tareas a un modelo en árbol, reducía el pesimismo en el cálculo de los tiempos de respuesta de peor caso. Para reducir el pesimismo se analizan las relaciones de precedencia más a fondo, teniendo en cuenta algunas relaciones que no se habían tenido en cuenta en el trabajo de Palencia y González Harbour [PGH99].

2.3.6. Método *slanted* con *offsets* dinámicos para FP

Este método presentado por Mäki-Turja y Nolin [MTN08] se basaba en la técnica de *offset* de Palencia y González Harbour presentada en la sección 2.3.3. La mejora se basa en definir una nueva función de estimación de interferencia con una aproximación más estrecha. Este método proporciona mejores resultados que el análisis aproximado basado en *offsets*, pero es desconocido si ofrece mejores resultados que el método con relaciones de precedencia. En general ambas técnicas deben ser aplicadas, y utilizar los mejores resultados como límite superior en los tiempos de respuesta, ya que ambos métodos son pesimistas. Esta técnica no se utilizará en esta tesis.

Capítulo 3

Demostración de la convergencia del análisis holístico en sistemas distribuidos con planificación EDF

RESUMEN: En este capítulo se demuestra la convergencia del análisis de tiempos de respuesta para sistemas distribuidos presentado por Spuri en [Spu96b] y resumido en la sección 2.3.2 y se extiende el algoritmo para el caso del análisis de sistemas con recursos compartidos. Esta convergencia no ha sido demostrada con anterioridad, pese a la importancia que tiene ya que es la base de muchos algoritmos de análisis presentados en la literatura, así como de los presentados posteriormente en esta tesis. En primer lugar, en la sección 3.1, se explica la extensión que se hace al análisis de Spuri. Posteriormente, en la sección 3.2, se analiza el posible problema de convergencia en el análisis para sistemas distribuidos mediante este algoritmo. En la sección 3.3, se detalla la demostración de la convergencia del algoritmo. Por último, en la sección 3.4, aplicaremos el algoritmo analizado a un ejemplo de un sistema industrial y, en la sección 3.5, se analizan las conclusiones obtenidas en este capítulo.

3.1. Extensión al algoritmo de Spuri

La extensión que se hace en este trabajo al algoritmo de Spuri, se centra en la asignación de techos de prioridad a los recursos compartidos. En el trabajo de Spuri se propone una asignación estática reflejada en la ecuación (2.37) de la sección 2.3.2, basada en el plazo efectivo de las tareas, es decir, la diferencia entre el plazo y el *jitter*. Pero, no se hace referencia a que la asignación debe ser dinámica, ya que el *jitter* varía en cada iteración del algoritmo. Por lo que se propone que el algoritmo de tiempo de respuesta pase a ser el siguiente:

Algoritmo 3.1: Extensión del algoritmo holístico para análisis de tiempos de respuesta de peor caso para EDF.

```
1 begin
2   Inicializar jitters;
3   repeat
4     por cada tarea, calcular los niveles de expulsión y los techos de
       los recursos;
5     por cada tarea, calcular el tiempo de respuesta de peor caso;
6     por cada tarea, calcular jitter;
7   until alcanzada una solución estable o los plazos no se cumplen;
```

Por lo tanto, los techos de los recursos cambian en cada iteración y los valores válidos, que se deben trasladar al sistema, se obtienen solo cuando se alcanza una solución estable.

3.2. Posible problema de no convergencia

Spuri argumentó la convergencia de su método iterativo en que sus ecuaciones iterativas eran no-decrecientes, es decir, en el progreso iterativo del algoritmo, la estimación de los tiempos de respuesta de peor caso crecen en cada iteración, así como, los *jitter* estimados. Pero esta afirmación no fue probada formalmente. Por lo tanto, es necesario demostrar que cuando el *jitter* de una tarea aumenta, no puede decrecer el tiempo de respuesta de peor caso de ninguna de las tareas del sistema.

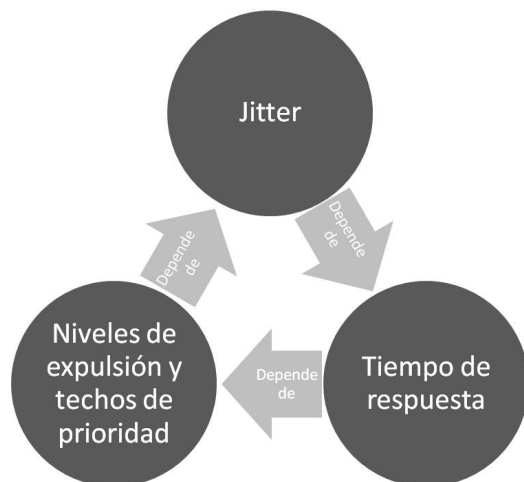


Figura 3.1: Dependencia circular en sistemas distribuidos con uso de recursos compartidos.

Por otra parte, en sistemas con uso de recursos compartidos surge una dependencia circular, mostrada en la Figura 3.1 entre el *jitter*, los niveles de expulsión o los techos de prioridad y los tiempos de respuesta de peor caso. Es decir, si cambia el *jitter* de una tarea, los niveles de expulsión de las tareas y los techos de prioridad de los recursos pueden variar. A su vez un cambio en los techos de prioridad, puede generar un cambio en los tiempos de respuesta de las tareas. Esto podría hacer que un cambio en el *jitter* de una tarea generase un decremento en el tiempo de respuesta de alguna tarea, lo que rompería la convergencia del cálculo. Por lo tanto, hay que demostrar que esto no puede ocurrir.

Por estos motivos, en la siguiente sección se estudiará la convergencia del algoritmo de análisis de tiempos de respuesta de peor caso propuesto por Spuri, tanto en sistemas sin recursos compartidos como con ellos.

3.3. Demostración de la convergencia del análisis

En esta sección se demuestra que la convergencia del método iterativo de Spuri para el cálculo de un límite superior del tiempo de respuesta de

peor caso para tareas planificadas mediante EDF en sistemas distribuidos de tiempo real es correcta. Como hemos comentado antes es necesario demostrarlo tanto con recursos compartidos como sin ellos. Para facilitar el proceso de demostración empezaremos con un sistema muy simple de tres tareas con periodos infinitos y sin recursos compartidos. Posteriormente, extenderemos el sistema añadiendo un número de tareas ilimitado. El tercer paso será ampliar la demostración a un sistema con recursos compartidos. Y, por último, eliminaremos la restricción de los periodos infinitos.

3.3.1. Sistema sin recursos compartidos

Como ya se ha comentado, la demostración para un sistema sin recursos se ha dividido en dos pasos para facilitar la comprensión del lector y proporcionar algunas nociones sobre el mecanismo de demostración. En el primer paso nos centraremos en un sistema de tres tareas. Para en un paso posterior, generalizar el sistema con un número n de tareas ilimitado.

3.3.1.1. Sistema de tres tareas

En primer lugar supondremos un sistema con tres tareas sin recursos compartidos. Las tareas están numeradas en función del plazo efectivo, por lo tanto, $D_1 - J_1 \leq D_2 - J_2 \leq D_3 - J_3$. En este primer paso demostraremos que el tiempo de respuesta de peor caso de estas tareas no puede decrecer debido a un incremento en el *jitter* de una de ellas. Para ello, analizaremos el tiempo de respuesta de peor caso de todas las tareas en dos escenarios. El primer escenario corresponde a la situación anterior a incrementar el *jitter*. Posteriormente, crearemos el segundo escenario incrementando el *jitter* de la tarea τ_3 , hasta un nuevo valor J'_3 ; de forma que su plazo efectivo se encuentre entre el de las tareas τ_1 y τ_2 : $D_1 - J_1 \leq D_3 - J'_3 \leq D_2 - J_2$ (Ver Figura 3.2). Este cambio en el orden de expulsión podría aparentemente hacer que el tiempo de respuesta de la tarea τ_3 decreciera, pero vamos a demostrar que esto no es así. Para ello, recalculemos los tiempos de respuesta de las tareas con el nuevo valor del *jitter* y, finalmente, demostraremos que el tiempo de respuesta en el segundo escenario nunca puede ser menor que el obtenido en el primer escenario. Es necesario aplicar este proceso a las tres tareas.

La Figura 3.2 permite comparar visualmente los intervalos de *jitter* (parte con sombreado claro) y del plazo efectivo (parte de sombreado oscuro).

3.3. Demostración de la convergencia del análisis

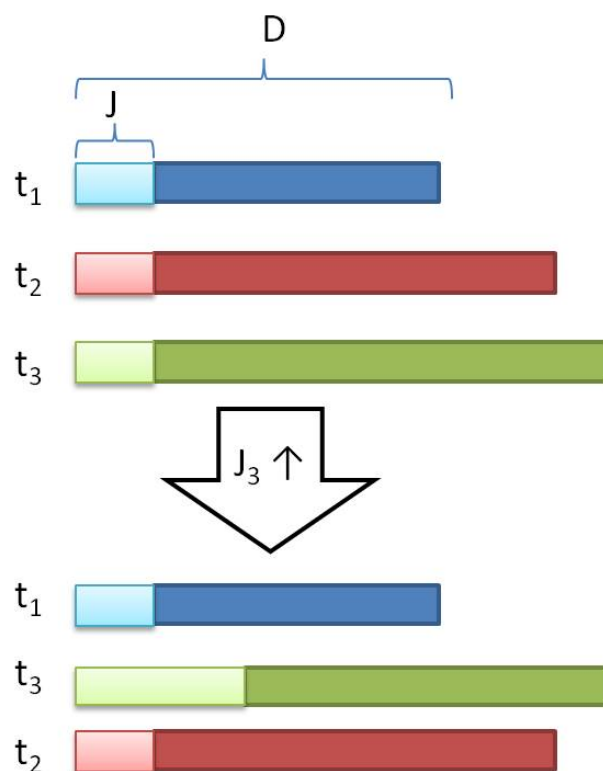


Figura 3.2: Cambio en el orden de expulsión de las tareas debido al aumento del *jitter* de la tarea τ_3 .

La suma de ambas partes nos daría el plazo total de la tarea. Como hemos comentado antes, en este primer paso los periodos se consideran infinitos, por lo que, cada tarea solo tiene una activación o instancia para analizar. Posteriormente, en la sección 3.3.3 analizaremos qué ocurre cuando los periodos son menores.

Teorema 3.1. *Supongamos que τ es un grupo de tres tareas con periodos infinitos, sin recursos compartidos y planificadas bajo EDF. El tiempo de respuesta de cualquiera de las tareas en τ no puede decrecer por el efecto de incrementar el jitter de cualquiera de las tareas en τ .*

Demostración. Si analizamos el primer escenario, antes de incrementar el

3. Demostración de la convergencia del análisis holístico en sistemas distribuidos con planificación EDF

jitter, tenemos los siguientes atributos para las tareas:

$$\begin{aligned}
 \tau_1 &: C_1, D_1, J_1, T_1 = \infty, \\
 \tau_2 &: C_2, D_2, J_2, T_2 = \infty, \\
 \tau_3 &: C_3, D_3, J_3, T_3 = \infty, \\
 &\text{donde } D_3 - J_3 \geq D_2 - J_2 \geq D_1 - J_1
 \end{aligned} \tag{3.1}$$

En cambio, para el segundo escenario, en el que se ha aumentado el *jitter* de la tarea τ_3 (Ver Figura 3.2), los atributos son los siguientes:

$$\begin{aligned}
 \tau_1 &: C_1, D_1, J_1, T_1 = \infty, \\
 \tau_2 &: C_2, D_2, J_2, T_2 = \infty, \\
 \tau_3 &: C_3, D_3, J'_3, T_3 = \infty, \\
 &\text{donde } D_2 - J_2 \geq D_3 - J'_3 \geq D_1 - J_1
 \end{aligned} \tag{3.2}$$

Vamos a empezar analizando la tarea τ_1 , luego continuaremos con τ_2 y acabaremos con τ_3 . De acuerdo, con el algoritmo de Spuri para calcular el tiempo de respuesta de peor caso de una tarea debemos analizar dos situaciones diferentes, correspondientes a diferentes tiempos de activación de la tarea bajo análisis en el primer periodo de ocupación. La primera situación se da cuando todas las tareas se activan sincronamente y con su máximo *jitter* ($a = -J_i$, ver Figura 3.3a). La segunda situación se da cuando el plazo de la tarea τ_1 coincide con el plazo de otra tarea. Remarcar que de esta situación se pueden sacar $n - 1$ casos de análisis, siendo n el número de tareas en el mismo procesador que la tarea bajo análisis ($a = D_2 - J_2 - D_1$ y $a = D_3 - J_3 - D_1$, ver Figura 3.3b y Figura 3.3c, respectivamente).

Si aplicamos el algoritmo de Spuri para el cálculo de tiempo de respuesta de peor caso a estas situaciones, obtenemos los siguientes tiempos de respuesta para la tarea τ_1 :

$$r_1(-J_1) = C_1 - (-J_1) \tag{3.3}$$

$$r_1(D_2 - J_2 - D_1) = C_2 + C_1 - (D_2 - J_2 - D_1) \tag{3.4}$$

$$r_1(D_3 - J_3 - D_1) = C_3 + C_2 + C_1 - (D_3 - J_3 - D_1) \tag{3.5}$$

3.3. Demostración de la convergencia del análisis

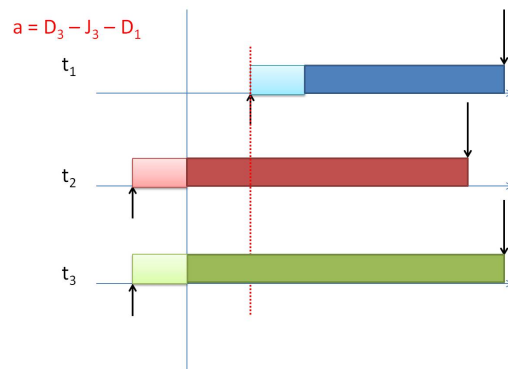
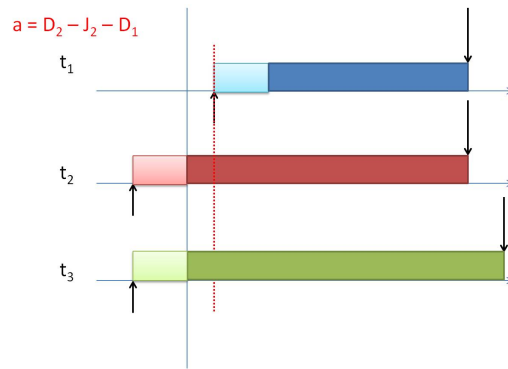
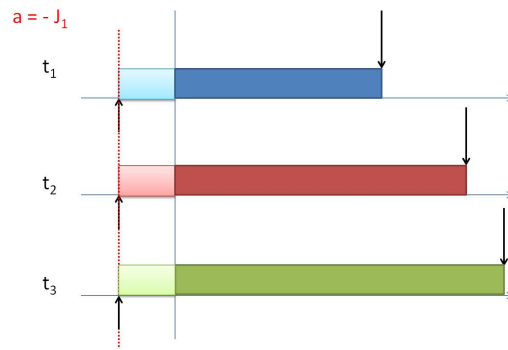


Figura 3.3: Primer escenario de análisis de la tarea τ_1 . (a) Activación síncrona. (b) Plazo síncrono con la tarea τ_2 . (c) Plazo síncrono con la tarea τ_3 .

Consecuentemente, el tiempo de respuesta de peor caso para la tarea τ_1

3. Demostración de la convergencia del análisis holístico en sistemas distribuidos con planificación EDF

es:

$$\begin{aligned}
 r_1 &= \max \{r_1(-J_1), r_1(D_2 - J_2 - D_1), r_1(D_3 - J_3 - D_1)\} \\
 &= \max \{C_1 - (-J_1), C_2 + C_1 - (D_2 - J_2 - D_1), \\
 &\quad C_3 + C_2 + C_1 - (D_3 - J_3 - D_1)\}
 \end{aligned} \tag{3.6}$$

Después de incrementar el *jitter* de la tarea τ_3 , necesitamos analizar el tiempo de respuesta de la tarea τ_1 en tres tiempos de activación diferentes ($a = -J_1$, $a = D_2 - J_2 - D_1$ y $a = D_3 - J'_3 - D_1$, ver Figura 3.4a, Figura 3.4b y Figura 3.4c, respectivamente). Para estas activaciones obtenemos los siguientes tiempos de respuesta:

$$r'_1(-J_1) = C_1 - (-J_1) \tag{3.7}$$

$$r'_1(D_2 - J_2 - D_1) = C_2 + C_3 + C_1 - (D_2 - J_2 - D_1) \tag{3.8}$$

$$r'_1(D_3 - J'_3 - D_1) = C_3 + C_1 - (D_3 - J'_3 - D_1) \tag{3.9}$$

Consecuentemente, el tiempo de respuesta para la tarea τ_1 en esta situación es:

$$\begin{aligned}
 r'_1 &= \max \{r_1(-J_1), r_1(D_2 - J_2 - D_1), r_1(D_3 - J'_3 - D_1)\} \\
 &= \max \{C_1 - (-J_1), C_2 + C_3 + C_1 - (D_2 - J_2 - D_1), \\
 &\quad C_3 + C_1 - (D_3 - J'_3 - D_1)\}
 \end{aligned} \tag{3.10}$$

Para finalizar con la tarea τ_1 , tenemos que demostrar que el tiempo de respuesta de la tarea τ_1 después de aumentar el *jitter* de la tarea τ_3 es mayor que el tiempo de respuesta antes de aumentarlo, es decir, $r'_1 \geq r_1$. Usando las ecuaciones (3.6) y (3.10), hay que probar que:

$$\begin{aligned}
 &\max \{C_1 - (-J_1), C_2 + C_3 + C_1 - (D_2 - J_2 - D_1), \\
 &\quad C_3 + C_1 - (D_3 - J'_3 - D_1)\} \geq \\
 &\max \{C_1 - (-J_1), C_2 + C_1 - (D_2 - J_2 - D_1), \\
 &\quad C_3 + C_2 + C_1 - (D_3 - J_3 - D_1)\}
 \end{aligned} \tag{3.11}$$

Por lo tanto, es necesario demostrar que todos los términos en la función máximo de la derecha de la inecuación tienen un término mayor o igual en el

3.3. Demostración de la convergencia del análisis

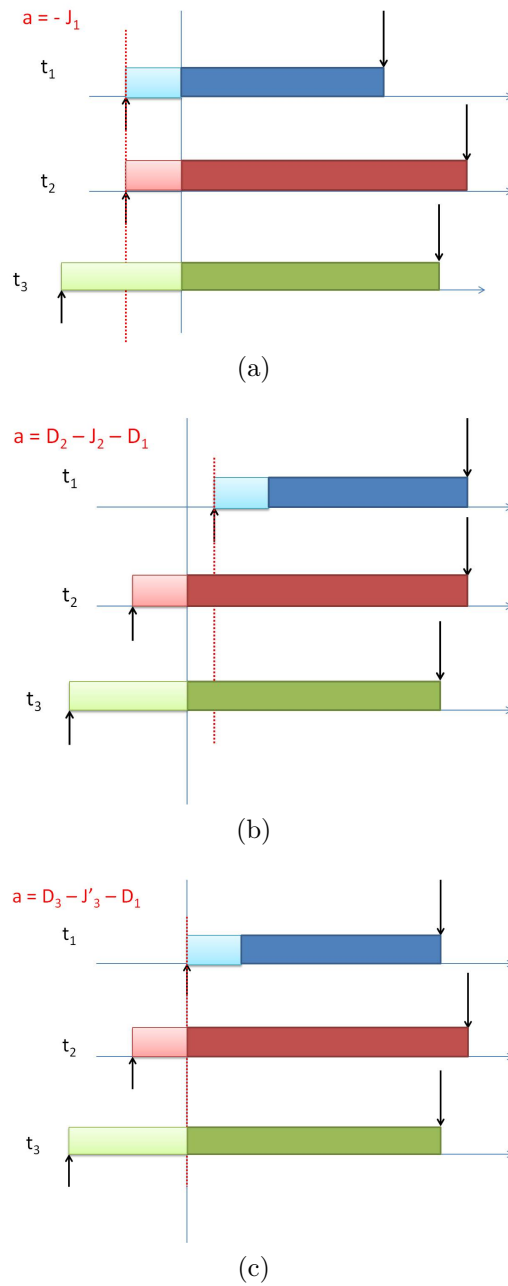


Figura 3.4: Segundo escenario de análisis de la tarea τ_1 . (a) Activación síncrona. (b) Plazo síncrono con la tarea τ_2 . (c) Plazo síncrono con la tarea τ_3 .

lado izquierdo. Consecuentemente se puede dividir la ecuación (3.11) en tres

3. Demostración de la convergencia del análisis holístico en sistemas distribuidos con planificación EDF

inecuaciones:

$$C_1 - (-J_1) \geq C_1 - (-J_1) \quad (3.12)$$

$$C_2 + C_3 + C_1 - (D_2 - J_2 - D_1) \geq C_2 + C_1 - (D_2 - J_2 - D_1) \quad (3.13)$$

$$C_2 + C_3 + C_1 - (D_2 - J_2 - D_1) \geq C_3 + C_2 + C_1 - (D_3 - J_3 - D_1) \quad (3.14)$$

La inecuación (3.12) es trivialmente verdadera, ya que ambos lados son iguales. La inecuación (3.13) es también trivialmente verdadera, ya que el único término que aparece en el lado izquierdo y no en el derecho es C_3 , que es siempre positivo por definición. Resolviendo la inecuación (3.14), se obtiene que $D_2 - J_2 \leq D_3 - J_3$. Debido a la condición inicial en la ecuación (3.1), esta inecuación es verdad.

En este punto, hemos demostrado que el tiempo de respuesta de peor caso de la tarea τ_1 no puede decrecer como consecuencia de un aumento del *jitter* de la tarea τ_3 . Ahora desarrollaremos el mismo proceso para la tarea τ_2 . Primero, analizaremos el tiempo de respuesta de la tarea τ_2 en el primer escenario. En este caso, solo es necesario analizar dos activaciones diferentes, cuando todas las tareas se activan sincronamente con su máximo *jitter* ($a = -J_2$, ver Figura 3.5a), y cuando el plazo de la tarea τ_2 coincide con el plazo de otras tareas, en este caso, τ_3 ($a = D_3 - J_3 - D_2$, ver Figura 3.5b). En este caso, no es necesario analizar el tiempo de respuesta cuando el plazo de la tarea τ_2 coincide con el plazo de la tarea τ_1 , dado que es imposible retrasar la activación de la tarea τ_2 de forma que coincida con el plazo de la tarea τ_1 . Como hemos establecido periodos infinitos para las tareas, la tarea τ_1 tendrá una única instancia, y su plazo será anterior al de la tarea τ_2 .

Si aplicamos el algoritmo de Spuri, explicado en la sección 2.3.2, a estas activaciones, obtenemos estos tiempos de respuesta para la tarea τ_2 :

$$r_2(-J_2) = C_1 + C_2 - (-J_2) \quad (3.15)$$

$$r_2(D_3 - J_3 - D_2) = C_1 + C_2 + C_3 - (D_3 - J_3 - D_2) \quad (3.16)$$

3.3. Demostración de la convergencia del análisis

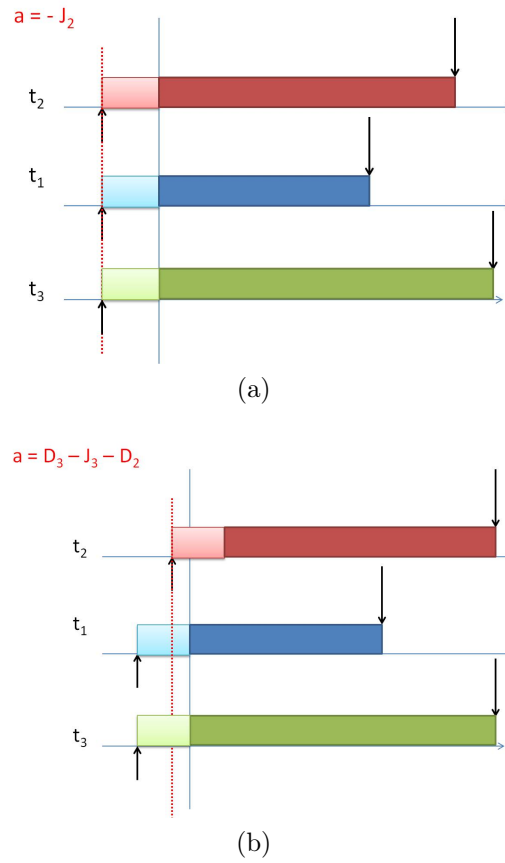


Figura 3.5: Primer escenario de análisis de la tarea τ_2 . (a) Activación síncrona. (b) Plazo síncrono con la tarea τ_3 .

$$\begin{aligned}
 r_2 &= \max \{r_2(-J_2), r_2(D_3 - J_3 - D_2)\} \\
 &= \max \{C_1 + C_2 - (-J_2), C_1 + C_2 + C_3 - (D_3 - J_3 - D_2)\} \quad (3.17)
 \end{aligned}$$

Ahora analizaremos el tiempo de respuesta de la tarea τ_2 en el segundo escenario, es decir, después de incrementar el *jitter* de la tarea τ_3 . En este caso, solo es necesario analizar una situación: cuando todas las tareas se activan síncronamente y con su máximo *jitter* ($a = -J_2$, ver Figura 3.6). La razón es que es imposible retrasar la tarea τ_2 de forma que su plazo coincida con el plazo de las otras tareas.

3. Demostración de la convergencia del análisis holístico en sistemas distribuidos con planificación EDF

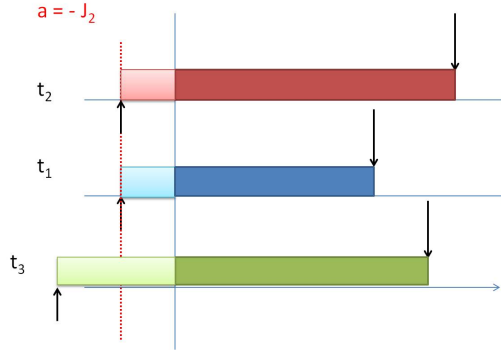


Figura 3.6: Segundo escenario de análisis de la tarea τ_2 , activación síncrona.

El tiempo de respuesta de peor caso que se obtiene en este caso es:

$$r'_2 = r'_2(-J_2) = C_1 + C_3 + C_2 - (-J_2) \quad (3.18)$$

Para finalizar con la tarea τ_2 , se debe demostrar que $r'_2 \geq r_2$. De las ecuaciones (3.17) y (3.18), se obtiene la inecuación a demostrar:

$$C_1 + C_3 + C_2 - (-J_2) \geq \max \{C_1 + C_2 - (-J_2), C_1 + C_2 + C_3 - (D_3 - J_3 - D_2)\} \quad (3.19)$$

Para demostrar que esta inecuación es cierta, debemos verificar que el término en la parte izquierda de la inecuación es mayor o igual que todos los términos de la parte derecha:

$$C_1 + C_3 + C_2 - (-J_2) \geq C_1 + C_2 - (-J_2) \quad (3.20)$$

$$C_1 + C_3 + C_2 - (-J_2) \geq C_1 + C_2 + C_3 - (D_3 - J_3 - D_2) \quad (3.21)$$

Por una parte, si resolvemos la inecuación (3.20) obtenemos $C_3 \geq 0$, lo que es trivialmente cierto ya que C_3 es positivo por definición. Por otra parte, resolviendo la inecuación (3.21) obtenemos que $D_2 - J_2 \leq D_3 - J_3$. Como se ha mencionado antes, la condición establecida en la inecuación (3.1), que

3.3. Demostración de la convergencia del análisis

establece las características del grupo de tareas, hace que esta inecuación sea cierta.

En este punto, hemos demostrado que las tareas τ_1 y τ_2 no pueden ver decrecido su tiempo de respuesta de peor caso por el hecho de incrementar el *jitter* de la tarea τ_3 . Para finalizar, analizaremos la tarea τ_3 . Primero, analizaremos el tiempo de respuesta de la tarea τ_3 en el primer escenario. En este caso, solo es necesario analizar una activación: cuando todas las tareas se activan sincronamente y con su máximo *jitter* ($a = -J_3$, ver Figura 3.7). No es necesario analizar el tiempo de respuesta cuando el plazo de la tarea τ_3 coincide con el plazo de otras tareas, porque es imposible retrasar la tarea τ_3 para que se dé esa situación.

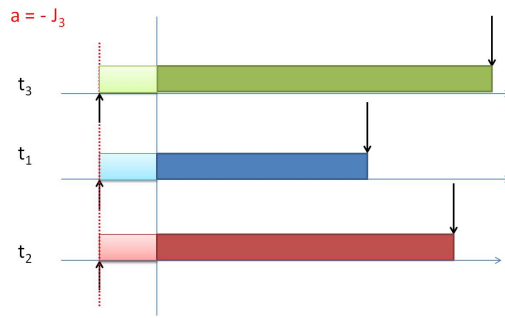


Figura 3.7: Primer escenario de análisis de la tarea τ_3 , activación sincrona.

Si aplicamos el análisis de Spuri para el cálculo de tiempos de respuesta obtenemos el siguiente tiempo para la tarea τ_3 :

$$r_3 = r_3(-J_3) = C_1 + C_2 + C_3 - (-J_3) \quad (3.22)$$

Ahora analizaremos el tiempo de respuesta de la tarea τ_3 en el segundo escenario, después de incrementar su propio *jitter*. En este caso, es necesario analizar dos situaciones: cuando todas las tareas se activan sincronamente y con su máximo *jitter* ($a = -J'_3$, ver Figura 3.8a); y cuando el plazo de la tarea τ_3 coincide con el de la tarea τ_2 ($a = D_2 - J_2 - D_3$, ver Figura 3.8b).

Si analizamos esta situación, obtenemos el siguiente tiempo de respuesta:

$$\begin{aligned} r'_3 &= \max \{r'_3(-J'_3), r'_3(D_2 - J_2 - D_3)\} \\ &= \max \{C_1 + C_3 - (-J'_3), C_1 + C_2 + C_3 - (D_2 - J_2 - D_3)\} \end{aligned} \quad (3.23)$$

3. Demostración de la convergencia del análisis holístico en sistemas distribuidos con planificación EDF

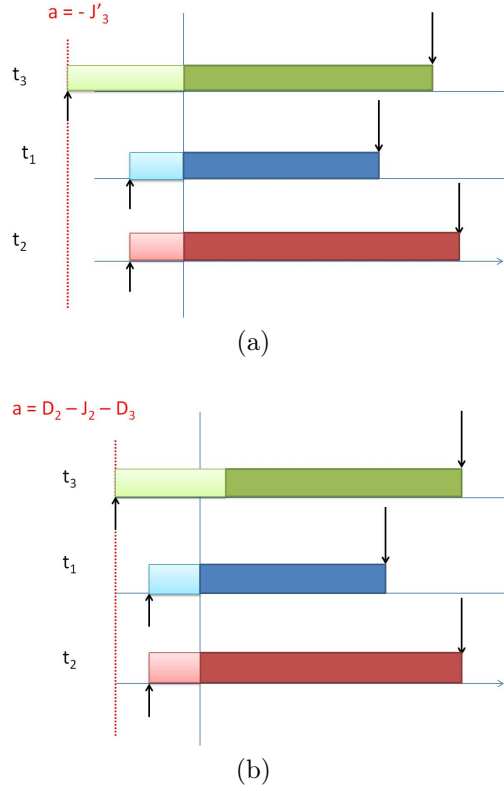


Figura 3.8: Segundo escenario de análisis de la tarea τ_3 . (a) Activación síncrona. (b) Plazo síncrono con la tarea τ_2 .

Para finalizar con la tarea τ_3 , hay que demostrar que $r'_3 \geq r_3$. A partir de las ecuaciones (3.22) y (3.23), obtenemos la siguiente inecuación:

$$\max \{C_1 + C_3 - (-J'_3), C_1 + C_2 + C_3 - (D_2 - J_2 - D_3)\} \geq C_1 + C_2 + C_3 - (-J_3) \quad (3.24)$$

Para demostrar que esta inecuación es correcta necesitamos verificar que hay por lo menos un término en la parte izquierda de la inecuación que es mayor o igual que el término en el lado derecho:

$$C_1 + C_2 + C_3 - (D_2 - J_2 - D_3) \geq C_1 + C_2 + C_3 - (-J_3) \quad (3.25)$$

Resolviendo la inecuación (3.25) obtenemos que $D_2 - J_2 \leq D_3 - J_3$. Como se ha mencionado antes, la condición en la inecuación (3.1) hace que esto sea cierto.

3.3. Demostración de la convergencia del análisis

Como consecuencia, hemos evidenciado que, si el *jitter* de la tarea τ_3 , J_3 , es incrementado, el tiempo de respuesta de peor caso de las tareas no decrementa su valor. Podemos extender esta afirmación al incremento de J_1 o J_2 . Las tareas que inicialmente están por debajo de la tarea cuyo *jitter* es incrementado en el orden de plazo efectivo, mantienen la misma posición en el orden de plazo efectivo después del incremento de *jitter* y, por lo tanto, su tiempo de respuesta de peor caso no se verá afectado. Como ejemplo, supongamos el grupo de tareas anterior pero con una cuarta tarea τ_4 , de forma que $D_3 - J_3 \leq D_4 - J_4$. Si aplicamos el algoritmo holístico de Spuri para calcular el tiempo de respuesta de peor caso antes y después de incrementar J_3 , obtenemos el mismo valor para el tiempo de respuesta de τ_4 , es decir, $r_4 = r'_4$.

En este punto, hemos demostrado que tres tareas con periodos infinitos y sin recursos compartidos no pueden ver decrecido su tiempo de respuesta de peor caso debido al incremento de sus *jitters*. \square

3.3.1.2. Sistema con un número de tareas indefinido

En esta sección vamos a generalizar el análisis realizado en la sección anterior para cubrir un sistema con cualquier número de tareas.

Teorema 3.2. *Tomando τ como un grupo de tareas con periodo infinito y sin recursos compartidos, planificadas bajo EDF, el tiempo de respuesta de cualquier tarea en τ no puede decrecer por el hecho de aumentar el jitter de cualquiera de las tareas en τ .*

Demostración. Para demostrar este teorema, supongamos un sistema con $n + m + k$ tareas divididas en 3 grupos que se verán afectados de diferente manera por el incremento del *jitter* de una tarea. En la Figura 3.9a, se puede ver que las tareas están ordenadas por plazo efectivo:

$$D_i - J_i \leq D_{i+1} - J_{i+1} \quad (3.26)$$

Como se puede observar en la Figura 3.9, las tareas están divididas en diferentes grupos: Grupo 1 (tareas desde τ_1 hasta τ_n que tienen su plazo efectivo siempre por debajo del resto de grupos, tanto en la primera situación como en la segunda), Grupo 2 (tareas desde τ_{n+1} hasta τ_{n+m} que inicialmente

3. Demostración de la convergencia del análisis holístico en sistemas distribuidos con planificación EDF

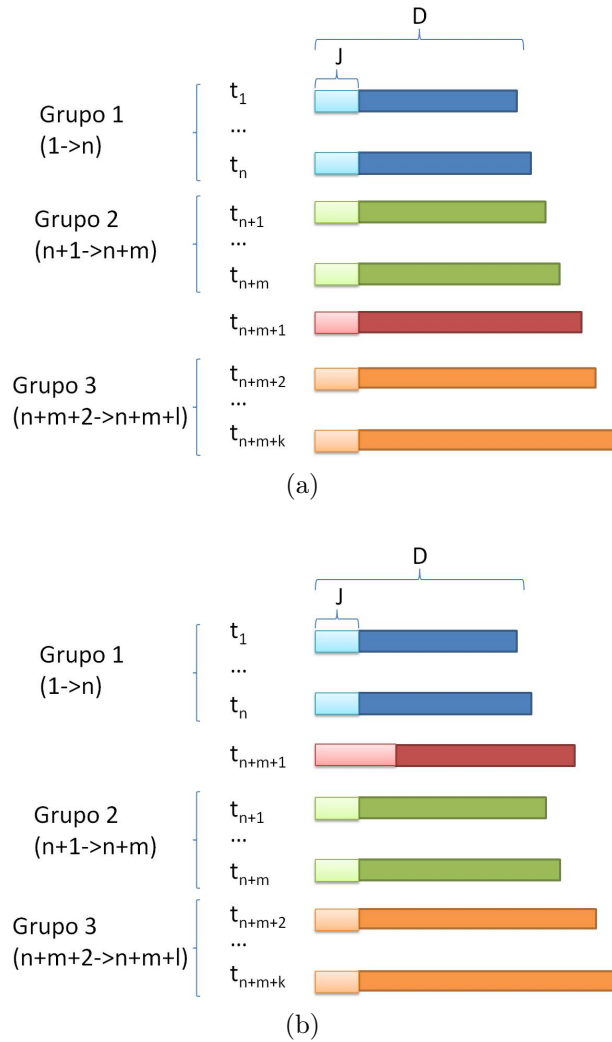


Figura 3.9: Sistema con cualquier número de tareas (a) Características del sistema antes de incrementar el *jitter* de la tarea τ_{n+m+1} . (b) Características del sistema después de incrementar el *jitter* de la tarea τ_{n+m+1} .

tienen el plazo efectivo por debajo de todos los grupos menos el Grupo 1, pero en la segunda situación se convierte en superior al de la tarea τ_{n+m+1} que incrementa su *jitter* en el segundo escenario), Grupo 3(tareas desde τ_{n+m+2} hasta τ_{n+m+k} cuyo plazo efectivo está siempre por encima de las tareas de otros grupos) y, por último, la tarea τ_{n+m+1} es la que incrementa su *jitter*. En la Figura 3.9b, se puede ver cómo el *jitter* de la tarea τ_{n+m+1} se incrementa de forma que el orden de plazos efectivos de las tareas cambia. El orden después

3.3. Demostración de la convergencia del análisis

del incremento del *jitter* de la tarea τ_{n+m+1} es:

$$\begin{aligned} D_z - J_z \leq D_{n+m+1} - J_{n+m+1} \leq D_y - J_y \leq D_x - J_x, \text{ donde} \\ 1 \leq z \leq n, n+1 \leq y \leq n+m, n+m+2 \leq x \leq n+m+k \end{aligned} \quad (3.27)$$

En resumen, la tarea τ_{n+m+1} está por debajo del Grupo 2 en la Figura 3.9a y después está posicionada entre el Grupo 1 y el Grupo 2 en la Figura 3.9b.

Para demostrar el teorema, vamos a probar que cualquier tarea de estos grupos no puede ver su tiempo de respuesta reducido cuando el sistema cambia del escenario mostrado en la Figura 3.9a al mostrado en la Figura 3.9b, esto es, cuando el *jitter* de la tarea τ_{n+m+1} crece. Cuando esto ocurre, el orden de expulsión de las tareas cambia y esto podría suponer una reducción del tiempo de respuesta de peor caso de esta o otra tarea. A continuación demostraremos que esto no puede ocurrir. Para conseguir este objetivo, analizaremos los diferentes grupos por separado, empezando por el Grupo 1 y acabando con el Grupo 3. También tendremos que analizar la tarea que cambia su *jitter*.

Para empezar con el Grupo 1, debemos demostrar el siguiente lema:

Lema 3.1. *Supongamos un grupo τ de $n+m+k$ tareas con periodos infinitos planificados bajo EDF. Asumiendo que el *jitter* de la tarea τ_{n+m+1} es incrementado, para una tarea τ_α en el Grupo 1 ($\tau_1 \leq \tau_\alpha \leq \tau_n$), en el que el plazo efectivo es siempre inferior que el de la tarea τ_{n+m+1} , el tiempo de respuesta de peor caso obtenido mediante el algoritmo de Spuri será igual o mayor que el tiempo de respuesta calculado antes de incrementar el *jitter* de la tarea τ_{n+m+1} .*

Demostración. Primero, analizaremos el tiempo de respuesta de peor caso en el primer escenario (ver Figura 3.9a). Después, incrementaremos el *jitter* de la tarea τ_{n+m+1} y haremos el análisis del segundo escenario (ver Figura 3.9b). Finalmente, compararemos ambos resultados para demostrar que el tiempo de respuesta de peor caso no puede decrecer. En el primer escenario, las activaciones a comprobar ocurren cuando todas las tareas se activan sincronamente con su máximo *jitter* y cuando el plazo de la tarea τ_α coincide con el plazo de alguna de las tareas con plazo efectivo más grande. En el caso de la activación sincrónica, la tarea bajo análisis, τ_α , es activada al principio del periodo de ocupación habiendo sufrido su máximo *jitter*. El grupo de activaciones a analizar, incluyendo el caso de activación sincrónica,

3. Demostración de la convergencia del análisis holístico en sistemas distribuidos con planificación EDF

es: $\Psi = \cup(D_i - J_i - D_\alpha) \vee i \in n + m + k \quad \& \quad D_i - J_i \geq D_\alpha - J_\alpha$. Vamos a separar Ψ en cuatro grupos dependiendo de a qué grupo pertenece τ_i : A1 cuando $\alpha \leq i \leq n$, A2 cuando $n + 1 \leq i \leq n + m$, A3 cuando $i = n + m + 1$ y A4 cuando $n + m + 2 \leq i \leq n + m + k$. Para cada caso, definiremos un tiempo de activación a como $a(i) = D_i - J_i - D_\alpha$. Por lo tanto, si aplicamos el algoritmo holístico de Spuri, explicado en la sección 2.3.2, a estos casos de activación, obtenemos los siguientes tiempos de respuesta:

$$A1 : r_\alpha(a(i)) = \left(\sum_{j=1}^i C_j \right) - (D_i - J_i - D_\alpha) \quad (3.28)$$

$$A2 : r_\alpha(a(i)) = \left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^i C_j \right) - (D_i - J_i - D_\alpha) \quad (3.29)$$

$$A3 : r_\alpha(a(i)) = \left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^{n+m} C_j \right) + C_{n+m+1} - (D_{n+m+1} - J_{n+m+1} - D_\alpha) \quad (3.30)$$

$$A4 : r_\alpha(a(i)) = \left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^{n+m} C_j \right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^i C_j \right) - (D_i - J_i - D_\alpha) \quad (3.31)$$

En la segunda situación, después de haber incrementado el *jitter* de la tarea τ_{n+m+1} ($J_{n+m+1} \rightarrow J'_{n+m+1}$), los casos a analizar son los mismos que en la primera situación. La razón es que la posición del Grupo 1 en el orden de plazo efectivo sigue siendo la misma. En este caso, utilizaremos la letra B para diferenciar los casos: B1 cuando $\alpha \leq i \leq n$, B2 cuando $n + 1 \leq i \leq n + m$, B3 cuando $i = n + m + 1$ y B4 cuando $n + m + 2 \leq i \leq n + m + k$. Los tiempos de respuesta son:

$$B1 : r_\alpha(a(i)) = \left(\sum_{j=1}^i C_j \right) - (D_i - J_i - D_\alpha) \quad (3.32)$$

$$B2 : r_\alpha(a(i)) = \left(\sum_{j=1}^n C_j \right) + C_{n+m+1} + \left(\sum_{j=n+1}^i C_j \right) - (D_i - J_i - D_\alpha) \quad (3.33)$$

$$B3 : r_\alpha(a(i)) = \left(\sum_{j=1}^n C_j \right) + C_{n+m+1} - (D_{n+m+1} - J'_{n+m+1} - D_\alpha) \quad (3.34)$$

$$B4 : r_\alpha(a(i)) = \left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^{n+m} C_j \right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^i C_j \right) - (D_i - J_i - D_\alpha) \quad (3.35)$$

Ahora, demostraremos que el tiempo de respuesta de la tarea τ_α en el Grupo 1 no puede ser más grande en el primer escenario que en el segundo. Por lo tanto, la siguiente inecuación tiene que ser cierta:

$$\max \{r(A1) \cup r(A2) \cup r(A3) \cup r(A4)\} \leq \max \{r(B1) \cup r(B2) \cup r(B3) \cup r(B4)\}. \quad (3.36)$$

donde $r(X)$ es el conjunto de tiempos de respuesta para el grupo de activaciones X .

Es suficiente probar que cada término en el lado izquierdo de la inecuación hay un término igual o mayor en el lado derecho. De esta forma, podemos partir la inecuación en cuatro diferentes:

$$\begin{aligned} r(B1) &\geq r(A1) \\ \left(\sum_{j=1}^i C_j \right) - (D_i - J_i - D_\alpha), (\alpha \leq i \leq n) &\geq \\ \left(\sum_{j=1}^i C_j \right) - (D_i - J_i - D_\alpha), (\alpha \leq i \leq n) & \end{aligned} \quad (3.37)$$

$$\begin{aligned} r(B2) &\geq r(A2) \\ \left(\sum_{j=1}^n C_j \right) + C_{n+m+1} + \left(\sum_{j=n+1}^i C_j \right) - (D_i - J_i - D_\alpha), (n \leq i \leq n+m) &\geq \\ \left(\sum_{j=1}^n C_j \right) + C_{n+m+1} + \left(\sum_{j=n+1}^i C_j \right) - (D_i - J_i - D_\alpha), (n \leq i \leq n+m) & \end{aligned} \quad (3.38)$$

3. Demostración de la convergencia del análisis holístico en sistemas distribuidos con planificación EDF

$$\begin{aligned}
 r(B2) &\geq r(A3) \\
 &\left(\sum_{j=1}^n C_j \right) + C_{n+m+1} + \left(\sum_{j=n+1}^i C_j \right) - (D_i - J_i - D_\alpha), (n \leq i \leq n+m) \geq \\
 &\left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^{n+m} C_j \right) + C_{n+m+1} - (D_{n+m+1} - J_{n+m+1} - D_\alpha)
 \end{aligned} \tag{3.39}$$

$$\begin{aligned}
 r(B4) &\geq r(A4) \\
 &\left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^n C_j \right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^i C_j \right) - \\
 &(D_i - J_i - D_\alpha), (n+m+2 \leq i \leq n+m+k) \geq \\
 &\left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^n C_j \right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^i C_j \right) - \\
 &(D_i - J_i - D_\alpha), (n+m+2 \leq i \leq n+m+k)
 \end{aligned} \tag{3.40}$$

Las inecuaciones (3.37) y (3.40) son siempre verdad ya que ambos lados de la inecuación tienen los mismos términos. De la inecuación (3.38), se puede obtener $C_{n+m+1} \geq 0$, lo que es siempre verdad, ya que el tiempo de ejecución de una tarea es positivo por definición. Finalmente, en la inecuación (3.39), si cogemos $i = n + m$, se obtiene $D_{n+m} - J_{n+m} \leq D_{n+m+1} - J_{n+m+1}$. Esto es siempre verdad debido a la condición establecida en la inecuación (3.26). Como hemos dicho es suficiente que haya un caso que haga cumplir la inecuación. \square

En segundo lugar, repetiremos el análisis con el Grupo 2 demostrando el siguiente lema:

Lema 3.2. *Supongamos un grupo τ de $n+m+k$ tareas con periodos infinitos planificados bajo EDF. Asumiendo que el jitter de la tarea τ_{n+m+1} es incrementado, para una tarea τ_α en el Grupo 2 ($\tau_{n+1} \leq \tau_\alpha \leq \tau_{n+m}$), tareas cuyo plazo efectivo en el primer escenario es menor que el de la tarea cuyo jitter se incrementa (τ_{n+m+1}), pero en el segundo escenario es superior, el tiempo de respuesta de peor caso obtenido mediante el algoritmo de Spuri será igual o mayor que el tiempo de respuesta calculado antes de incrementar el jitter de la tarea τ_{n+m+1} .*

3.3. Demostración de la convergencia del análisis

Demostración. En el primer escenario, los casos de activación a analizar son: A1 cuando $\alpha \leq i \leq n+m$, A2 cuando $i = n+m+1$ y A3 cuando $n+m+2 \leq i \leq n+m+k$. Por lo tanto, si aplicamos el algoritmo holístico de Spuri, descrito en la sección 2.3.2, a estos grupos de activaciones, obtenemos los siguientes tiempos de respuesta:

$$A1 : r_\alpha(a(i)) = \left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^i C_j \right) - (D_i - J_i - D_\alpha) \quad (3.41)$$

$$A2 : r_\alpha(a(i)) = \left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^{n+m} C_j \right) + C_{n+m+1} - (D_{n+m+1} - J_{n+m+1} - D_\alpha) \quad (3.42)$$

$$A3 : r_\alpha(a(i)) = \left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^{n+m} C_j \right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^i C_j \right) - (D_{n+m+1} - J_{n+m+1} - D_\alpha) \quad (3.43)$$

En el segundo escenario, después de incrementar el *jitter* de la tarea τ_{n+m+1} ($J_{n+m+1} \rightarrow J'_{n+m+1}$), los casos de activación a analizar son solo dos. Esto se debe a que la posición del Grupo 2 en el orden de plazo efectivo ha cambiado. En este caso, usaremos la letra B para diferenciarlos: B1 cuando $\alpha \leq i \leq n+m$ y B2 cuando $n+m+2 \leq i \leq n+m+k$. Los tiempos de respuesta son:

$$B1 : r_\alpha(a(i)) = \left(\sum_{j=1}^n C_j \right) + C_{n+m+1} + \left(\sum_{j=n+1}^i C_j \right) - (D_i - J_i - D_\alpha) \quad (3.44)$$

$$B2 : r_\alpha(a(i)) = \left(\sum_{j=1}^n C_j \right) + C_{n+m+1} + \left(\sum_{j=n+1}^{n+m} C_j \right) + \left(\sum_{j=n+m+2}^i C_j \right) - (D_i - J_i - D_\alpha) \quad (3.45)$$

Ahora, tenemos que demostrar que el tiempo de respuesta de la tarea $\tau_\alpha \in (n+1 \leq \alpha \leq n+m)$, no puede ser más grande en el primer escenario

3. Demostración de la convergencia del análisis holístico en sistemas distribuidos con planificación EDF

que en el segundo. Por lo tanto, la siguiente inecuación tiene que ser verdad:

$$\max \{r(A1) \cup r(A2) \cup r(A3)\} \leq \max \{r(B1) \cup r(B2)\}. \quad (3.46)$$

Es suficiente con probar que para cada caso en la parte izquierda de la inecuación tiene uno en la parte derecha que es mayor o igual. De esta manera, se puede dividir la inecuación en tres:

$$\begin{aligned} r(B1) &\geq r(A1) \\ &\left(\sum_{j=1}^n C_j \right) + C_{n+m+1} + \left(\sum_{j=n+1}^i C_j \right) - (D_i - J_i - D_\alpha), (\alpha \leq i \leq n+m) \geq \\ &\left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^i C_j \right) - (D_i - J_i - D_\alpha), (\alpha \leq i \leq n+m) \end{aligned} \quad (3.47)$$

$$\begin{aligned} r(B1) &\geq r(A2) \\ &\left(\sum_{j=1}^n C_j \right) + C_{n+m+1} + \left(\sum_{j=n+1}^i C_j \right) - (D_i - J_i - D_\alpha), (\alpha \leq i \leq n+m) \geq \\ &\left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^{n+m} C_j \right) + C_{n+m+1} - (D_{n+m+1} - J_{n+m+1} - D_\alpha) \end{aligned} \quad (3.48)$$

$$\begin{aligned} r(B2) &\geq r(A3) \\ &\left(\sum_{j=1}^n C_j \right) + C_{n+m+1} + \left(\sum_{j=n+1}^{n+m} C_j \right) + \left(\sum_{j=n+m+2}^i C_j \right) - \\ &(D_i - J_i - D_\alpha), (n+m+2 \leq i \leq n+m+k) \geq \\ &\left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^{n+m} C_j \right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^i C_j \right) - \\ &(D_i - J_i - D_\alpha), (n+m+2 \leq i \leq n+m+k) \end{aligned} \quad (3.49)$$

La inecuación (3.49) es siempre verdad ya que ambos lados tienen los mismos términos. De la inecuación (3.47), se obtiene que $C_{n+m+1} \geq 0$, lo que es verdad, ya que por definición el tiempo de ejecución es siempre positivo. Por último, si en la inecuación (3.48) cogemos $i = n+m$, se obtiene que $D_{n+m} - J_{n+m} \leq D_{n+m+1} - J_{n+m+1}$. Esto es siempre verdad debido a la condición establecida en la ecuación (3.26). \square

3.3. Demostración de la convergencia del análisis

En tercer lugar, repetiremos el proceso con la tarea τ_{n+m+1} demostrando el siguiente lema:

Lema 3.3. *Supongamos un grupo τ de $n+m+k$ tareas con periodos infinitos planificados bajo EDF y una tarea τ_{n+m+1} del sistema que tiene tareas por encima y por debajo en el orden de plazo efectivo. Si el jitter de esta tarea es incrementado, reduciendo su plazo efectivo de forma que adelante a otras tareas en el orden de plazo efectivo, el tiempo de respuesta de peor caso, calculado mediante el algoritmo de Spuri, no puede decrecer.*

Demostración. En el primer escenario, los grupos de casos de activación a analizar son: A1 cuando $i = n+m+1$ y A2 cuando $n+m+2 \leq i \leq n+m+k$. Por lo tanto, si aplicamos el algoritmo holístico de Spuri a estos casos de activación, obtenemos los siguientes tiempos de respuesta:

$$A1 : r_{n+m+1}(a(i)) = \left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^{n+m} C_j \right) + C_{n+m+1} - (-J_{n+m+1}) \quad (3.50)$$

$$A2 : r_{n+m+1}(a(i)) = \left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^{n+m} C_j \right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^{n+m+k} C_j \right) - (D_i - J_i - D_{n+m+1}) \quad (3.51)$$

En el segundo escenario, después de incrementar el *jitter* de la tarea τ_{n+m+1} ($J_{n+m+1} \rightarrow J'_{n+m+1}$), los casos de activación a analizar son tres. El motivo es que la posición de la tarea τ_{n+m+1} en el orden de plazo efectivo ha cambiado. Para diferenciar los casos de activación de los anteriores, utilizaremos la letra B : B1 cuando $n \leq i \leq n+m$, B2 cuando $i = n+m+1$ y B3 cuando $n+m+2 \leq i \leq n+m+k$. Los tiempos de respuesta obtenidos para dichas activaciones son:

$$B1 : r_{n+m+1}(a(i)) = \left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^i C_j \right) + C_{n+m+1} - (D_i - J_i - D_{n+m+1}) \quad (3.52)$$

$$B2 : r_{n+m+1}(a(i)) = \left(\sum_{j=1}^n C_j \right) + C_{n+m+1} - (-J'_{n+m+1}) \quad (3.53)$$

3. Demostración de la convergencia del análisis holístico en sistemas distribuidos con planificación EDF

$$B3 : r_{n+m+1}(a(i)) = \left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^{n+m} C_j \right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^i C_j \right) - (D_i - J_i - D_{n+m+1}) \quad (3.54)$$

Por lo tanto, hay que demostrar que el tiempo de respuesta de la tarea $\tau_\alpha \in (n+1 \leq a \leq n+m)$ no puede ser mayor en el primer escenario que en el segundo. Consecuentemente, la siguiente inecuación tiene que ser verdad:

$$\max \{r(A1) \cup r(A2)\} \leq \max \{r(B1) \cup r(B2) \cup r(B3)\}. \quad (3.55)$$

Es suficiente con probar que cada grupo en la parte izquierda de la inecuación tiene uno igual o mayor en la parte derecha. De esta forma, se puede dividir la inecuación en dos:

$$\begin{aligned} r(B1) &\geq r(A1) \\ \left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^i C_j \right) + C_{n+m+1} - (D_i - J_i - D_{n+m+1}), & \quad (3.56) \\ (n \leq i \leq n+m) &\geq \left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^{n+m} C_j \right) + C_{n+m+1} - (-J_{n+m+1}) \end{aligned}$$

$$\begin{aligned} r(B3) &\geq r(A2) \\ \left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^{n+m} C_j \right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^i C_j \right) - & \\ (D_i - J_i - D_{n+m+1}), (n+m+1 \leq i \leq n+m+k) &\geq \quad (3.57) \\ \left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^{n+m} C_j \right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^i C_j \right) - & \\ (D_i - J_i - D_{n+m+1}), (n+m+1 \leq i \leq n+m+k) & \end{aligned}$$

La inecuación (3.57) es siempre verdad ya que ambas partes de la inecuación tienen los mismos términos. Por otra parte, en la inecuación (3.56), si

3.3. Demostración de la convergencia del análisis

tomamos $i = n + m$, obtenemos que $D_{n+m} - J_{n+m} \leq D_{n+m+1} - J_{n+m+1}$. Esto es siempre verdad debido a la condición inicial establecida en la ecuación (3.26). \square

Por último, solo queda repetir el proceso con el Grupo 3 y demostrar el siguiente lema:

Lema 3.4. *Supongamos un grupo τ de $n+m+k$ tareas con periodos infinitos planificados bajo EDF. Asumiendo que el jitter de la tarea τ_{n+m+1} es incrementado, para una tarea τ_α en el Grupo 3 ($\tau_{n+m+2} \leq \tau_\alpha \leq \tau_{n+m+k}$), tareas cuyo plazo efectivo es siempre superior al de las tareas de otros grupos, el tiempo de respuesta de peor caso obtenido mediante el algoritmo de Spuri será igual o mayor que el tiempo de respuesta calculado antes de incrementar el jitter de la tarea τ_{n+m+1} .*

Demostración. En el primer escenario, solo es necesario analizar un conjunto de activaciones: A1 cuando $n + m + 2 \leq i \leq n + m + k$. Por lo tanto, si aplicamos el algoritmo holístico de Spuri a este conjunto, se obtienen los siguientes tiempos de respuesta:

$$A1 : r_\alpha(a(i)) = \left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^{n+m} C_j \right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^i C_j \right) - (D_i - J_i - D_\alpha) \quad (3.58)$$

En el segundo escenario, después de incrementar el *jitter* de la tarea τ_{n+m+1} ($J_{n+m+1} \rightarrow J'_{n+m+1}$), vuelve a haber un único conjunto de activaciones a analizar. La razón es que el orden del Grupo 3 en el plazo efectivo no ha cambiado. En este segundo escenario utilizaremos la letra B para diferenciar los conjuntos de activaciones de los de la primera situación: B1 cuando $n + m + 2 \leq i \leq n + m + k$. Los tiempos de respuesta que se obtienen son:

$$B1 : r_\alpha(a(i)) = \left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^{n+m} C_j \right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^i C_j \right) - (D_i - J_i - D_\alpha) \quad (3.59)$$

3. Demostración de la convergencia del análisis holístico en sistemas distribuidos con planificación EDF

Por último es necesario demostrar que el tiempo de respuesta de la tarea $\tau_\alpha \in (n+1 \leq \alpha \leq n+m)$, no puede ser mayor en el primer escenario que en el segundo. Por lo tanto, la siguiente inecuación tiene que ser verdad:

$$\max \{r(A1)\} \leq \max \{r(B1)\}. \quad (3.60)$$

Es suficiente con probar que los tiempos de respuesta del conjunto B1 son mayores o iguales que los del conjunto A1:

$$\begin{aligned} r(B1) &\geq r(A1) \\ &\left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^{n+m} C_j \right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^i C_j \right) - \\ &(D_i - J_i - D_\alpha), (n+m+2 \leq i \leq n+m+k) \geq \\ &\left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^{n+m} C_j \right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^i C_j \right) - \\ &(D_i - J_i - D_\alpha), (n+m+2 \leq i \leq n+m+k) \end{aligned} \quad (3.61)$$

La inecuación (3.61) es siempre verdad ya que ambos lados de la inecuación tienen los mismos términos. \square

En conclusión, se ha evidenciado que, si el *jitter* de la tarea τ_{n+m+1} , J_{n+m+1} , se incrementa, el tiempo de respuesta de peor caso del conjunto de tareas del sistema no puede decrementar su valor. Esta afirmación puede ser extendida al incremento del *jitter* de cualquier tarea del conjunto del sistema, debido a que, mediante los grupos de tareas 1,2 y 3, la demostración cubre todos los roles en que se puede encontrar una tarea con respecto a la tarea cuyo *jitter* aumenta. Si hay varias tareas que aumentan su *jitter*, se pueden hacer sucesivos cambios con una tarea en cada uno, y observar que el tiempo de respuesta nunca decrece.

En este punto, se ha demostrado que un sistema con cualquier número de tareas con periodos infinitos y sin recursos compartidos no puede ver el tiempo de respuesta de peor caso de sus tareas decrementado por el hecho de incrementar el *jitter* de las mismas tareas. \square

3.3.2. Sistema con recursos compartidos

En esta sección extenderemos el análisis realizado en la sección anterior para incluir el uso de recursos compartidos, lo que introduce una nueva di-

3.3. Demostración de la convergencia del análisis

mención al problema. La propuesta de Spuri [Spu96a, Spu96b] maneja los recursos compartidos de acuerdo al protocolo SRP [Bak91] y asigna los niveles de expulsión de las tareas π_n y los techos de prioridad de los recursos compartidos $\lceil s \rceil$ en función de:

$$\begin{aligned} \pi_i > \pi_j &\iff D_i - J_i < D_j - J_j \\ \lceil s \rceil &= \max \{ \pi_i : i \text{ may lock } s \} \end{aligned} \tag{3.62}$$

Por lo tanto, los niveles de expulsión de las tareas y los techos de prioridad de los recursos están ordenados por plazo efectivo y, por lo tanto, dependen del *jitter*. Este hecho puede provocar que un incremento en el *jitter* de una tarea puede cambiar el orden de los niveles de expulsión (y/o de los techos de prioridad de los recursos). Este efecto podría suponer un cambio en los tiempos de bloqueo y, como consecuencia, los tiempos de respuesta de las tareas podrían decrecer. Consecuentemente, este efecto podría provocar la no convergencia del análisis iterativo para el tiempo de respuesta en planificadores EDF. Con el objetivo de demostrar que este potencial problema no puede ocurrir nunca, vamos a tomar un sistema con un número indefinido de tareas, como el presentado en la Figura 3.9, pero en este caso las tareas podrán usar un número indeterminado de recursos compartidos locales al procesador donde se ejecutan esas tareas. Como en la sección anterior, demostraremos que el tiempo de respuesta de cada tarea no puede decrecer por el incremento del *jitter* de cualquiera de las tareas.

Teorema 3.3. *Supongamos un grupo τ de tareas con periodos infinitos planificados mediante EDF y que usan recursos compartidos locales bajo los protocolos SRP. El tiempo de respuesta de cualquier tarea en τ no puede decrecer por el efecto de incrementar el jitter de cualquier tarea en τ .*

Demostración. Para probar este lema, seguiremos el mismo proceso que en la sección 3.3.1.2 y como hemos comentado antes partiremos del grupo de tareas representado en la Figura 3.9.

Hemos separado las tareas en tres grupos y una tarea, que será a la que se le incremente el *jitter*: Grupo 1 (tareas desde τ_1 hasta τ_n), Grupo 2 (tareas desde τ_{n+1} hasta τ_{n+m}), Grupo 3 (tareas desde τ_{n+m+2} hasta τ_{n+m+k}), y la tarea a la que se incrementa el *jitter* es τ_{n+m+1} . Empezamos en un escenario inicial con unos valores de *jitter* mostrados en la Figura 3.9a y después cambiamos a un segundo escenario donde se ha incrementado el *jitter* de la tarea τ_{n+m+1} (ver Figura 3.9b). Cada grupo se analiza por separado para

3. Demostración de la convergencia del análisis holístico en sistemas distribuidos con planificación EDF

demostrar que el tiempo de respuesta en el segundo escenario es siempre mayor que en el primer escenario.

Para empezar con el Grupo 1, debemos demostrar el siguiente lema:

Lema 3.5. *Supongamos un grupo τ de $n+m+k$ tareas con periodos infinitos planificados bajo EDF y que usan recursos compartidos locales bajo el protocolo SRP. Asumiendo que el jitter de la tarea τ_{n+m+1} es incrementado, para una tarea τ_α en el Grupo 1 ($\tau_1 \leq \tau_\alpha \leq \tau_n$), en el que el plazo efectivo es siempre inferior al de la tarea τ_{n+m+1} , el tiempo de respuesta de peor caso obtenido mediante el algoritmo de Spuri será igual o mayor que el tiempo de respuesta calculado antes de incrementar el jitter de la tarea τ_{n+m+1} .*

Demostración. En primer lugar, analizaremos el tiempo de respuesta en el primer escenario (ver Figura 3.9a). Después, incrementaremos el *jitter* de la tarea τ_{n+m+1} y realizaremos el análisis para el segundo escenario (ver Figura 3.9b). Finalmente, compararemos los resultados en ambos escenarios para demostrar que el tiempo de respuesta no puede decrecer.

En el primer escenario, las activaciones que debemos analizar ocurren cuando todas las tareas son activadas sincronamente habiendo sufrido el máximo *jitter* o cuando el plazo de la tarea bajo análisis τ_α coincide con el plazo de otra tarea. El conjunto de activaciones a analizar, incluyendo la caso de la activación sincrónica, son: $\Psi = \{i \mid D_i - J_i \geq D_\alpha - J_\alpha\}$. Vamos a separar Ψ en cuatro casos diferentes dependiendo de a qué grupo pertenece la tarea τ_i : A1 cuando $\alpha \leq i \leq n$, A2 cuando $n+1 \leq i \leq n+m$, A3 cuando $i = n+m+1$ y A4 cuando $n+m+2 \leq i \leq n+m+k$. Si aplicamos el algoritmo holístico de Spuri a estos casos de activación, obtenemos los siguientes tiempos de respuesta:

$$A1 : r_\alpha(a(i)) = \left(\sum_{j=1}^i C_j \right) - (D_i - J_i - D_\alpha) + B_i, \quad (3.63)$$

$$\text{donde } B_i = \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \ \& \ i+1 \leq j \leq n+m+k\}$$

$$A2 : r_\alpha(a(i)) = \left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^i C_j \right) - (D_i - J_i - D_\alpha) + B_i, \quad (3.64)$$

$$\text{donde } B_i = \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \ \& \ i+1 \leq j \leq n+m+k\}$$

$$\begin{aligned}
 A3 : r_\alpha(a(i)) &= \left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^{n+m} C_j \right) + C_{n+m+1} - \\
 &- (D_{n+m+1} - J_{n+m+1} - D_\alpha) + B_{n+m+1}, \\
 \text{donde } B_{n+m+1} &= \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \ \& \ n+m+2 \leq j \leq n+m+k\}
 \end{aligned} \tag{3.65}$$

$$\begin{aligned}
 A4 : r_\alpha(a(i)) &= \left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^{n+m} C_j \right) + C_{n+m+1} + \\
 &+ \left(\sum_{j=n+m+2}^i C_j \right) - (D_i - J_i - D_\alpha) + B_i, \\
 \text{donde } B_i &= \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \ \& \ i+1 \leq j \leq n+m+k\}
 \end{aligned} \tag{3.66}$$

Como se puede ver, la diferencia con el análisis de la sección 3.3.1.2 es el término de bloqueo generado por secciones críticas que usan recursos compartidos con niveles de expulsión mayores que el de la tarea bajo análisis y que pertenecen a tareas que no pueden desalojar a dicha tarea.

En el segundo escenario, después de incrementar el *jitter* de la tarea τ_{n+m+1} ($J_{n+m+1} \rightarrow J'_{n+m+1}$), los casos de activación a analizar son los mismos que en el primer escenario. El motivo es que la posición de las tareas del grupo 1 en el orden de plazo efectivo sigue siendo el mismo. En este caso, usaremos la letra B para diferenciar los casos de activación: B1 cuando $\alpha \leq i \leq n$, B2 cuando $n+1 \leq i \leq n+m$, B3 cuando $i = n+m+1$ y B4 cuando $n+m+2 \leq i \leq n+m+k$. Los tiempos de respuesta para estas activaciones son:

$$\begin{aligned}
 B1 : r_\alpha(a(i)) &= \left(\sum_{j=1}^i C_j \right) - (D_i - J_i - D_\alpha) + B_i, \\
 \text{donde } B_i &= \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \ \& \ i+1 \leq j \leq n+m+k\}
 \end{aligned} \tag{3.67}$$

$$\begin{aligned}
 B2 : r_\alpha(a(i)) &= \left(\sum_{j=1}^n C_j \right) + C_{n+m+1} + \left(\sum_{j=n+1}^i C_j \right) - (D_i - J_i - D_\alpha) + B_i, \\
 \text{donde } B_i &= \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \ \& \ i+1 \leq j \leq n+m+k \\
 &\ \& \ j \neq n+m+1\}
 \end{aligned} \tag{3.68}$$

3. Demostración de la convergencia del análisis holístico en sistemas distribuidos con planificación EDF

$$\begin{aligned}
 B3 : r_\alpha(a(i)) &= \left(\sum_{j=1}^n C_j \right) + C_{n+m+1} - (D_{n+m+1} - J'_{n+m+1} - D_\alpha) + B_i, \\
 \text{donde } B_i &= \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \ \& \ n+1 \leq j \leq n+m+k \\
 &\ \& \ j \neq n+m+1\}
 \end{aligned} \tag{3.69}$$

$$\begin{aligned}
 B4 : r_\alpha(a(i)) &= \left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^{n+m} C_j \right) + C_{n+m+1} + \\
 &+ \left(\sum_{j=n+m+2}^i C_j \right) - (D_i - J_i - D_\alpha) + B_i, \\
 \text{donde } B_i &= \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \ \& \ i+1 \leq j \leq n+m+k\}
 \end{aligned} \tag{3.70}$$

Como podemos ver, los tiempos de bloqueo en B1 son los mismos que en A1 porque el cambio en el orden del nivel de expulsión solo afecta a las tareas que están siempre por debajo de las tareas del grupo 1. Igualmente, los tiempos de bloqueo en B4 son los mismos que en A4 por que los niveles de expulsión en el grupo 4 no cambian. Sin embargo, los tiempos de bloqueo pueden cambiar para los casos de activación de los grupos B2 y B3, con respecto a los grupos A2 y A3 del escenario 1. Los tiempos de bloqueo para las tareas del grupo 1 en los casos de activación de B2 pueden ser menores que en las activaciones de A2 porque ya no se ven influenciados por las secciones críticas de la tarea τ_{n+m+1} . Por contra, los tiempos de bloqueo de las tareas del grupo 1 en los casos de activación de B3 pueden ser mayor que en los casos de la activación A3 porque ahora están afectados por las secciones críticas de las tareas del grupo 2.

Por lo tanto, tenemos que demostrar que el tiempo de respuesta de una tarea $\tau_\alpha \in (1 \leq \alpha \leq n)$ no puede ser mayor en el primer escenario que en el segundo, por lo que la siguiente inecuación tiene que ser verdad:

$$\begin{aligned}
 \max \{r(A1) \cup r(A2) \cup r(A3) \cup r(A4)\} &\leq \\
 \leq \max \{r(B1) \cup r(B2) \cup r(B3) \cup r(B4)\}
 \end{aligned} \tag{3.71}$$

Es suficiente probar que cada grupo en el lado izquierdo de la inecuación tiene uno mayor o igual en el lado derecho. De esta forma, podemos dividir

la inecuación en cuatro diferentes:

$$\begin{aligned}
 r(B1) &\geq r(A1) \\
 &\left(\sum_{j=1}^i C_j \right) - (D_i - J_i - D_\alpha) + B_i, \text{ donde } (\alpha \leq i \leq n) \\
 &\& B_i = \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \ \& \ i + 1 \leq j \leq n + m + k\} \geq \quad (3.72) \\
 &\left(\sum_{j=1}^i C_j \right) - (D_i - J_i - D_\alpha) + B_i, \text{ donde } (\alpha \leq i \leq n) \\
 &\& B_i = \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \ \& \ i + 1 \leq j \leq n + m + k\}
 \end{aligned}$$

$$\begin{aligned}
 r(B2) &\geq r(A2) \\
 &\left(\sum_{j=1}^n C_j \right) + C_{n+m+1} + \left(\sum_{j=n+1}^i C_j \right) - (D_i - J_i - D_\alpha) + B_i, \\
 &\text{donde } (n \leq i \leq n + m) \ \& \ B_i = \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \\
 &\ \& \ i + 1 \leq j \leq n + m + k \ \& \ j \neq n + m + 1\} \geq \quad (3.73) \\
 &\left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^i C_j \right) - (D_i - J_i - D_\alpha) + B_i, \\
 &\text{donde } (n \leq i \leq n + m) \ \& \ B_i = \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \\
 &\ \& \ i + 1 \leq j \leq n + m + k\}
 \end{aligned}$$

$$\begin{aligned}
 r(B2) &\geq r(A3) \\
 &\left(\sum_{j=1}^n C_j \right) + C_{n+m+1} + \left(\sum_{j=n+1}^i C_j \right) - (D_i - J_i - D_\alpha) + B_i, \\
 &\text{donde } (n \leq i \leq n + m) \ \& \ B_i = \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \\
 &\ \& \ i + 1 \leq j \leq n + m + k \ \& \ j \neq n + m + 1\} \geq \quad (3.74) \\
 &\left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^{n+m} C_j \right) + C_{n+m+1} - (D_{n+m+1} - J_{n+m+1} - D_\alpha) + \\
 &+ B_{n+m+1}, \text{ donde } B_{n+m+1} = \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \\
 &\ \& \ n + m + 2 \leq j \leq n + m + k\}
 \end{aligned}$$

3. Demostración de la convergencia del análisis holístico en sistemas
distribuidos con planificación EDF

$$\begin{aligned}
r(B4) &\geq r(A4) \\
&\left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^n C_j \right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^i C_j \right) - (D_i - J_i - D_\alpha) + \\
&+ B_i, \text{ donde } (n + m + 2 \leq i \leq n + m + k) \quad \& \\
B_i &= \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \quad \& \quad i + 1 \leq j \leq n + m + k\} \geq \\
&\left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^n C_j \right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^i C_j \right) - (D_i - J_i - D_\alpha) + \\
&+ B_i, \text{ donde } (n + m + 2 \leq i \leq n + m + k) \quad \& \\
B_i &= \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \quad \& \quad i + 1 \leq j \leq n + m + k\}
\end{aligned} \tag{3.75}$$

Las inecuaciones (3.72) y (3.75) son siempre verdad ya que ambos lados de las inecuaciones tienen los mismos términos. De la inecuación (3.73), se obtiene:

$$\begin{aligned}
&C_{n+m+1} + B_i, \text{ donde } (n \leq i \leq n + m) \quad \& \quad B_i = \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \\
&\quad \& \quad i + 1 \leq j \leq n + m + k \quad \& \quad j \neq n + m + 1\} \geq \\
&B_i, \text{ donde } (n \leq i \leq n + m) \quad \& \quad B_i = \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \\
&\quad \& \quad i + 1 \leq j \leq n + m + k\}
\end{aligned} \tag{3.76}$$

El tiempo de bloqueo (B_i) en la parte derecha de la inecuación solo puede ser mayor que el tiempo de bloqueo de la parte izquierda cuando $j = n + m + 1$. En este caso, B_i sería igual a cs_{n+m+1} , que nunca puede ser mayor que C_{n+m+1} . Por lo tanto, la inecuación (3.73) es verdad.

Por último, en la inecuación (3.74) es suficiente encontrar una activación en B2 cuyo tiempo de respuesta sea mayor o igual que para la activación en A3. Cogemos el valor $i = n + m$ para obtener:

$$\begin{aligned}
&-(D_{n+m} - J_{n+m}) + B_{n+m}, \text{ donde } B_{n+m} = \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \\
&\quad \& \quad n + m + 1 \leq j \leq n + m + k \quad \& \quad j \neq n + m + 1\} \geq \\
&-(D_{n+m+1} - J_{n+m+1}) + B_{n+m+1}, \text{ donde } B_{n+m+1} = \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \\
&\quad \& \quad n + m + 2 \leq j \leq n + m + k\}
\end{aligned} \tag{3.77}$$

3.3. Demostración de la convergencia del análisis

Analizando la inecuación resultante, podemos decir que el tiempo de bloqueo B_{n+m+1} antes de incrementar el *jitter* de la tarea τ_{n+m+1} es el mismo que el tiempo de bloqueo B_{n+m} después de incrementar el *jitter*. La razón es que las tareas contenidas en $\{n+m+1 \leq j \leq n+m+k \ \& \ j \neq n+m+1\}$ son las mismas contenidas en $\{n+m+2 \leq j \leq n+m+k\}$. Por lo tanto, la ecuación resultante es $D_{n+m} - J_{n+m} \leq D_{n+m+1} - J_{n+m+1}$, lo que hace que la inecuación sea verdad debido a la condición establecida en la ecuación (3.26). Por lo tanto hemos demostrado que el tiempo de respuesta de las tareas en el grupo 1 no pueden decrecer por un incremento de *jitter*. \square

En segundo lugar, repetiremos el análisis con el Grupo 2 demostrando el siguiente lema:

Lema 3.6. *Supongamos un grupo τ de $n+m+k$ tareas con periodos infinitos planificados bajo EDF y que usan recursos compartidos locales bajo el protocolo SRP. Asumiendo que el *jitter* de la tarea τ_{n+m+1} es incrementado, para una tarea τ_α en el Grupo 2 ($\tau_{n+1} \leq \tau_\alpha \leq \tau_{n+m}$), tareas cuyo plazo efectivo en el primer escenario es menor que el de la tarea cuyo *jitter* se incrementa (τ_{n+m+1}), pero en el segundo escenario es superior, el tiempo de respuesta de peor caso obtenido mediante el algoritmo de Spuri será igual o mayor que el tiempo de respuesta calculado antes de incrementar el *jitter* de la tarea τ_{n+m+1} .*

Demostración. En el primer escenario los casos de activación a analizar son: A1 cuando $\alpha \leq i \leq n+m$, A2 cuando $i = n+m+1$ y A3 cuando $n+m+2 \leq i \leq n+m+k$. Por lo tanto, si aplicamos el algoritmo holístico de Spuri a estos casos de activación, obtenemos los siguientes tiempos de respuesta:

$$A1 : r_\alpha(a(i)) = \left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^i C_j \right) - (D_i - J_i - D_\alpha) + B_i, \quad (3.78)$$

donde $B_i = \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \ \& \ i+1 \leq j \leq n+m+k\}$

$$A2 : r_\alpha(a(i)) = \left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^{n+m} C_j \right) + C_{n+m+1} - (D_{n+m+1} - J_{n+m+1} - D_\alpha) + B_{n+m+1},$$

donde $B_{n+m+1} = \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \ \& \ n+m+2 \leq j \leq n+m+k\}$

$$(3.79)$$

3. Demostración de la convergencia del análisis holístico en sistemas distribuidos con planificación EDF

$$\begin{aligned}
 A3 : r_\alpha(a(i)) &= \left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^{n+m} C_j \right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^i C_j \right) - \\
 &- (D_{n+m+1} - J_{n+m+1} - D_\alpha) + B_i, \\
 \text{donde } B_i &= \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \ \& \ i + 1 \leq j \leq n + m + k\}
 \end{aligned} \tag{3.80}$$

En el segundo escenario, después de incrementar el *jitter* de la tarea τ_{n+m+1} ($J_{n+m+1} \rightarrow J'_{n+m+1}$), los casos de activación a analizar son solo dos. La razón es que la posición del grupo 2 en el orden de plazo efectivo ha cambiado. En este caso utilizaremos la letra B para diferenciar los casos de activación de los del primer escenario: B1 cuando $\alpha \leq i \leq n+m$ y B2 cuando $n+m+2 \leq i \leq n+m+k$. Los tiempos de respuesta correspondientes son:

$$\begin{aligned}
 B1 : r_\alpha(a(i)) &= \left(\sum_{j=1}^n C_j \right) + C_{n+m+1} + \left(\sum_{j=n+1}^i C_j \right) - (D_i - J_i - D_\alpha) + B_i, \\
 \text{donde } B_i &= \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \ \& \ n + 1 \leq j \leq n + m + k \\
 &\ \& \ j \neq n + m + 1\}
 \end{aligned} \tag{3.81}$$

$$\begin{aligned}
 B2 : r_\alpha(a(i)) &= \left(\sum_{j=1}^n C_j \right) + C_{n+m+1} + \left(\sum_{j=n+1}^{n+m} C_j \right) + \\
 &+ \left(\sum_{j=n+m+2}^i C_j \right) - (D_i - J_i - D_\alpha) + B_i, \\
 \text{donde } B_i &= \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \ \& \ n + 1 \leq j \leq n + m + k\}
 \end{aligned} \tag{3.82}$$

Teniendo en cuenta las ecuaciones anteriores, hay que demostrar que el tiempo de respuesta de la tarea $\tau_\alpha \in (n+1 \leq \alpha \leq n+m)$, no puede ser mayor en la primera situación que en la segunda. Por lo tanto, la siguiente inecuación tiene que ser verdad:

$$\max \{r(A1) \cup r(A2) \cup r(A3)\} \leq \max \{r(B1) \cup r(B2)\}. \tag{3.83}$$

Es suficiente con probar que cada grupo en el lado izquierdo de la inecuación tiene uno mayor o igual en el lado derecho. De esta forma, podemos

dividir la inecuación anterior en tres diferentes:

$$\begin{aligned}
 r(B1) &\geq r(A1) \\
 &\left(\sum_{j=1}^n C_j \right) + C_{n+m+1} + \left(\sum_{j=n+1}^i C_j \right) - (D_i - J_i - D_\alpha) + B_i, \\
 &\text{donde } (\alpha \leq i \leq n+m) \ \& \ B_i = \max \{cs_j(s) : [s] \geq \pi_i \ \& \\
 &\quad n+1 \leq j \leq n+m+k \ \& \ j \neq n+m+1\} \geq \\
 &\left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^i C_j \right) - (D_i - J_i - D_\alpha), \text{ donde } (\alpha \leq i \leq n+m) \\
 &\ \& \ B_i = \max \{cs_j(s) : [s] \geq \pi_i \ \& \ i+1 \leq j \leq n+m+k\}
 \end{aligned} \tag{3.84}$$

$$\begin{aligned}
 r(B2) &\geq r(A2) \\
 &\left(\sum_{j=1}^n C_j \right) + C_{n+m+1} + \left(\sum_{j=n+1}^{n+m} C_j \right) + \left(\sum_{j=n+m+2}^i C_j \right) - (D_i - J_i - D_\alpha) + \\
 &\quad + B_i, \text{ donde } (n+m+2 \leq i \leq n+m+k) \ \& \\
 &\quad B_i = \max \{cs_j(s) : [s] \geq \pi_i \ \& \ n+1 \leq j \leq n+m+k\} \geq \\
 &\left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^{n+m} C_j \right) + C_{n+m+1} - (D_{n+m+1} - J_{n+m+1} - D_\alpha) + \\
 &\quad + B_{n+m+1}, \text{ donde } B_{n+m+1} = \max \{cs_j(s) : [s] \geq \pi_i \ \& \\
 &\quad n+m+2 \leq j \leq n+m+k\}
 \end{aligned} \tag{3.85}$$

$$\begin{aligned}
 r(B2) &\geq r(A3) \\
 &\left(\sum_{j=1}^n C_j \right) + C_{n+m+1} + \left(\sum_{j=n+1}^{n+m} C_j \right) + \left(\sum_{j=n+m+2}^i C_j \right) - (D_i - J_i - D_\alpha) + \\
 &\quad + B_i, \text{ donde } (n+m+2 \leq i \leq n+m+k) \ \& \\
 &\quad B_i = \max \{cs_j(s) : [s] \geq \pi_i \ \& \ n+1 \leq j \leq n+m+k\} \geq \\
 &\left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^{n+m} C_j \right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^i C_j \right) - (D_i - J_i - D_\alpha) + \\
 &\quad B_i, \text{ donde } (n+m+2 \leq i \leq n+m+k) \ \& \\
 &\quad B_i = \max \{cs_j(s) : [s] \geq \pi_i \ \& \ i+1 \leq j \leq n+m+k\}
 \end{aligned} \tag{3.86}$$

3. Demostración de la convergencia del análisis holístico en sistemas distribuidos con planificación EDF

La inecuación (3.86) es siempre verdad ya que ambas partes de la inecuación tienen los mismos términos. Simplificando la inecuación (3.84), obtenemos la misma inecuación que en (3.76). Por lo tanto, se puede resolver de la misma forma. Y finalmente, de la inecuación (3.85) se obtiene:

$$\begin{aligned} & \left(\sum_{j=n+m+2}^i C_j \right) - (D_i - J_i) + B_i, \text{ donde } (n+m+2 \leq i \leq n+m+k) \quad \& \\ B_i = \max \{ & cs_j(s) : \lceil s \rceil \geq \pi_i \quad \& \quad n+1 \leq j \leq n+m+k \} \geq \\ & - (D_{n+m+1} - J_{n+m+1}) + B_{n+m+1}, \text{ donde } B_{n+m+1} = \max \{ cs_j(s) : \lceil s \rceil \geq \pi_i \\ & \quad \& \quad n+m+2 \leq j \leq n+m+k \} \end{aligned} \quad (3.87)$$

Esta inecuación se puede dividir en dos nuevas separando por una parte los elementos de *jitter* y plazo, y por otra parte, los elementos de bloqueo y tiempo de ejecución:

$$(D_i - J_i), \text{ donde } (n+m+2 \leq i \leq n+m+k) \leq (D_{n+m+1} - J_{n+m+1}) \quad (3.88)$$

$$\begin{aligned} & \left(\sum_{j=n+m+2}^i C_j \right) + B_i, \text{ donde } (n+m+2 \leq i \leq n+m+k) \quad \& \\ B_i = \max \{ & cs_j(s) : \lceil s \rceil \geq \pi_i \quad \& \quad n+1 \leq j \leq n+m+k \} \geq \quad (3.89) \\ B_{n+m+1}, & \text{ donde } B_{n+m+1} = \max \{ cs_j(s) : \lceil s \rceil \geq \pi_i \\ & \quad \& \quad n+m+2 \leq j \leq n+m+k \} \end{aligned}$$

La inecuación (3.88) es siempre verdad para todos los casos de i debido a la condición de partida establecida en la ecuación (3.26). En la inecuación (3.89), es suficiente con encontrar un valor de i que asegure que la parte izquierda de la inecuación es mayor o igual que la parte derecha. En este caso, ese valor es $i = n + m + k$. De esta forma, cualquier sección crítica cs_j de la tarea τ_j que puede bloquear a la tarea τ_{n+m+1} en la parte derecha de la inecuación tendrá su respectivo tiempo de ejecución C_j en el sumatorio de la parte izquierda de la inecuación. Como por definición, cs_j puede ser como mucho igual de grande que C_j , pero nunca mayor, la inecuación (3.89) es verdad siempre. Por lo tanto, queda demostrado que el tiempo de respuesta de las tareas del grupo 2 solo puede verse incrementado cuando el *jitter* de una tarea incrementa su valor. \square

3.3. Demostración de la convergencia del análisis

En tercer lugar, repetiremos el proceso con la tarea τ_{n+m+1} demostrando el siguiente lema:

Lema 3.7. *Supongamos un grupo τ de $n+m+k$ tareas con periodos infinitos planificados bajo EDF y que usan recursos compartidos locales bajo el protocolo SRP y una tarea τ_{n+m+1} del sistema que tiene tareas por encima y por debajo en el orden de plazo efectivo. Si el jitter de esta tarea es incrementado, reduciendo su plazo efectivo de forma que adelante a otras tareas en el orden de plazo efectivo, el tiempo de respuesta de peor caso, calculado mediante el algoritmo de Spuri, no puede decrecer.*

Demostración. En el primer escenario, los casos de activación a analizar son: A1 cuando $i = n + m + 1$ y A2 cuando $n + m + 2 \leq i \leq n + m + k$. Si aplicamos el análisis de Spuri a estos casos de activación, los tiempos de respuesta obtenidos son:

$$\begin{aligned}
 A1 : r_{n+m+1}(a(i)) &= \left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^{n+m} C_j \right) + C_{n+m+1} - (-J_{n+m+1}) + \\
 &B_{n+m+1}, \text{ donde } B_{n+m+1} = \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \ \& \\
 &n + m + 2 \leq j \leq n + m + k\}
 \end{aligned} \tag{3.90}$$

$$\begin{aligned}
 A2 : r_{n+m+1}(a(i)) &= \left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^{n+m} C_j \right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^{n+m+k} C_j \right) - \\
 &-(D_i - J_i - D_{n+m+1}) + B_i, \text{ donde} \\
 &B_i = \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \ \& \ i + 1 \leq j \leq n + m + k\}
 \end{aligned} \tag{3.91}$$

En el segundo escenario, después de incrementar el *jitter* de la propia tarea τ_{n+m+1} ($J_{n+m+1} \rightarrow J'_{n+m+1}$), los casos de activación a analizar son tres en vez de dos. Esto es debido al cambio de posición en el orden de plazo efectivo de la tarea τ_{n+m+1} . Usaremos la letra B para definir los grupos de activaciones: B1 cuando $n \leq i \leq n + m$, B2 cuando $i = n + m + 1$ y B3 cuando $n + m + 2 \leq i \leq n + m + k$. El tiempo de respuesta para estos grupos

3. Demostración de la convergencia del análisis holístico en sistemas distribuidos con planificación EDF

de activaciones es:

$$\begin{aligned}
 B1 : r_{n+m+1}(a(i)) &= \left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^i C_j \right) + C_{n+m+1} - \\
 &- (D_i - J_i - D_{n+m+1}) + B_i, \text{ donde } B_i = \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \ \& \\
 &i + 1 \leq j \leq n + m + k \ \& \ j \neq n + m + 1\}
 \end{aligned} \tag{3.92}$$

$$\begin{aligned}
 B2 : r_{n+m+1}(a(i)) &= \left(\sum_{j=1}^n C_j \right) + C_{n+m+1} - (-J'_{n+m+1}) + B_i, \text{ donde} \\
 B_i &= \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \ \& \ n + 1 \leq j \leq n + m + k \ \& \\
 &j \neq n + m + 1\}
 \end{aligned} \tag{3.93}$$

$$\begin{aligned}
 B3 : r_{n+m+1}(a(i)) &= \left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^{n+m} C_j \right) + C_{n+m+1} + \\
 &+ \left(\sum_{j=n+m+2}^i C_j \right) - (D_i - J_i - D_{n+m+1}) + B_i, \text{ donde} \\
 B_i &= \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \ \& \ i + 1 \leq j \leq n + m + k\}
 \end{aligned} \tag{3.94}$$

Para finalizar con la tarea τ_{n+m+1} , tenemos que demostrar que su tiempo de respuesta en el primer escenario es mayor que en el segundo escenario, por lo que la siguiente inecuación tiene que ser verdad:

$$\max \{r(A1) \cup r(A2)\} \leq \max \{r(B1) \cup r(B2) \cup r(B3)\}. \tag{3.95}$$

Es suficiente con probar que cada grupo de tiempos de respuesta de la parte izquierda de la inecuación tiene uno con mayor o igual valor en la parte

3.3. Demostración de la convergencia del análisis

derecha. De esta forma, podemos dividir la inecuación en dos:

$$\begin{aligned}
 & r(B1) \geq r(A1) \\
 & \left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^i C_j \right) + C_{n+m+1} - (D_i - J_i - D_{n+m+1}) + B_i, \\
 & \text{donde } (n \leq i \leq n+m) \ \& \ B_i = \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \ \& \\
 & \quad i+1 \leq j \leq n+m+k \ \& \ j \neq n+m+1\}, \geq \\
 & \left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^{n+m} C_j \right) + C_{n+m+1} - (-J_{n+m+1}) + B_{n+m+1}, \\
 & \text{donde } B_{n+m+1} = \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \ \& \ n+m+2 \leq j \leq n+m+k\} \\
 & \hspace{15em} (3.96)
 \end{aligned}$$

$$\begin{aligned}
 & r(B3) \geq r(A2) \\
 & \left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^{n+m} C_j \right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^i C_j \right) - \\
 & (D_i - J_i - D_{n+m+1}) + B_i, \text{ donde } (n+m+1 \leq i \leq n+m+k) \ \& \\
 & B_i = \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \ \& \ i+1 \leq j \leq n+m+k\} \geq \hspace{10em} (3.97) \\
 & \left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^{n+m} C_j \right) + C_{n+m+1} + \left(\sum_{j=n+m+2}^i C_j \right) - \\
 & (D_i - J_i - D_{n+m+1}) + B_i, \text{ donde } (n+m+1 \leq i \leq n+m+k) \\
 & B_i = \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \ \& \ i+1 \leq j \leq n+m+k\}
 \end{aligned}$$

La inecuación (3.97) es siempre verdad ya que ambos lados de la inecuación tienen los mismos términos. En la inecuación (3.96) es suficiente con encontrar un valor para i que demuestre que el lado izquierdo de la inecuación es mayor o igual que el lado derecho. Si cogemos el valor $i = n+m$ y simplificamos la inecuación, obtenemos:

$$\begin{aligned}
 & - (D_i - J_i - D_{n+m+1}) + B_{n+m}, \text{ donde} \\
 & B_{n+m} = \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \ \& \ n+m+2 \leq j \leq n+m+k\}, \geq \\
 & - (-J_{n+m+1}) + B_{n+m+1}, \text{ donde} \\
 & B_{n+m+1} = \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \ \& \ n+m+2 \leq j \leq n+m+k\} \\
 & \hspace{15em} (3.98)
 \end{aligned}$$

Analizando la inecuación (3.98), podemos ver que el tiempo de bloqueo B_{n+m+1} antes de incrementar el *jitter* de la tarea τ_{n+m+1} es el mismo que el

3. Demostración de la convergencia del análisis holístico en sistemas distribuidos con planificación EDF

tiempo de bloqueo B_{n+m} después de incrementar el *jitter*, porque las tareas con plazo efectivo mayor que pueden influir en esos tiempos de bloqueo son las mismas. Por consiguiente, podemos simplificar la inecuación para obtener que $D_{n+m} - J_{n+m} \leq D_{n+m+1} - J_{n+m+1}$, lo que es siempre verdad debido a la condición inicial establecida en la inecuación (3.26). En definitiva, hemos demostrado que el tiempo de respuesta de la tarea τ_{n+m+1} no puede decrecer debido a un incremento de su *jitter*. \square

Por último, solo queda repetir el proceso con el Grupo 3 y demostrar el siguiente lema:

Lema 3.8. *Supongamos un grupo τ de $n+m+k$ tareas con periodos infinitos planificados bajo EDF y que usan recursos compartidos locales bajo el protocolo SRP. Asumiendo que el jitter de la tarea τ_{n+m+1} es incrementado, para una tarea τ_α en el Grupo 3 ($\tau_{n+m+2} \leq \tau_\alpha \leq \tau_{n+m+k}$), tareas cuyo plazo efectivo es siempre superior al de las tareas de otros grupos, el tiempo de respuesta de peor caso obtenido mediante el algoritmo de Spuri será igual o mayor que el tiempo de respuesta calculado antes de incrementar el jitter de la tarea τ_{n+m+1} .*

Demostración. En el primer escenario, solo es necesario analizar un grupo de casos de activación: A1 cuando $n+m+2 \leq i \leq n+m+k$. Por lo tanto, si aplicamos el algoritmo holístico de Spuri a dicho grupo de activaciones, obtenemos los siguientes tiempos de respuesta:

$$\begin{aligned}
 A1 : r_\alpha(a(i)) &= \left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^{n+m} C_j \right) + C_{n+m+1} + \\
 &+ \left(\sum_{j=n+m+2}^i C_j \right) - (D_i - J_i - D_\alpha) + B_i, \text{ donde} \\
 B_i &= \max \{cs_j(s) : [s] \geq \pi_i \ \& \ i+1 \leq j \leq n+m+k\}
 \end{aligned} \tag{3.99}$$

En el segundo escenario, después de incrementar el *jitter* de la tarea τ_{n+m+1} ($J_{n+m+1} \rightarrow J'_{n+m+1}$), también tenemos un único caso de activaciones para analizar. La razón es que la posición en el orden de plazo efectivo no ha cambiado. En este caso usaremos la letra B para identificar el grupo, de forma que se diferencie del grupo del primer escenario: B1 cuando $n+m+2 \leq$

3.3. Demostración de la convergencia del análisis

$i \leq n + m + k$. Los tiempos de respuesta para estas activaciones son:

$$\begin{aligned}
 B1 : r_\alpha(a(i)) &= \left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^{n+m} C_j \right) + C_{n+m+1} + \\
 &\left(\sum_{j=n+m+2}^i C_j \right) - (D_i - J_i - D_\alpha) + B_i, \text{ donde} \\
 B_i &= \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \ \& \ i + 1 \leq j \leq n + m + k\}
 \end{aligned} \tag{3.100}$$

Por último, tenemos que demostrar que las tareas del grupo 3 no pueden tener un tiempo de respuesta mayor en el primer escenario que en el segundo. Por lo tanto, la siguiente inecuación tiene que ser verdad:

$$\max \{r(A1)\} \leq \max \{r(B1)\}. \tag{3.101}$$

Es suficiente con probar que el tiempo de respuesta en B1 es mayor que en A1, por lo que hay que probar que:

$$\begin{aligned}
 r(B1) &\geq r(A1) \\
 &\left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^{n+m} C_j \right) + C_{n+m+1} + \\
 &\left(\sum_{j=n+m+2}^i C_j \right) - (D_i - J_i - D_\alpha) + B_i, \text{ donde} \\
 B_i &= \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \ \& \ i + 1 \leq j \leq n + m + k\} \geq \\
 &\left(\sum_{j=1}^n C_j \right) + \left(\sum_{j=n+1}^{n+m} C_j \right) + C_{n+m+1} + \\
 &\left(\sum_{j=n+m+2}^i C_j \right) - (D_i - J_i - D_\alpha) + B_i, \text{ donde} \\
 B_i &= \max \{cs_j(s) : \lceil s \rceil \geq \pi_i \ \& \ i + 1 \leq j \leq n + m + k\}
 \end{aligned} \tag{3.102}$$

La inecuación (3.102) es siempre verdad ya que ambas partes de la inecuación tienen los mismos términos. \square

Recapitulando, hemos evidenciado que, si el *jitter* de la tarea τ_{n+m+1} , J_{n+m+1} , es incrementado, el tiempo de respuesta de las tareas del sistema

no puede decrecer. Esta afirmación puede ser extendida al incremento del *jitter* de cualquier tarea o grupo de tareas del sistema. Mediante los grupos de tareas 1, 2 y 3 se contemplan todos los posibles roles que una tarea puede tener en relación a la tarea cuyo *jitter* es incrementado. Si hay un grupo de tareas que cambian su *jitter*, podemos hacer sucesivos cambios individuales y observar que el tiempo de respuesta no decrece.

En este punto, hemos demostrado que en un sistema con un número indefinido de tareas con periodos infinitos y con uso de recursos compartidos, los tiempos de respuesta de peor caso de las tareas no decrece por el incremento de los *jitters*. \square

3.3.3. Sistema con Periodos Finitos

En las secciones anteriores hemos analizado grupos de tareas con periodos restringidos a valores infinitos. En esta sección explicaremos por qué esta restricción se puede quitar sin invalidar la demostración anterior.

Teorema 3.4. *Supongamos un grupo τ de tareas planificado mediante EDF y con la posibilidad de usar recursos compartidos locales bajo el protocolo SRP o IPCP. El tiempo de respuesta de peor caso de cualquier tarea en τ no puede decrecer como efecto del incremento del jitter de cualquier tarea en τ .*

Demostración. Empecemos por un sistema como τ pero con periodos infinitos. A este sistema le llamaremos τ' . De acuerdo con el lema 3.3 el tiempo de respuesta solo puede crecer como consecuencia de cambiar de un escenario inicial con unos *jitters* específicos a un segundo escenario en el que los *jitters* se han incrementado. Ahora imaginemos que reducimos el periodo de una de las tareas de forma que su segunda activación o instancia interfiera en el tiempo de respuesta de la tarea bajo análisis en el segundo escenario. Este nuevo sistema con periodos no infinitos se llamará τ'' .

En el Teorema 3.3 hemos demostrado que en un sistema τ' , el tiempo de respuesta de una tarea τ_i bajo análisis es más grande en el escenario 2 (con *jitters* más grandes) que en el escenario 1:

$$r_i(\tau', \text{escenario2}) \geq r_i(\tau', \text{escenario1}) \quad (3.103)$$

En la Ecuación (2.56), se puede observar que, cuando el periodo de las tareas decrece, la longitud del periodo de ocupación puede crecer debido a

3.3. Demostración de la convergencia del análisis

que nuevas instancias de dicha tarea pueden activarse dentro del periodo de ocupación. Consecuentemente, el tiempo de respuesta puede crecer. Por lo tanto, podemos decir que, para el mismo escenario, los tiempos de respuesta en el sistema τ'' son siempre mayores o iguales que en el sistema τ' :

$$r_i(\tau'', \text{escenario1}) \geq r_i(\tau', \text{escenario1}) \quad (3.104)$$

$$r_i(\tau'', \text{escenario2}) \geq r_i(\tau', \text{escenario2}) \quad (3.105)$$

Estas nuevas instancias de la tarea τ_i que entran en el periodo de ocupación del escenario, que hacen que $r_i(\tau'', \text{escenario1})$ sea más grande que $r_i(\tau', \text{escenario1})$, también afectan a la tarea τ_i en el escenario 2, cuyo periodo de ocupación es más largo que el escenario 1 de la tarea τ_i . Esto hace que las nuevas instancias alargarán el tiempo de respuesta de la tarea τ_i en el escenario 2 por lo menos en la misma cantidad, causando que el tiempo de respuesta de la tarea bajo análisis en el escenario 2 se convierta incluso más grande que el del escenario 1:

$$r_i(\tau'', \text{escenario2}) \geq r_i(\tau'', \text{escenario1}) \quad (3.106)$$

Consecuentemente, no puede haber un decremento del tiempo de respuesta si vamos del escenario 1 al escenario 2 en el caso de un sistema con periodos no infinitos.

Lo mismo ocurre si reducimos el periodo de más tareas introduciendo nuevas instancias en el periodo de ocupación, haciéndolo más grande. Cada vez que un periodo es reducido, más instancias de esa tarea o de tareas diferentes pueden ser añadidas al periodo de ocupación, incrementando su longitud y posiblemente provocando que se incrementen los tiempos de respuesta. No es posible generar un decremento en la longitud del periodo de ocupación reduciendo el periodo de las tareas. \square

El teorema 3.4 nos permite concluir que el análisis holístico definido en [Spu96b] no sufre de problemas de convergencia debido al uso de recursos compartidos, siempre que los niveles de expulsión y los techos de prioridad de los recursos compartidos se mantengan en un orden coherente en relación a las variaciones en el plazo efectivo de las tareas en las diferentes iteraciones del algoritmo. Este último requerimiento no aparece explícitamente en [Spu96b] y supone una extensión del algoritmo de cálculo de tiempos de respuesta en el que los niveles de expulsión y los techos de prioridad de los recursos se recalculan en cada iteración, tal como se describió en la Sección 3.1 (ver Algoritmo 3.1).

3.4. Aplicación en un ejemplo industrial

En esta sección analizaremos un ejemplo industrial con la técnica descrita en las secciones anteriores. El modelo de tareas escogido es parte de un sistema de ascensores, concretamente el operador de puertas. Este módulo es responsable de la apertura y cierre de las puertas del ascensor. Además este sistema debe cumplir unas restricciones temporales. La restricción más importante se da cuando el sistema está cerrando la puerta y un obstáculo (normalmente una persona) se interpone en el camino de la puerta. El sistema debe completar la acción de parar el cierre de las puertas e iniciar la reapertura en un intervalo de tiempo específico, con el objetivo de no causar daños a la persona que se encuentra en el camino de la puerta.

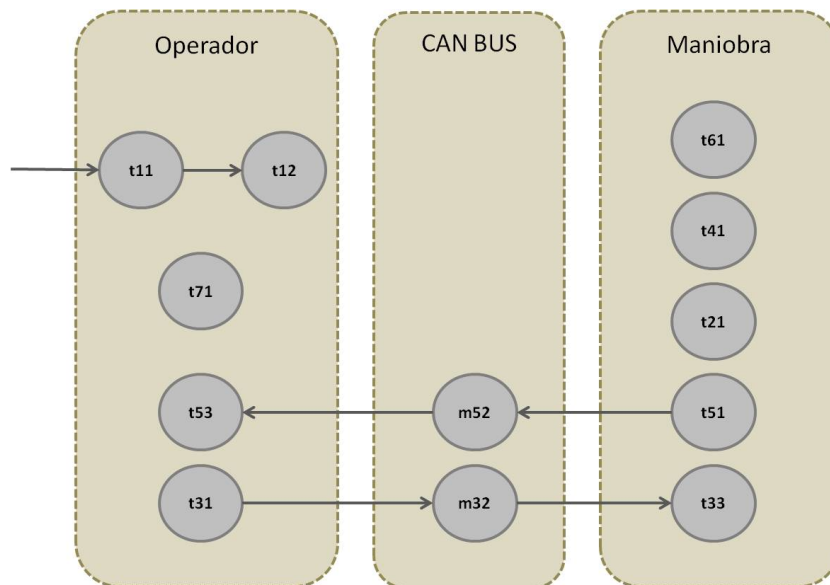


Figura 3.10: Modelo de tareas del sistema del operador de puertas.

Como podemos ver en el modelo de despliegue de las tareas (ver Figura 3.10), se trata de un sistema distribuido de tiempo real compuesto por dos procesadores (operador y maniobra) conectados por un bus CAN. El operador controla las tareas mecánicas de apertura/cierre de la puerta; mientras que la maniobra toma las decisiones sobre todo el sistema. Podemos ver también que el sistema está compuesto por 7 flujos de tareas, que a su vez están compuestos por 10 tareas y 2 mensajes cuyos parámetros están descritos en las Tablas 3.1, 3.2 y 3.3, respectivamente. Si observamos la tabla de flujos

3.4. Aplicación en un ejemplo industrial

de tareas, vemos que existe un flujo de tareas esporádico llamado “obstacle”. Para el análisis de tiempo de respuesta de peor caso, éste ha sido modelado como un flujo de tareas periódico con periodo igual a su MIT (*Minimum Inter-arrival Time*, Tiempo Mínimo entre Llegadas). La política de planificación aplicada en ambos procesadores es EDF y se ha usado la técnica descrita en [Spu96a, Spu96b] para calcular el tiempo de respuesta de las tareas. Para calcular el tiempo de respuesta de los mensajes en el bus CAN se ha utilizado la técnica para prioridades fijas sin expulsiones introducida en [THW94] y revisada en [DBBL07]. Como las políticas de planificación son diferentes en los procesadores y en la red, se ha usado la técnica de análisis para sistemas heterogéneos [RGPGH11].

Tabla 3.1: Flujos de tareas del sistema.

Flujo de tareas	Nombre	Tipo de evento	T / MIT (ms)
e2eF1	obstacle	esporádico	1000.00
e2eF2	door_power	periódico	1000.00
e2eF3	op_vss	periódico	200.00
e2eF4	man_vss	periódico	100.00
e2eF5	pdo_transmission	periódico	100.00
e2eF6	man_reports	periódico	500.00
e2eF7	op_reports	periódico	500.00

Tabla 3.2: Tareas del sistema.

Tarea	Nombre	C (ms)	D (ms)	proc
t11	op_obstacle_isr	0.01	0.10	op
t12	op_obstacle_task	1.00	10.00	op
t21	man_powersafe_timer	1.00	10.00	man
t31	op_vss_machine_task	120.00	200.00	op
t33	man_comm_task	1.0	200.00	man
t41	man_vss_machine_task	50.00	100.00	man
t51	man_periodic_send_task	1.00	100.00	man
t53	op_comm_task	1.00	100.00	op
t61	man_reporter_task	5.00	500.00	man
t71	op_reporter_task	5.00	500.00	op

El sistema también tiene recursos compartidos que se encuentran representados en la Tabla 3.4. El uso de estos recursos por parte de las tareas se

3. Demostración de la convergencia del análisis holístico en sistemas distribuidos con planificación EDF

Tabla 3.3: Mensajes del sistema.

Mensaje	Nombre	C (ms)	Prioridad	Red
m32	msj_sdo_server	0.40	2	can
m52	msj_pdo_server	0.40	1	can

puede ver en la Tabla 3.5.

Tabla 3.4: Recursos compartidos en el sistema.

Recurso	Nombre
R1	man_events_fifo
R2	man_reports_fifo
R3	op_events_fifo
R4	op_reports_fifo

Tabla 3.5: Utilización de los recursos compartidos por las tareas.

Tarea	R1 (ms)	R2 (ms)	R3 (ms)	R4 (ms)
t11	-	-	-	-
t12	-	-	0.07	-
t21	0.07	-	-	-
t31	-	-	0.06	1.0
t33	0.07	-	-	-
t41	0.06	1.00	-	-
t51	-	-	-	-
t53	-	-	0.07	-
t61	-	0.01	-	-
t71	-	-	-	0.01

Todos estos elementos nos dan un modelo completo para realizar el análisis mediante la técnica holística. Tras aplicar dicha técnica se obtienen los resultados recogidos en las tablas 3.6 y 3.7. En la Tabla 3.7 podemos ver el límite superior del tiempo de respuesta de peor caso para cada tarea. Se puede apreciar que todas las tareas tienen un tiempo de respuesta de peor caso menor que el plazo asociado. Por lo tanto, se puede concluir que el sistema es planificable. Además, la Tabla 3.6 muestra una visión general de la

3.5. Conclusiones

utilización de los procesadores. La baja utilización del bus CAN se debe a la poca cantidad de mensajes que se envían por la red. La utilización de los procesadores Operador y Maniobra (62 % y 52 % respectivamente) revela que el sistema podría soportar más carga de procesamiento.

Tabla 3.6: Utilización de los procesadores y la red.

Procesador/Red	U(%)
Operador	62.10
bus CAN	0.60
Maniobra	52.60

Tabla 3.7: Tiempos de respuesta de peor caso de tareas y mensajes.

Tarea/Mensaje	Nombre	D (ms)	R (ms)
t11	op_obstacle_isr	0.10	0.08
t12	op_obstacle_task	10.00	1.16
t21	man_powersafe_timer	10.00	1.06
t31	op_vss_machine_task	200.00	123.00
m32	msj_sdo_server	200.00	123.82
t33	man_comm_task	200.00	153.06
t41	man_vss_machine_task	100.00	53.01
t51	man_periodic_send_task	100.00	54.00
m52	msj_pdo_server	100.00	54.80
t53	op_comm_task	100.00	56.87
t61	man_reporter_task	500.00	58.00
t71	op_reporter_task	500.00	128.00

Los resultados obtenidos en este análisis nos permiten mostrar que el análisis holístico puede ser utilizado para analizar sistemas industriales reales y que se pueden aplicar herramientas que lo implementen.

3.5. Conclusiones

En este capítulo se ha probado la corrección de la técnica holística de Spuri para el análisis de planificabilidad de sistemas distribuidos de tiempo

3. Demostración de la convergencia del análisis holístico en sistemas distribuidos con planificación EDF

real con planificación EDF, recogida en la sección 2.3.2. Para ello, se ha demostrado que sus fórmulas iterativas son no-decrecientes como se afirmaba en [Spu96a, Spu96b], tanto usando recursos compartidos, como sin ellos. Además, hemos aplicado esta técnica al subsistema de control de apertura/cierre de puertas de un ascensor. Los resultados obtenidos verifican la planificabilidad del sistema y el cumplimiento de las restricciones temporales.

Una conclusión importante que se ha obtenido del trabajo descrito en esta sección es que, debido a que los niveles de expulsión y los techos de prioridad de los recursos compartidos dependen de los términos de *jitter*, cuyos valores definitivos únicamente se conocen tras finalizar el análisis, dichos niveles y techos deberán ir variando durante el análisis y los valores definitivos se obtendrán como resultado del proceso de análisis. Esto no se mencionaba en el trabajo original de Spuri.

Se han implementado herramientas prototipo con las técnicas descritas en este capítulo para analizar el caso de estudio de la sección 3.4. Aunque el método holístico está implementado en la herramienta MAST (*Modeling and Analysis Suite for Real-Time Applications*, Herramientas de Modelado y Análisis para Aplicaciones de Tiempo Real) [MAS], desarrollada por el grupo ISTR (Ingeniería de Software y Tiempo Real) de la Universidad de Cantabria, está restringido a sistemas sin recursos compartidos, porque antes de este trabajo la aplicabilidad de la técnica holística a sistemas con recursos compartidos planificados bajo EDF estaba cuestionada. Al haberse demostrado su correcto funcionamiento en este trabajo, en el futuro MAST se extenderá para cubrir estos sistemas. El algoritmo holístico extendido, presentado en este capítulo, tendrá un impacto en la arquitectura de la herramienta de análisis MAST, que trata los niveles de expulsión y los techos de los recursos como datos de entrada del análisis. Como se ha comentado antes, los niveles y los techos deben cambiar en cada iteración del sistema, obteniéndose los valores definitivos a la finalización del proceso de análisis. Por ello, la arquitectura de la herramienta debe cambiar.

Capítulo 4

Relaciones de precedencia en el análisis de planificabilidad para sistemas distribuidos de tiempo real con planificación EDF

RESUMEN: Las técnicas para el cálculo de tiempos de respuesta de peor caso para sistemas distribuidos basadas en *offsets* o WCDO (sección 2.3.3 y 2.3.4) y en explotar las relaciones de precedencia o WC-DOPS (sección 2.3.5) han demostrado ser muy efectivas en reducir el pesimismo inherente a la técnica holística (sección 2.3.1 y 2.3.2). En la tabla 4.1 se puede ver un resumen de algunas de estas técnicas para el cálculo de tiempos de respuesta de peor caso en sistemas distribuidos con sus respectivas referencias para sistemas de prioridades fijas y EDF, tanto global, como local. Como se puede observar, hay técnicas que no están desarrolladas para planificadores EDF.

En este capítulo trataremos de extender estas técnicas a la planificación EDF. En primer lugar, en la sección 4.1, analizaremos la aplicabilidad de la técnica de precedencias de la sección 2.3.5 a sistemas planificados con EDF, tanto en sistemas con reloj global (GC-EDF), como con reloj local (LC-EDF). En segundo lugar, en la sección 4.2, como primera contribución en el capítulo, extenderemos la técnica basada en *offsets* (sección 2.3.4) a sistemas LC-EDF [DdCUGHP14a]. Esto será un paso previo a la extensión de la técnica de precedencias (sección 2.3.5) a sistemas LC-EDF [DdCUGHP14a], en la sección 4.3 y la contribución principal de este capítulo. La aplicación de las técnicas

4. Relaciones de precedencia en el análisis de planificabilidad para sistemas distribuidos de tiempo real con planificación EDF

Tabla 4.1: Técnicas para cálculo de tiempos de respuesta de peor caso.

Técnica	FP	GC-EDF	LC-EDF
Holística	[THW94]	[Spu96b]	[RGPGH10]
WCDO	[PGH98]	[PGH03]	
WCDOPS	[PGH99]		
Slanted	[MTN08]		
Slanted+PS			

Slanted y Slanted+PS a EDF la dejamos fuera de esta tesis, ya que la unión de la técnica slanted con la técnica de relaciones de precedencia aún no esta desarrollada ni siquiera para FP y solo se plantea como un trabajo futuro en el trabajo de Mäki-Turja y Nolin [MTN08]. En la sección 4.4 veremos una comparativa entre las técnicas previas y las presentadas en este capítulo. Esto nos permitirá ver las ventajas que presentan estos nuevos algoritmos. Por último, en la sección 4.5, analizaremos las conclusiones obtenidas durante la realización del trabajo presentado en este capítulo.

4.1. Aplicabilidad de las relaciones de precedencia a la planificación EDF

Como se ha comentado antes, la técnica de análisis que explota las relaciones de precedencia [PGH99] fue desarrollada para sistemas con planificación FP. El objetivo de este capítulo es modificar esta técnica para que se adapte a sistemas con planificación EDF. Para ello, en esta sección analizaremos la aplicabilidad que tiene la técnica en dichos sistemas. En la planificación EDF, las tareas son planificadas con una prioridad dependiente del momento en el que se cumple su plazo. Es decir, cuanto más cercano tiene una tarea su plazo absoluto, más alta es la prioridad de esa tarea. Por lo tanto, el análisis de la interferencia que un flujo de tareas genera sobre el tiempo de respuesta de otra tarea depende del plazo absoluto, el cual cambia de una instancia a otra, contrariamente a lo que ocurre en sistemas con planificación FP. Para

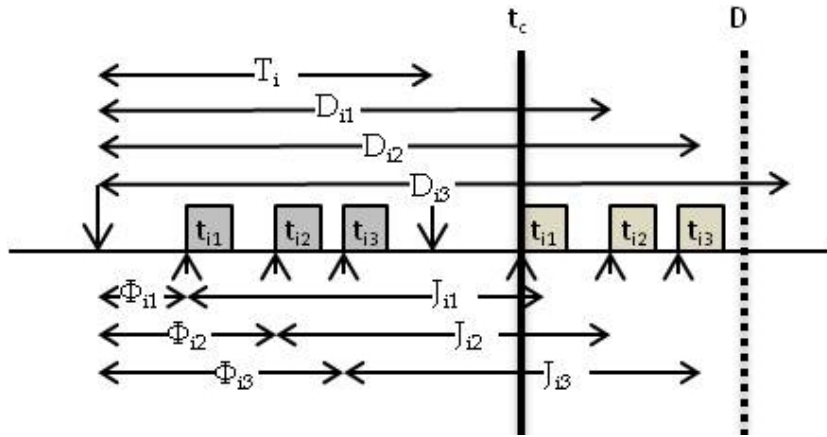


Figura 4.1: Posibles escenarios de ejecución de un flujo de tareas en un sistema GC-EDF

analizar la aplicabilidad estudiaremos dos tipos de sistemas planificados mediante EDF: sistemas con reloj global (GC-EDF) y sistemas con reloj local (LC-EDF).

Empezando con los sistemas GC-EDF, en la Figura 4.1 podemos ver un ejemplo de un posible escenario de ejecución de un flujo de tareas. Tanto los *jitters*, J_{ij} , y los *offsets*, Φ_{ij} , de las tareas; como el periodo del flujo de tareas se representan con flechas horizontales. Las flechas verticales hacia abajo indican la llegada del evento externo y las flechas verticales hacia arriba la activación de las tareas. Las cajas sombreadas simbolizan la ejecución de las tareas. Por último, los plazos absolutos de la primera instancia de las tareas del flujo de tareas Γ_i , D_{ij} , están representados por flechas horizontales y el instante de finalización del plazo absoluto de la tarea bajo análisis, D , con una línea discontinua vertical. En un sistema con reloj global, como el que se puede ver en la imagen, los plazos son relativos a la llegada del evento externo. Además, también se puede observar que, la precedencia de las tareas implica que, dentro de una instancia del flujo de tareas, el plazo de cada tarea es siempre posterior al de la tarea previa, esto es, $D_{ij} > D_{ij-1}$. Como se ha comentado en la sección 2.3.5, para sacar beneficio de las relaciones de precedencia, es necesario que se den conflictos entre las tareas del flujo de tareas. Esto ocurre cuando entre dos tareas con prioridad más alta que la tarea bajo análisis se encuentra una con prioridad menor. Como en un sistema GC-EDF los plazos, y por tanto, la prioridad, se encuentran ordenados, no se le puede sacar partido a esta técnica. Por lo tanto, podemos concluir que la técnica que explota las relaciones de precedencia no es aplicable a un sistema

4. Relaciones de precedencia en el análisis de planificabilidad para sistemas distribuidos de tiempo real con planificación EDF

GC-EDF, o por lo menos, no obtiene beneficios frente a la técnica con offsets (sección 2.3.4).

Si nos trasladamos a sistemas LC-EDF, en los que no existe un reloj global y cada nodo tiene su propio reloj local, podemos ver un ejemplo de ejecución de un flujo de tareas en la Figura 4.2-a. En este caso, los plazos de las tareas, d_{ij} , son relativos a la activación de la tarea en su procesador. Se puede observar que en este caso el plazo absoluto de una tarea τ_{ij} , representado en la figura por la terminación derecha de la flecha que representa el plazo, puede ser anterior al de la tarea precedente τ_{ij-1} o posterior al de la tarea sucesora τ_{ij+1} . Por lo tanto, en un instante dado, la prioridad de las tareas no se encuentra ordenada por orden de precedencia como sí ocurría en los sistemas GC-EDF. Por ejemplo, se puede observar en la Figura 4.2-b cómo el plazo de una instancia de la tarea τ_{i2} (d_{i2}) finaliza más tarde que el de la instancia de la tarea τ_{i3} (d_{i2}) de la misma activación del flujo de tareas. Teniendo en cuenta esta observación, podemos afirmar que en los sistemas LC-EDF, se pueden dar conflictos de activación entre las tareas de un mismo flujo de tareas, tal como veremos a continuación.

En la Figura 4.2-b podemos observar los diferentes escenarios que pueden ocurrir al analizar la interferencia generada por el flujo de tareas de la Figura 4.2-a para una tarea bajo análisis τ_{ab} (representada con fondo rallado) y cuyo plazo es D (representado con una línea discontinua vertical). Podemos ver un primer escenario en el que las activaciones de las tareas en la primera instancia del flujo de tareas se han visto retrasadas hasta el instante crítico. Si nos fijamos en ese escenario, debido a que la primera instancia de la tarea τ_{i2} tiene un plazo posterior a la tarea bajo análisis, esta instancia no podrá ejecutarse en el periodo de ocupación y no generará interferencia en el tiempo de respuesta. Pero además, impedirá que las instancias de las tareas τ_{i1} y τ_{i3} de la misma instancia del flujo de tareas puedan interferir en el mismo periodo de ocupación para la tarea bajo análisis. Si observamos el segundo escenario, veremos que la primera instancia de las tareas τ_{i1} y τ_{i2} se ha ejecutado con anterioridad al periodo de ocupación y, por lo tanto, en este caso no impiden que la instancia de la tarea τ_{i3} interfiera con la tarea bajo análisis. Por lo ello, tenemos un conflicto similar a los que se describen en la sección 2.3.5 para prioridades fijas, y, consecuentemente, podemos adaptar la técnica de relaciones de precedencia para reducir el pesimismo en el análisis de sistemas LC-EDF.

Se puede concluir que la técnica de análisis de relaciones de precedencia no proporciona ventajas para sistemas planificados con GC-EDF, ya que no

4.1. Aplicabilidad de las relaciones de precedencia a la planificación EDF

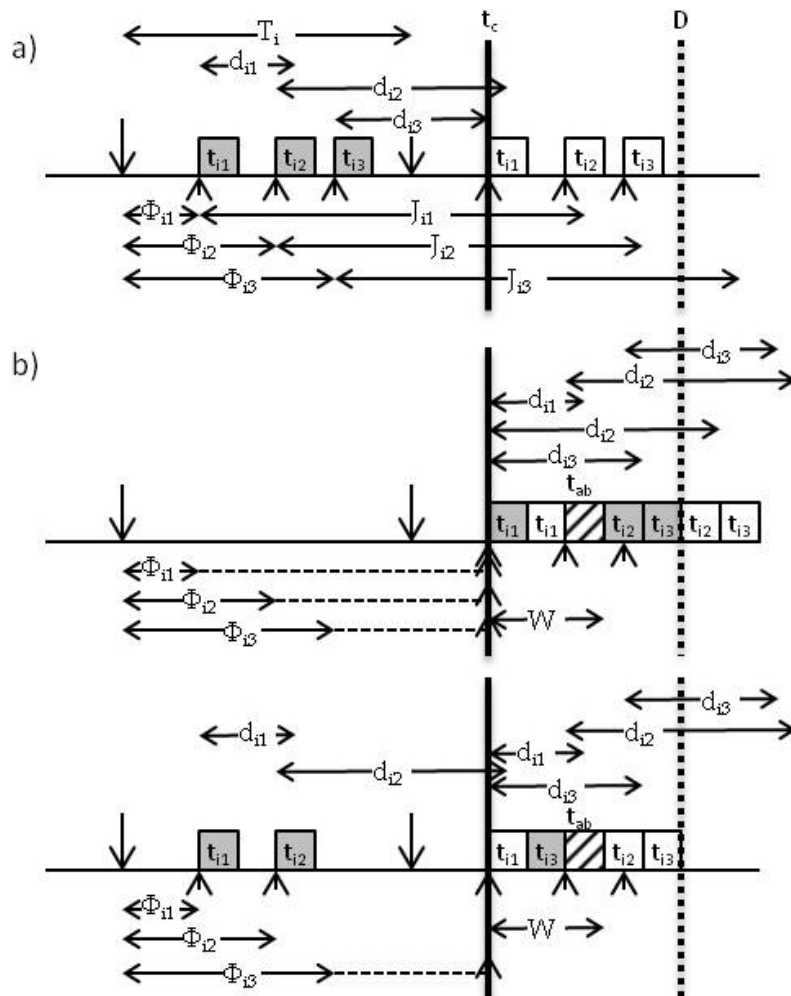


Figura 4.2: Posible escenario de ejecución de un flujo de tareas en un sistema LC-EDF

4. Relaciones de precedencia en el análisis de planificabilidad para sistemas distribuidos de tiempo real con planificación EDF

se generan conflictos de interferencia en la creación de instantes críticos. Por contra, sí que puede suponer una ventaja en sistemas planificados con LC-EDF. Por tanto, a partir de este punto nos centraremos en sistemas con planificación LC-EDF.

4.2. Técnica basada en offsets para sistemas LC-EDF

En esta sección vamos a adaptar el análisis de tiempos de respuesta basado en *offsets* para sistemas distribuidos con planificación EDF [PGH05] con objeto de analizar sistemas con planificación LC-EDF, en los que no existe un reloj global. Aunque el nuevo algoritmo de análisis se basa en los mismos principios que el anterior, las ecuaciones resultantes varían ligeramente.

En el análisis basado en *offsets* el análisis se realiza teniendo en cuenta que las tareas forman flujos de tareas, y por lo tanto, no son independientes. Las relaciones temporales entre las tareas del mismo flujo de tareas se establecen mediante los términos de *offset*. Además, el análisis de tiempos de respuesta se basa en la técnica holística, en la que cada procesador se analiza por separado. Es decir, se analizan los tiempos de respuesta de las tareas de cada procesador, sin tener en cuenta las demás tareas. Posteriormente estos tiempos de respuesta se utilizan para recalcular los nuevos *jitters* y *offsets* de todas las tareas. Este proceso se repite hasta que se obtiene una solución estable. El análisis en cada procesador se basa en estudiar el tiempo de respuesta en periodos de ocupación, que son intervalos de tiempo durante los cuales el sistema está continuamente ejecutando tareas con plazo anterior o igual al plazo absoluto de la instancia de la tarea bajo análisis. El tiempo de respuesta de peor caso se encuentra en el periodo de ocupación más largo, empezando en un instante crítico dependiente de la tarea bajo análisis.

El Teorema 4.1, probado en [PGH05], permite encontrar el instante crítico para una instancia de una tarea bajo análisis.

Teorema 4.1. *La contribución de peor caso de un flujo de tareas Γ_i al tiempo de respuesta de una instancia de la tarea τ_{ab} (tarea bajo análisis) se obtiene cuando la primera activación de una de sus tareas τ_{ik} se da dentro del periodo de ocupación, después de haber sufrido el máximo retraso posible, es decir, el máximo jitter, J_{ik} .*

El teorema no especifica cuál de las tareas debe ser la que tiene que ser posicionada en el instante crítico que inicia el periodo de ocupación. Por ello es necesario analizar todos los posibles periodos de ocupación creados con cada tarea del flujo de tareas. Dado un periodo de ocupación creado con una tarea τ_{ik} , tenemos que calcular la contribución de cada tarea τ_{ij} del flujo de tareas Γ_i .

En la Figura 4.3 se pueden ver dos escenarios diferentes del patrón de activación de la tarea τ_{ij} . Definamos Φ como el desfase con el inicio del periodo de ocupación, es decir, la diferencia de tiempo entre la llegada del evento que activa el flujo de tareas Γ_i que ocurre inmediatamente antes del periodo de ocupación y el inicio del periodo de ocupación. Con motivo de calcular la contribución de la tarea τ_{ij} al tiempo de respuesta de la tarea τ_{ab} , clasificaremos cada instancia de la tarea τ_{ij} en uno de los siguientes conjuntos:

- Conjunto 0: Activaciones que ocurren antes del periodo de ocupación y que, incluso aplicando el máximo *jitter*, no pueden ser retrasadas hasta el periodo de ocupación.
- Conjunto 1: Activaciones que ocurren antes del periodo de ocupación y pueden ser retrasadas con una cantidad de *jitter* de forma que la activación coincida con el inicio del periodo de ocupación.
- Conjunto 2: Activaciones que ocurren dentro del periodo de ocupación.

Cogiendo como ejemplo el Escenario 1 de la Figura 4.3, en el que el valor del término φ_{ijk} (que indica el lapso de tiempo entre el inicio del periodo de ocupación y el instante de la primera activación dentro de este periodo) es mayor que el valor de ϕ_{ij} , la instancia t_2 de la tarea será parte del Conjunto 1, así como las instancias t_0 y t_1 . Por contra, en el Escenario 2, en el que el valor de φ_{ijk} es menor que el valor de ϕ_{ij} , la instancia t_2 pertenece al Conjunto 2. En este caso, únicamente las instancias t_0 y t_1 pertenecen al Conjunto 1.

Una vez categorizadas todas las instancias en los diferentes conjuntos, se puede calcular el *jitter* que hay que aplicar a cada conjunto para maximizar la contribución al tiempo de respuesta de peor caso de la tarea bajo análisis τ_{ab} :

Teorema 4.2. *Dada una instancia de la tarea τ_{ab} con plazo D , y una relación de desfase Φ entre el patrón de llegadas del flujo de tareas Γ_i y el inicio del periodo de ocupación, la contribución de peor caso de la tarea τ_{ij} al tiempo*

4. Relaciones de precedencia en el análisis de planificabilidad para sistemas distribuidos de tiempo real con planificación EDF

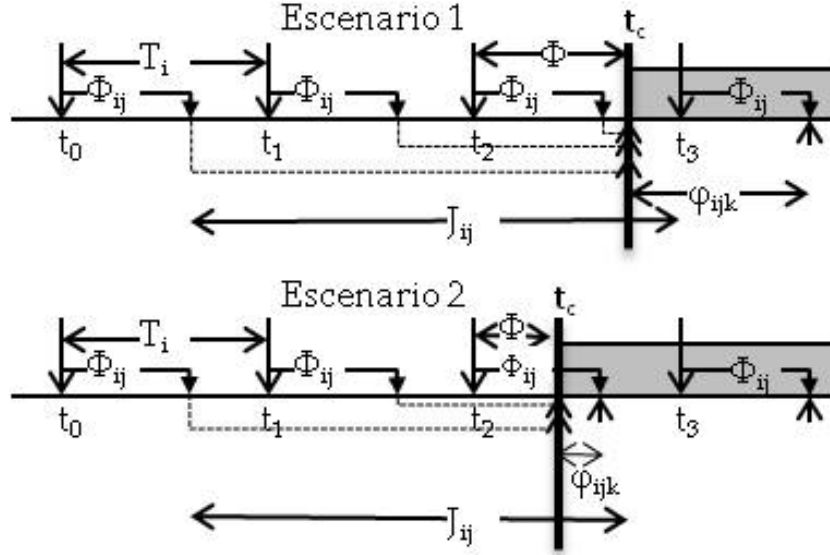


Figura 4.3: Escenarios para calcular la contribución de la tarea τ_{ij} al periodo de ocupación.

de respuesta de la tarea τ_{ab} ocurre cuando las activaciones del Conjunto 1 sufren una cantidad de jitter que les hace activarse al inicio del periodo de ocupación, y cuando las activaciones del conjunto 2 no sufren jitter.

Teniendo en cuenta este teorema, probado en [PGH05], podemos calcular la contribución de peor caso de las tareas pertenecientes al flujo de tareas Γ_i a un periodo de ocupación de longitud t y plazo D , cuando el periodo de ocupación se ha creado en el instante crítico con la tarea τ_{ik} . Con dicho objetivo, vamos a analizar las instancias de los Conjuntos 1 y 2 por separado. Las instancias en el Conjunto 1 se acumulan al inicio del periodo de ocupación y se puede calcular su número con la siguiente ecuación:

$$n_{ij}^{Set1} = \left\lfloor \frac{J_{ij} + \varphi_{ijk}}{T_i} \right\rfloor \quad (4.1)$$

donde φ_{ijk} se puede obtener mediante la ecuación:

$$\varphi_{ijk} = T_i - (\phi_{ik} + J_{ik} - \phi_{ij}) \text{ mód } T_i \quad (4.2)$$

Como se puede ver en la Figura 4.4 los plazos de las instancias del Conjunto 1 (d_{ij}^0 y d_{ij}^1) toman como referencia sus instantes de activación al inicio

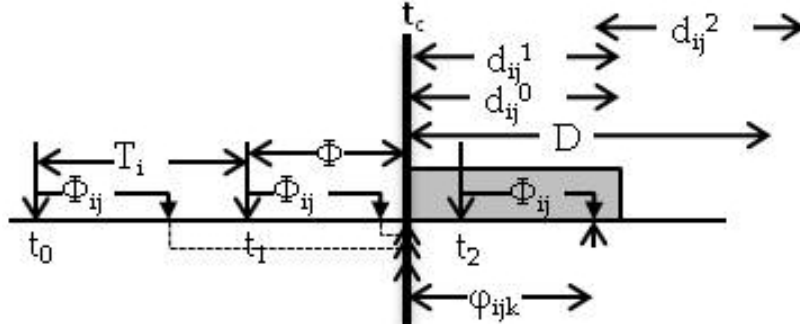


Figura 4.4: Efecto de los plazos locales para el cálculo de la contribución de la tarea τ_{ij} al periodo de ocupación.

del periodo de ocupación. Esta parte del análisis difiere del análisis para GC-EDF en [PGH05]. Estas instancias solo pueden contribuir al periodo de ocupación si sus plazos d_{ij} son menores o iguales que D . Si no fuese así, tendrían un plazo posterior al de la tarea bajo análisis y no podrían desalojarla del procesador. Consecuentemente, la contribución de peor caso de las instancias del Conjunto 1 es:

$$W_{ijk}^{Set1}(D) = (n_{ij}^{Set1} * H(D - d_{ij})) * C_{ij} \quad (4.3)$$

donde $H(x)$ es la función escalón de Heaviside o función escalón unitario, cuyo resultado se calcula mediante:

$$H(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} \quad (4.4)$$

Analizando las instancias del Conjunto 2, se puede calcular el número de instancias que se activarán dentro del periodo de longitud t y tienen un plazo anterior al plazo D con la siguiente ecuación:

$$n_{ij}^{Set2} = \min \left(\left\lceil \frac{t - \varphi_{ijk}}{T_i} \right\rceil, \left\lfloor \frac{D - \varphi_{ijk} - d_{ij}}{T_i} \right\rfloor + 1 \right)_0 \quad (4.5)$$

donde $\min(x, y)_0$ es el valor mínimo entre x e y con un valor límite mínimo de 0. La primera parte de la función mínimo define el número de instancias que pueden ser activados dentro del periodo de ocupación, y la segunda parte indica el número de instancias que tienen su plazo antes que D . Por lo tanto, la contribución de peor caso de las instancias del Conjunto 2 es:

$$W_{ijk}^{Set2}(t, D) = n_{ij}^{Set2} * C_{ij} \quad (4.6)$$

4. Relaciones de precedencia en el análisis de planificabilidad para sistemas distribuidos de tiempo real con planificación EDF

Combinando las ecuaciones (4.1), (4.3), (4.5) y (4.6), obtenemos la contribución total de la tarea τ_{ij} al periodo de ocupación de longitud t y plazo D creado con la tarea τ_{ik} :

$$W_{ijk}(t, D) = \left[\left(\left\lfloor \frac{J_{ij} + \varphi_{ijk}}{T_i} \right\rfloor * H(D - d_{ij}) \right) + \min \left(\left\lfloor \frac{t - \varphi_{ijk}}{T_i} \right\rfloor, \left\lfloor \frac{D - \varphi_{ijk} - d_{ij}}{T_i} \right\rfloor + 1 \right) \right] * C_{ij} \quad (4.7)$$

Finalmente, la contribución total del flujo de tareas Γ_i se calcula sumando las contribuciones de todas sus tareas:

$$W_{ik}(t, D) = \sum W_{ijk}(t, D), \forall j \in \Gamma_i \quad (4.8)$$

Cuando las relaciones de precedencia no se tienen en cuenta el análisis anterior es exacto. Sin embargo, para calcular el tiempo de respuesta de peor caso de la tarea τ_{ab} , se deben analizar todos los posibles instantes críticos construidos eligiendo una tarea de cada flujo de tareas. Excepto para sistemas muy simples, este análisis resulta intratable debido al alto número de combinaciones. Para resolver este problema, Tindell [Tin94] propuso un límite superior a la interferencia de las tareas de un flujo de tareas, que consiste en asumir la máxima interferencia que cada flujo de tareas puede causar, independientemente de la tarea usada para crear el instante crítico:

$$W_i^*(t, D) = \max(W_{ik}(t, D)), \forall k \in \Gamma_i \quad (4.9)$$

Usando esta función con cada flujo de tareas, se puede obtener el tiempo de respuesta de una tarea calculando solamente un caso. Esta solución lleva a un análisis pesimista en el que se obtiene un límite superior del tiempo de respuesta de peor caso. Para minimizar el pesimismo, esta función de interferencia máxima no se utilizará para el flujo de tareas de la tarea bajo análisis τ_{ab} . Por lo tanto, habrá que repetir el análisis tantas veces como tareas tenga el flujo de tareas Γ_a , cogiendo cada vez una tarea como la primera en activarse en el periodo de ocupación. Se utilizará el máximo tiempo de respuesta obtenido.

Como no conocemos el instante en el periodo de ocupación en el que se activa la tarea bajo análisis, tenemos que probar diferentes posibilidades. Debemos probar el inicio del periodo de ocupación, así como, todos los instantes

que provocan que el plazo de τ_{ab} coincida con el plazo de otra tarea. Representaremos las diferentes instancias (p) de las tareas con un número entero consecutivo. Cada número representa el orden de activación en el periodo de ocupación. Las activaciones en el Conjunto 2 tendrán números positivos, esto es, $p = 1$ para la activación en $(0, T_i]$, $p = 2$ para la activación en $(T_i, 2T_i]$, etc. Las activaciones en el Conjunto 1 tendrán valores $p \leq 0$, esto es, $p=0$ para la activación en $(-T_i, 0]$, $p=-1$ para la activación en $(-2T_i, -T_i]$, etc. De acuerdo a este esquema de numeración, la primera activación de la tarea τ_{ij} en el periodo de ocupación construido con la tarea τ_{ik} en el instante crítico, $p_{0,ijk}$, tiene el valor:

$$p_{0,ijk} = - \left\lfloor \frac{J_{ij} + \varphi_{ijk}}{T_i} \right\rfloor + 1 \quad (4.10)$$

Y la última activación en el periodo de ocupación, $p_{L,ijk}$:

$$p_{L,ijk} = \left\lceil \frac{L - \varphi_{ijk}}{T_i} \right\rceil \quad (4.11)$$

donde L es la longitud del periodo de ocupación calculado con la ecuación (4.19). Teniendo en cuenta las instancias en el periodo de ocupación, se puede obtener el conjunto de puntos a comprobar en el análisis, Ψ . Como las instancias del Conjunto 1 se activan al inicio del periodo de ocupación y los del Conjunto 2 se activan tan pronto como es posible, calcularemos los puntos a analizar de diferente manera, Ψ^- para el Conjunto 1 y Ψ^+ para el Conjunto 2:

$$\begin{aligned} \Psi^- &= \cup \{d_{ij}\} \forall j, k \in \Gamma_i, p_{0,ijk} \leq 0 \\ \Psi^+ &= \cup \{\varphi_{ijk} + (p - 1)T_i + d_{ij}\} \forall p = 1..p_{L,ijk}, \forall j, k \in \Gamma_i \\ \Psi &= \Psi^- \cup \Psi^+ \end{aligned} \quad (4.12)$$

Sabiendo que pueden ocurrir varias activaciones de la tarea τ_{ab} dentro del periodo de ocupación, debemos analizar todas ellas. En un periodo de ocupación creado con la tarea τ_{ac} , las activaciones a analizar son $p = [p_{0,abc}, p_{L,abc}]$. Si la primera activación en el periodo de ocupación, $p = 1$, se activa en el instante A , el tiempo de finalización de la activación p , $w_{abc}^A(p)$, se calcula sumando las contribuciones de todos los flujos de tareas:

$$\begin{aligned} w_{abc}^A(p) &= B_{ab} + (p - p_{0,abc} + 1)C_{ab} + W_{ac}^-(w_{abc}^A(p), D_{abc}^A(p)) + \\ &\quad + \sum_{\forall i \neq a} W_i(w_{abc}^A(p), D_{abc}^A(p)) \end{aligned} \quad (4.13)$$

4. Relaciones de precedencia en el análisis de planificabilidad para sistemas distribuidos de tiempo real con planificación EDF

donde W_{ac}^- es el resultado de la ecuación (4.8) para el flujo de tareas Γ_a sin la contribución de la tarea τ_{ab} y $D_{abc}^A(p)$ es el plazo de la instancia p cuando la primera activación, $p = 1$, ocurre en el instante A :

$$D_{abc}^A(p) = \begin{cases} d_{ab}, p \leq 0 \\ A + \varphi_{abc} + (p-1)T_a + d_{ab}, p > 0 \end{cases} \quad (4.14)$$

El tiempo de respuesta globalmente referenciado se calcula restando el instante de llegada del evento externo al tiempo de finalización:

$$R_{abc}^A(p) = w_{abc}^A(p) - A - \varphi_{abc} - (p-1)T_a + \phi_{ab} \quad (4.15)$$

Por contra, el tiempo de respuesta localmente referenciado es relativo al instante de activación de la instancia. Como el instante de activación varía para instancias del Conjunto 1 y 2, también lo hace la ecuación para calcular el tiempo de respuesta:

$$r_{abc}^A(p) = \begin{cases} w_{abc}^A(p), p \leq 0 \\ w_{abc}^A(p) - A - \varphi_{abc} - (p-1)T_a, p > 0 \end{cases} \quad (4.16)$$

Para cada instancia p únicamente es necesario analizar los valores de A en $[0, T_a]$, por que, de otra forma, estaríamos analizando activaciones de otra instancia. Esto significa que solo tenemos que analizar un subconjunto de Ψ :

$$\Psi^* = \begin{cases} d_{ab}, p \leq 0 \\ \Psi_x \in \Psi | \varphi_{abc} + (p-1)T_a + d_{ab} \leq \Psi_x < \varphi_{abc} + pT_a + d_{ab}, p > 0 \end{cases} \quad (4.17)$$

Por cada elemento en Ψ^* , el valor de A es:

$$A = \begin{cases} \Psi_x - d_{ab}, p \leq 0 \\ \Psi_x - [\varphi_{abc} + (p-1)T_a + d_{ab}], p > 0 \end{cases} \quad (4.18)$$

La longitud del periodo de ocupación se puede calcular considerando la contribución de todas las tareas sin tener en cuenta la restricción del plazo:

$$L_{abc} = W_{ac}(L_{abc}, D = \infty) + \sum_{\forall i \neq a} W_i^*(L_{abc}, D = \infty) \quad (4.19)$$

Para calcular el tiempo de respuesta de peor caso de la tarea τ_{ab} debemos calcular el valor máximo entre todas las instancia de la tarea τ_{ab} , todos los

instantes críticos creados a partir de las tareas de Γ_a , y todas las posibles coincidencias con plazos de otras tareas:

$$r_{ab} = \max \left(r_{abc}^A(p) \right) \forall p = p_{0,ijk} \cdot p_{L,ijk}, \forall c \in \Gamma_a, \forall A \in \Psi^* \quad (4.20)$$

Finalmente, si los tiempos de respuesta (tanto global o localmente referenciados) de todas las tareas son menores o iguales que los plazos asociados (global o localmente referenciados, respectivamente), el sistema es planificable.

4.3. Relaciones de precedencia en sistemas LC-EDF

En esta sección vamos a extender el análisis basado en *offsets* para sistemas distribuidos de tiempo real con planificación LC-EDF presentado en la sección 4.2, adaptándolo a la técnica de relaciones de precedencia resumida en la sección 2.3.5 y publicada en [PGH99].

En la sección 2.3.5, se han definido los conflictos de activación y las *H-sections* desde el punto de vista de las tareas (Definiciones 2.1 y 2.2). Recordar que dos tareas están en conflicto cuando sus ejecuciones son incompatibles en el mismo periodo de ocupación. Una *H-section* es un segmento de tareas contiguas de un flujo de tareas cuya prioridad es siempre más alta que la tarea bajo análisis. La definición de estos conceptos desde el punto de vista de la tarea se debe a que en la planificación FP todas las instancias p de una tarea τ_{ij} tienen la misma prioridad. Por contra, en la planificación EDF la prioridad de una instancia depende de su plazo absoluto: cuanto más cercano mayor prioridad tiene. Por tanto, cada instancia p de una tarea τ_{ij} tiene una prioridad diferente y debemos adaptar las definiciones a esta nueva situación, definiéndolas desde el punto de vista de la instancia.

Definición 4.1. Dos instancias de dos tareas diferentes del mismo flujo de tareas están en conflicto cuando sus ejecuciones son incompatibles en el mismo periodo de ocupación. Supongamos que τ_{ij}^p y τ_{ik}^p son dos instancias de las tareas τ_{ij} y τ_{ik} en la misma activación p del flujo de tareas Γ_i , con un plazo absoluto más cercano o igual que el plazo de la tarea bajo análisis τ_{ab} . Supongamos también que τ_{ij} precede a τ_{ik} , $j < k$. Si existe alguna otra

4. Relaciones de precedencia en el análisis de planificabilidad para sistemas distribuidos de tiempo real con planificación EDF

instancia intermedia τ_{il}^p ($j < l < k$) en la activación p del flujo de tareas, con plazo más lejano que τ_{ab} y alojada en el mismo procesador, entonces, en un periodo de ocupación de la tarea τ_{ab} , las instancias τ_{ij}^p y τ_{ik}^p no pueden interferir simultáneamente en el mismo periodo de ocupación. Por lo tanto, las activaciones de las instancias τ_{ij}^p y τ_{ik}^p están en conflicto.

Con el objetivo de detectar y resolver los conflictos de activación, las instancias del flujo de tareas Γ_i se clasifican en diferentes secciones llamadas *H-sections*. Una *H-section* es un conjunto de instancias de tareas contiguas (y en el mismo procesador que τ_{ab}) con plazo más cercano o igual que el de τ_{ab} .

Definición 4.2. Dos instancias τ_{ij}^p y τ_{ik}^p pertenecen a la misma *H-section*, para el análisis de τ_{ab} , si ambas se ejecutan en el mismo procesador que τ_{ab} con un plazo más cercano o igual que el de τ_{ab} y no hay instancias de tareas intermedias que se ejecutan en el mismo procesador y con un plazo más lejano que el de τ_{ab} . $H_{ij}^p(\tau_{ab})$ identifica la *H-section* a la que pertenece la instancia τ_{ij}^p y está compuesta por un conjunto de instancias de las tareas del flujo de tareas Γ_i que verifican:

$$\begin{aligned} H_{ij}^p(\tau_{ab}) = \{ & \tau_{il}^p \in \Gamma_i \mid (\nexists \tau_{ix}^p \mid [j \leq x \leq l \vee l \leq x \leq j]) \mid \\ & \text{proc}(\tau_{ix}) = \text{proc}(\tau_{il}) = \text{proc}(\tau_{ij}) = \text{proc}(\tau_{ab}) \wedge \\ & dl(\tau_{ix}^p) > D \wedge dl(\tau_{ij}^p) < D \wedge dl(\tau_{il}^p) < D \} \end{aligned} \quad (4.21)$$

donde D es el plazo absoluto de la tarea bajo análisis τ_{ab} y $dl(\tau_{ix}^p)$ es el plazo absoluto de la instancia p de la tarea τ_{ix} .

Si analizamos el flujo de tareas de la Figura 4.2, podemos ver que en la primera activación ($p = 0$) del flujo de tareas tenemos dos *H-sections* ($H_{i1}^0(\tau_{ab}) = \{\tau_{i1}^0\}$ y $H_{i3}^0(\tau_{ab}) = \{\tau_{i3}^0\}$). En cambio, para la segunda activación ($p = 1$) solo tenemos una ($H_{i1}^1(\tau_{ab}) = \{\tau_{i1}^1\}$). Por lo tanto, podemos decir que en la activación $p = 0$, las instancias τ_{i1}^0 y τ_{i3}^0 están en conflicto y solo una de ellas puede contribuir al periodo de ocupación de la tarea bajo análisis τ_{ab} ($\text{Max}(C_{i1}, C_{i3})$). En la activación $p = 1$ no hay conflictos entre las instancias porque solo hay una *H-section*, por lo que la contribución al periodo de ocupación es C_{i1} . Por lo tanto, la contribución máxima total al periodo de ocupación es:

$$W = \text{Max}(C_{i1}, C_{i3}) + C_{i1} + C_{ab} \quad (4.22)$$

Este resultado es el mismo que se obtiene calculando el máximo de los dos diferentes esquemas de ejecución en la Figura 4.2-b. Mientras que si aplicamos la técnica sin relaciones de precedencia, obtenemos un periodo de ocupación $W = 2C_{i1} + C_{i3} + C_{ab}$.

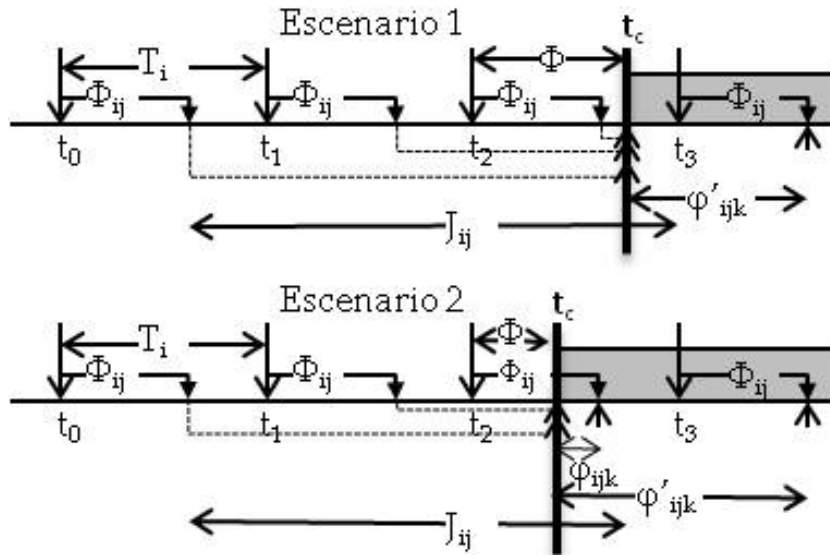


Figura 4.5: Escenarios para el cálculo de la contribución de la tarea τ_{ij} al periodo de ocupación con relaciones de precedencia

4.3.1. Interferencia de peor caso de un flujo de tareas

En esta sección definiremos la interferencia máxima que pueden generar las tareas de un flujo de tareas Γ_i en la ejecución de la tarea bajo análisis τ_{ab} . Para ello deberemos estudiar todos los posibles instantes críticos creados a partir de las tareas del flujo de tareas Γ_i . Por ahora nos centraremos en el estudio del instante crítico creado con la tarea τ_{ik} . Para calcular esta interferencia crearemos lo que llamamos una tabla de conflictos en la que se pueden identificar y analizar los conflictos entre las diferentes instancias. Para la construcción de dicha tabla, tenemos que asegurarnos que todas las instancias en una activación del flujo de tareas tienen el mismo índice p' . Por este motivo, vamos a cambiar la numeración de las instancias en el flujo de tareas, de forma que, la instancia $p' = 1$, en vez de ser la primera instancia activada en el periodo de ocupación, pasará a ser la instancia asociada al primer evento externo que llega dentro del periodo de ocupación. Si nos fijamos en el Escenario 2 de la Figura 4.5, siguiendo la numeración anterior la instancia $p = 1$ sería la instancia asociada al evento t_2 . En cambio, en la nueva numeración, el $p' = 1$ es el asociado al evento t_3 . Por lo que, para calcular las instancias que interfieren en el periodo de ocupación, se necesita una nueva magnitud, ϕ'_{ijk} que representa el periodo de tiempo entre el inicio del periodo

4. Relaciones de precedencia en el análisis de planificabilidad para sistemas distribuidos de tiempo real con planificación EDF

de ocupación y la activación de la instancia de la tarea correspondiente a la primera activación del inicio del flujo de tareas dentro del periodo de ocupación:

$$\varphi'_{ijk} = T_i - (\phi_{ik} + J_{ik}) \text{ mód } T_i + \phi_{ij} \quad (4.23)$$

Teniendo en cuenta esta numeración, la primera instancia de τ_{ij} que puede retrasar su activación para causar interferencia en el periodo de ocupación es:

$$p'_{0,ijk} = - \left\lfloor \frac{J_{ij} + \varphi'_{ijk}}{T_i} \right\rfloor + 1 \quad (4.24)$$

y la última instancia en el periodo de ocupación que puede interferir debido a que su plazo es más cercano que el de τ_{ab} es:

$$p'_{t,D,ijk} = \min \left(\left\lceil \frac{t - \varphi'_{ijk}}{T_i} \right\rceil, \left\lfloor \frac{D - \varphi'_{ijk} - d_{ij}}{T_i} \right\rfloor + 1 \right) \quad (4.25)$$

También necesitamos conocer cuáles de estas activaciones ocurren estrictamente dentro del periodo de ocupación con su mínimo *jitter* (Conjunto 2) y cuáles han retrasado su activación para coincidir con el inicio del periodo de ocupación (Conjunto 1). Como hemos visto en la sección anterior necesitamos tratarlas de diferente manera. Con el objetivo de identificar cuál es la primera instancia p' del Conjunto 2, nos fijaremos en la Figura 4.5. En el primer escenario, φ_{ijk} y φ'_{ijk} son iguales; lo que significa que $p' = 1$ es la primera instancia en el Conjunto 2. Si φ'_{ijk} es mayor que φ_{ijk} , como en el segundo escenario, esto significa que hay una o más instancias previas a $p' = 1$ que se han activado dentro del periodo de ocupación (Conjunto 2) y, consecuentemente, podemos calcular que la primera instancia en el Conjunto 1 es:

$$p'_{f,ijk} = - \left\lfloor \frac{\varphi'_{ijk} - \varphi_{ijk}}{T_i} \right\rfloor + 1 \quad (4.26)$$

Para calcular la contribución de peor caso del flujo de tareas Γ_i en un periodo de ocupación de una longitud t para el análisis de la tarea τ_{ab} , necesitamos crear una tabla para representar los conflictos entre las diferentes instancias. Cada fila de la tabla de conflictos representará un vector de las tareas pertenecientes al flujo de tareas Γ_i y cada columna representará un array de las diferentes instancias de cada tarea que pueden interferir en el periodo de ocupación. Cada celda (j, p') puede tomar uno de estos dos valores: 0 si la instancia p' de la tarea τ_{ij} no puede contribuir al periodo de ocupación o C_{ij} si puede.

4.3. Relaciones de precedencia en sistemas LC-EDF

Por otra parte, todas las instancias después del instante crítico, con índices $p' \geq 1$, deben ejecutarse en orden de precedencia empezando por la primera tarea y, por lo tanto, solo la primera *H-section* de cada instancia del flujo de tareas ($MP_i^{p'}(\tau_{ab})$) puede contribuir al periodo de ocupación:

$$MP_i^{p'}(\tau_{ab}) = \{\tau_{il}^p \in \Gamma_i \mid (\nexists \tau_{ix}^p \mid x < l \mid \text{proc}(\tau_{ix}) = \text{proc}(\tau_{il}) = \text{proc}(\tau_{ab}) \wedge dl(\tau_{ix}^p) > D \wedge dl(\tau_{il}^p) < D)\} \quad (4.27)$$

donde D es el plazo absoluto de la tarea bajo análisis τ_{ab} .

Se pueden reducir algunos conflictos por el hecho de que estamos analizando el instante crítico creado con la tarea τ_{ik} . Las instancias precedidas por la instancia $p'_{0,ikk}$ de la tarea τ_{ik} (creador del instante crítico) y en conflicto con ella, no pueden interferir. El proceso de creación de la tabla de conflictos se describe en el Algoritmo 4.1.

Algoritmo 4.1: construir _tabla_ conflictos _EDF

Input: τ_{ab} , D , t , Γ_i , τ_{ik}
Result: Tabla

```

1 begin
2   Inicializar _Tabla;
3   for  $\tau_{ij} \in \Gamma_i \mid \text{proc}(\tau_{ij}) = \text{proc}(\tau_{ab})$  do
4     if  $D > d_{ij}$  then
5        $p'_0 := p'_{0,ijk}$ ;
6     else
7        $p'_0 := p'_{f,ijk}$ ;
8     for  $p' \in p'_0 \cdot p'_{t,D,ijk}$  do
9        $Table[j, p'] := C_{ij}$ ;
10      if  $p' \geq 1$  and  $j \notin MP_i^{p'}(\tau_{ab})$  then
11         $Table[j, p'] := 0$ ;
12      if  $p' \geq p'_{0,ikk}$  and  $j > k$  and  $H_{ij}^{p'}(\tau_{ab}) \neq H_{ik}^{p'}(\tau_{ab})$  then
13         $Table[j, p'] := 0$ ;

```

Para resolver los conflictos de activación que hemos reflejado en la tabla es necesario recorrerla fila a fila, seleccionando en cada una la *H-section* con mayor contribución. La contribución de una *H-section* se calcula sumando los valores de las columnas correspondientes a las tareas que forman dicha *H-section*. La interferencia total se obtiene sumando los valores obtenidos en cada fila, como se muestra en el Algoritmo 4.2.

4. Relaciones de precedencia en el análisis de planificabilidad para sistemas distribuidos de tiempo real con planificación EDF

Algoritmo 4.2: resolver_conflictos_EDF

Input: $\tau_{ab}, D, t, \Gamma_i, \tau_{ik}$

Result: Total: tiempo

```

1 begin
2   construir_tabla_conflictos( $\tau_{ab}, D, t, \Gamma_i, \tau_{ik}$ ) ;
3   Total := 0;
4   for  $p' \in p'_{min} \cdot p'_{max}$  do
5     max_section,sum:= 0;
6     for  $\tau_{ij} \in \Gamma_i | proc(\tau_{ij}) = proc(\tau_{ab})$  do
7       if  $Table[j, p'] = 0$  then
8         | sum := 0 ;
9       else
10        |  $sum := sum + Table[j, p']$  ;
11        | if  $sum > max\_section$  then
12          | max_section := sum;
13      Total := Total + max_section ;

```

Teniendo en cuenta estos algoritmos, se puede obtener la interferencia de un flujo de tareas Γ_i a un periodo de ocupación de longitud t iniciado con un instante crítico creado por la tarea τ_{ik} y con un plazo de la tarea bajo análisis D :

$$W_{ik}(\tau_{ab}, t, D) = resolve_conflicts(\tau_{ab}, D, t, \Gamma_i, \tau_{ik}) \quad (4.28)$$

Para calcular el tiempo de respuesta de peor caso de la tarea τ_{ab} , debemos chequear todos los posibles instantes críticos construidos seleccionando una de las tareas de cada flujo de tareas. Como esto es intratable debido al gran número de combinaciones posible, aplicaremos el límite superior propuesto por Tindell [Tin94], que consiste en asumir la máxima interferencia de cada flujo de tareas:

$$W_i^*(\tau_{ab}, t, D) = max(W_{ik}(t, D)), \forall k \in \Gamma_i | proc(\tau_{ik}) = proc(\tau_{ab}) \quad (4.29)$$

4.3.2. Interferencia del flujo de tareas bajo análisis

En esta sección analizaremos las relaciones de precedencia y los conflictos de activación entre instancias de las tareas en el flujo de tareas Γ_a , al que

pertenece la tarea bajo análisis τ_{ab} . Para calcular la interferencia del flujo de tareas Γ_a , necesitamos estudiar todos los diferentes instantes críticos creados a partir de todas las tareas pertenecientes a Γ_a . En esta sección definiremos la ecuación para calcular la interferencia de las tareas de Γ_a a la tarea τ_{ab} en un periodo de ocupación iniciado con la activación con máximo *jitter* de la tarea τ_{ac} .

La creación de la tabla de conflictos es similar a la del algoritmo de la sección anterior (Algoritmo 4.1), pero en este caso podemos aplicar algunas reducciones a la tabla de conflictos. La primera razón para esta reducción es debida a la consideración de la tarea τ_{ac} como la iniciadora del periodo de ocupación. Por este motivo, por lo menos la instancia de la tarea τ_{ac} con la que se ha creado el instante crítico se ejecutará en el periodo de ocupación, y, por lo tanto, las instancias en conflicto con ésta pueden ser desechadas. La segunda razón es que la instancia de la tarea τ_{ab} que está bajo análisis tiene que ejecutarse dentro del periodo de ocupación. Consecuentemente, las instancias en conflicto con ésta se pueden eliminar. Además, teniendo en cuenta estas dos primeras restricciones, podemos reducir el número de instantes críticos a analizar si existe un conflicto de activación entre la instancia $\tau_{ik}^{p'}$ que inicia el periodo de ocupación y la instancia bajo análisis $\tau_{ab}^{p'}$. Por lo tanto, siguiendo estas restricciones, podemos aplicar las siguientes reglas de reducción a la tabla de conflictos:

1ª regla de reducción. Las activaciones correspondientes a $p'_{0,acc}$ y posteriores que entran en conflicto con instancias de la tarea τ_{ac} pueden ser eliminadas. Por lo tanto, podemos eliminar las celdas (j,p') que verifican:

$$p' \geq p'_{0,acc} \wedge j > c \wedge H_{aj}^{p'}(\tau_{ab}) \neq H_{ac}^{p'}(\tau_{ab}) \quad (4.30)$$

2ª regla de reducción. Las activaciones correspondientes a la instancia p'_{ab} y anteriores, incompatibles con la ejecución de $\tau_{ab}^{p'}$, pueden ser eliminadas. Por lo que se pueden eliminar las celdas (j,p') que verifican:

$$p' \leq p'_{ab} \wedge j < b \wedge H_{aj}^{p'}(\tau_{ab}) \neq H_{ab}^{p'}(\tau_{ab}) \quad (4.31)$$

Además de estas reducciones, podemos considerar una más debido a que las instancias de tareas precedidas por τ_{ab} no pueden interferir en la ejecución de la instancia p'_{ab} si corresponden a activaciones posteriores. Por lo tanto, podemos aplicar la siguiente regla:

4. Relaciones de precedencia en el análisis de planificabilidad para sistemas distribuidos de tiempo real con planificación EDF

3ª regla de reducción. Las activaciones correspondientes a tareas precedidas por τ_{ab} en la instancia p'_{ab} o posteriores, pueden ser eliminadas de la tabla de conflictos. Por lo que podemos eliminar las celdas (j, p') que verifican:

$$(p' \geq p'_{ab} \wedge j > b) \vee (p' > p'_{ab} \wedge j = c) \quad (4.32)$$

La tabla de conflictos para el flujo de tareas Γ_a puede ser construida con el algoritmo `construir_tabla_conflictos_en_Ga_EDF` (Algoritmo 4.3), muy similar al Algoritmo 4.1, pero aplicando las reglas de reducción anteriores.

Algoritmo 4.3: `construir_tabla_conflictos_en_Ga_EDF`

Input: $\tau_{ab}, p'_{ab}, D, t, \Gamma_a, \tau_{ac}$

Result: Tabla

```

1 begin
2   Inicializar_Tabla;
3   for  $\tau_{aj} \in \Gamma_a | \text{proc}(\tau_{aj}) = \text{proc}(\tau_{ab})$  do
4     if  $D > d_{aj}$  then
5       |  $p'_0 := p'_{0,ajc}$ ;
6     else
7       |  $p'_0 := p'_{f,ajc}$ ;
8     for  $p' \in p'_0 \cdot p'_{t,D,ajc}$  do
9       |  $\text{Table}[j, p'] := C_{aj}$  ;
10      if  $p' \geq 1$  and  $j \notin MP_i^{p'}(\tau_{ab})$  then
11        |  $\text{Table}[j, p'] := 0$ ;
12      if  $p' \geq p'_{0,acc}$  and  $j > c$  and  $H_{aj}^{p'}(\tau_{ab}) \neq H_{ac}^{p'}(\tau_{ab})$  then
13        |  $\text{Table}[j, p'] := 0$ ;
14      if  $p' \leq p'_{ab}$  and  $j < b$  and  $H_{aj}^{p'}(\tau_{ab}) \neq H_{ab}^{p'}(\tau_{ab})$  then
15        |  $\text{Table}[j, p'] := 0$ ;
16      if  $(p' \geq p'_{ab} \text{ and } j > b)$  or  $(p' > p'_{ab} \text{ and } j = c)$  then
17        |  $\text{Table}[j, p'] := 0$ ;

```

En la Figura 4.6, podemos ver un ejemplo de aplicación de las reglas de reducción. El sistema analizado cuenta con un flujo de tareas Γ_a en el que la tarea bajo análisis es la tarea τ_{a4} . La parte a) de la figura representa la creación de la tabla de conflictos para un instante crítico creado con la tarea τ_{a6} . Podemos ver una tabla con una columna por cada tarea de Γ_a que está alojada en el mismo procesador que τ_{a6} (tarea generadora del periodo de ocupación) y τ_{a4} (tarea bajo análisis). Cada fila representa una activación

4.3. Relaciones de precedencia en sistemas LC-EDF

a) Instante crítico creado con τ_{a6}

$$\begin{aligned}
 p'_{0,a16} = -1 \quad p'_{0,a46} = -1 \quad p'_{0,a66} = -2 \\
 p'_{0,a76} = 0 \quad p'_{0,a86} = -2
 \end{aligned}
 \Rightarrow
 \begin{array}{c}
 \begin{array}{cccccc}
 \tau_{a1} & \tau_{a3} & \tau_{a4} & \tau_{a6} & \tau_{a7} & \tau_{a8} \\
 \hline
 p' = -2 & 0 & 0 & 0 & C_{a6} & 0 & C_{a8} \\
 p' = -1 & C_{a1} & 0 & C_{a4} & C_{a6} & 0 & C_{a8} \\
 p' = 0 & C_{a1} & 0 & C_{a4} & C_{a6} & C_{a7} & C_{a8}
 \end{array}
 \end{array}$$

b) Resolución de conflictos en el análisis de la tarea τ_{a4}

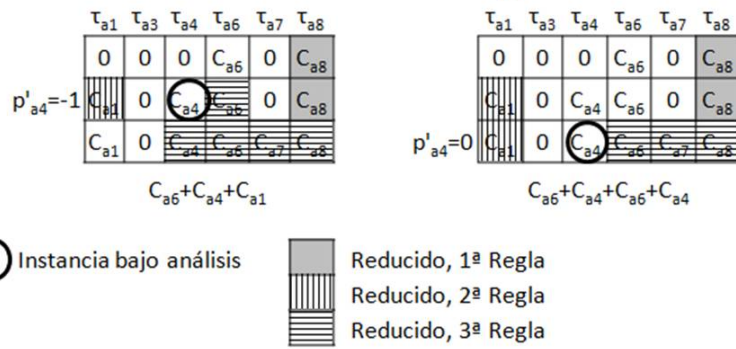


Figura 4.6: Ejemplo de las reglas de reducción.

diferente de Γ_a . La parte b) de la figura muestra la aplicación de las diferentes reglas de reducción para el análisis de la tarea τ_{a4} en el instante crítico de la parte a). Hay una tabla por cada instancia de la tarea τ_{a4} que hay que analizar. En este caso son solo dos instancias: $p'_{a4} = -1$ y $p'_{a4} = 0$. Las reducciones debidas a la primera regla están identificadas con un fondo gris, las de la segunda regla con un fondo rayado vertical y las de la tercera con un fondo rayado horizontal.

La resolución de conflictos se realiza mediante el algoritmo `resolver_conflictos_en_Gamma_a_EDF` (Algoritmo 4.4).

Finalmente, con la ayuda de estos algoritmos podemos calcular la contribución total del flujo de tareas Γ_a al periodo de ocupación de longitud t creado con la tarea τ_{ac} para el análisis de la instancia p'_{ab} de la tarea τ_{ab} :

$$W_{ac}(\tau_{ab}, p'_{ab}, t, D) = \text{resolve_conflicts_in_}\Gamma_a_EDF(\tau_{ab}, p'_{ab}, D, t, \Gamma_a, \tau_{ac}) \quad (4.33)$$

En este punto ya podemos calcular el tiempo de finalización $w_{abc}(p'_{ab})$ de

4. Relaciones de precedencia en el análisis de planificabilidad para sistemas distribuidos de tiempo real con planificación EDF

Algoritmo 4.4: resolver_conflictos_en_ Γ_a _EDF

Input: $\tau_{ab}, p'_{ab}, D, t, \Gamma_a, \tau_{ac}$

Result: Total: tiempo

```

1 begin
2   construir_tabla_conflictos_en_ $\Gamma_a$ _EDF( $\tau_{ab}, p'_{ab}, D, t, \Gamma_a, \tau_{ac}$ ) ;
3   Total := 0;
4   for  $p' \in p'_{min} \cdot p'_{max}$  do
5     max_section,sum:= 0;
6     for  $\tau_{aj} \in \Gamma_a | proc(\tau_{aj}) = proc(\tau_{ab})$  do
7       if Table[j,  $p'$ ] = 0 then
8         sum := 0 ;
9       else
10        sum := sum + Table[j,  $p'$ ] ;
11        if sum > max_section then
12          max_section := sum;
13      Total := Total + max_section ;

```

la activación p'_{ab} :

$$w_{abc}(p'_{ab}) = B_{ab} + W_{ac}(\tau_{ab}, p'_{ab}, w_{abc}(p'_{ab}), D) + \sum_{\forall i \neq a} W_i^*(\tau_{ab}, w_{abc}(p'_{ab}), D) \quad (4.34)$$

Usando esta ecuación, podemos calcular el tiempo de respuesta de la tarea τ_{ab} siguiendo el proceso que se ha utilizado en la sección 4.2 para la técnica análisis basada en *offsets*. Para ello, se pueden usar las ecuaciones (4.10) a (4.20).

4.4. Resultados de simulación

En esta sección compararemos los resultados obtenidos con las técnicas de análisis anteriormente existentes con los obtenidos mediante los algoritmos presentados en esta sección. Esta comparación se hace teniendo como punto de partida la técnica presentada por Rivas et al. en [RGPGH10], que es una variante para sistemas LC-EDF de la presentada por Spuri en [Spu96b]. En

4.4. Resultados de simulación

esta técnica, las tareas son analizadas como si se tratasen de tareas independientes. Realizaremos la comparativa con las técnicas:

- WCDO-EDF (*Worst Case Dynamic Offset for EDF*, Peor Caso con Intervalos de Desfase (Offsets) Dinámicos para EDF): Técnica descrita en la sección 4.2 que utiliza los *offsets* dinámicos para reducir el pesimismo.
- WCDOPS-EDF (*Worst Case Dynamic Offset Precedence Schemes for EDF*, Peor Caso con Intervalos de Desfase (Offsets) Dinámicos y con Esquemas de Precedencia para EDF): Técnica descrita en la sección 4.3 que utiliza las relaciones de precedencia para reducir el pesimismo.

Con el objetivo de hacer una comparativa lo más completa posible, se ha realizado un número extenso de simulaciones con diferentes conjuntos de tareas usando el generador GEN4MAST [RGGH13]. Este generador permite generar automáticamente y de forma aleatoria sistemas de tareas preparados para ser analizados mediante la herramienta MAST [MAS]. Los nuevos algoritmos WCDO-EDF y WCDOPS-EDF han sido implementados en una versión interna de la herramienta MAST. Los resultados de las simulaciones se muestran en esta sección.

El primer conjunto de gráficos (Figura 4.7 y Figura 4.8) compara los tiempos de respuesta obtenidos con la técnica holística de Rivas et al. [RGGH14], R_{indep} , con el tiempo de respuesta obtenido usando el algoritmo WCDO-EDF, $R_{WCDO-EDF}$, y el algoritmo WCDOPS-EDF, $R_{WCDOPS-EDF}$. En estos gráficos se muestran los ratios medios de tiempo de respuesta $R_{indep}/R_{WCDO-EDF}$ y $R_{indep}/R_{WCDOPS-EDF}$ obtenidos a partir de veinte sistemas de tareas aleatorios para cada punto de la gráfica. En cada gráfica, los resultados son representado para tres ratios diferentes entre el máximo y el mínimo periodo de flujo de tareas, $\delta = T_{max}/T_{min}$. Se han elegido valores de $\delta = 10, 100, 1000$ para cubrir sistemas con periodos más homogéneos y más dispersos. El tiempo de respuesta de mejor caso es considerado cero, por lo que los *offsets* iniciales son nulos también. Los plazos de los flujos de tareas tienen valores máximos $D = N * T$, siendo N el número de tareas del flujo de tareas. Los plazos de planificación locales de las tareas son asignados mediante el algoritmo PD propuesto en [Liu00].

La Figura 4.7 muestra los resultados para sistemas formados por 10 flujos de tareas con 10 tareas por cada flujo de tareas y un único procesador. Se

4. Relaciones de precedencia en el análisis de planificabilidad para sistemas distribuidos de tiempo real con planificación EDF

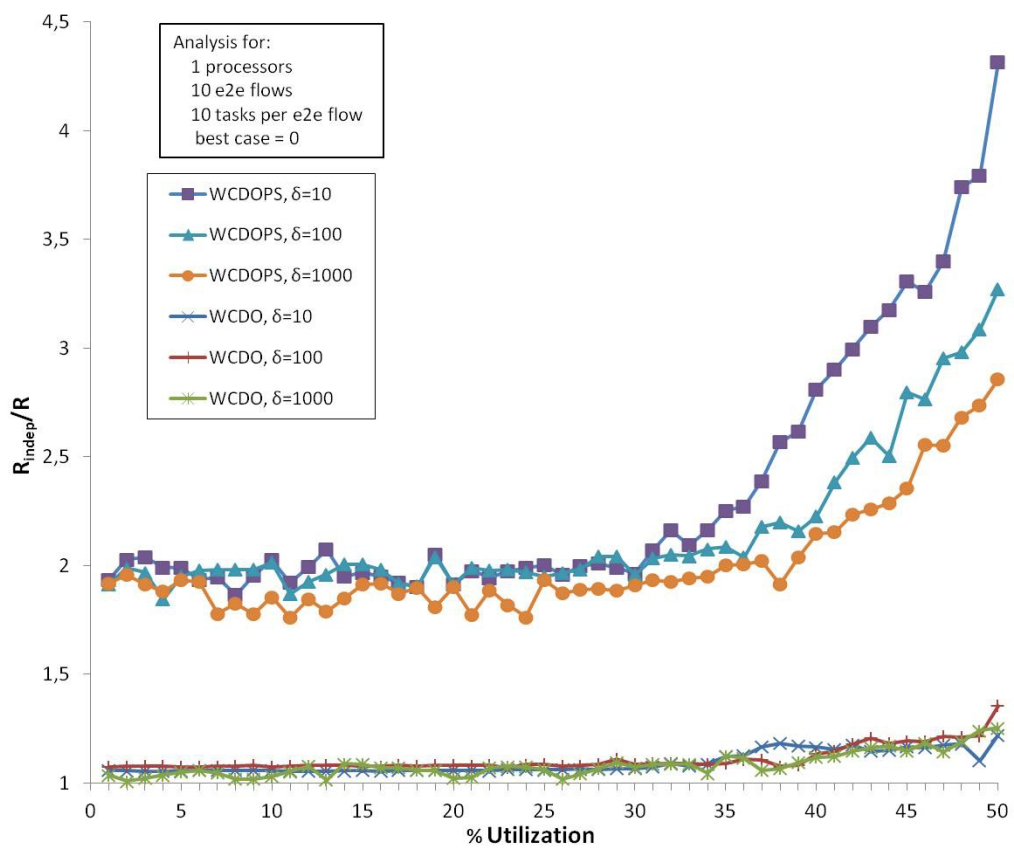


Figura 4.7: Comparación de tiempos de respuesta para las diferentes técnicas en un procesador.

4.4. Resultados de simulación

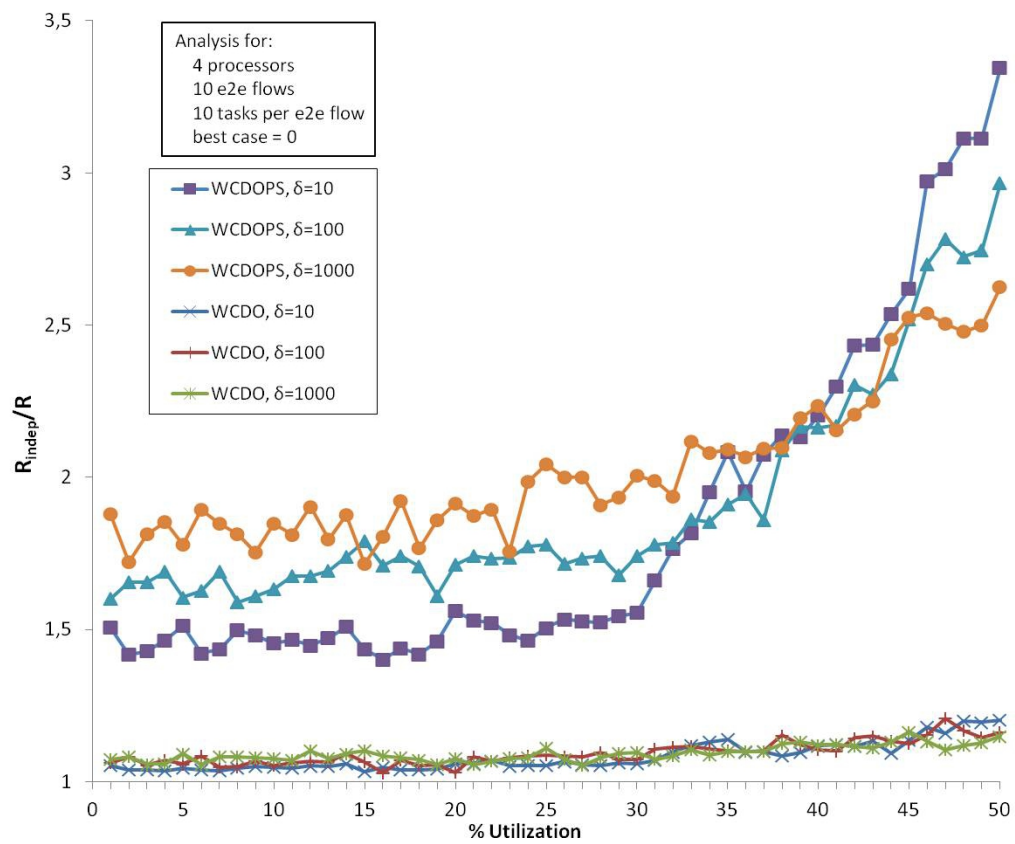


Figura 4.8: Comparación de tiempos de respuesta para las diferentes técnicas en cuatro procesadores.

4. Relaciones de precedencia en el análisis de planificabilidad para sistemas distribuidos de tiempo real con planificación EDF

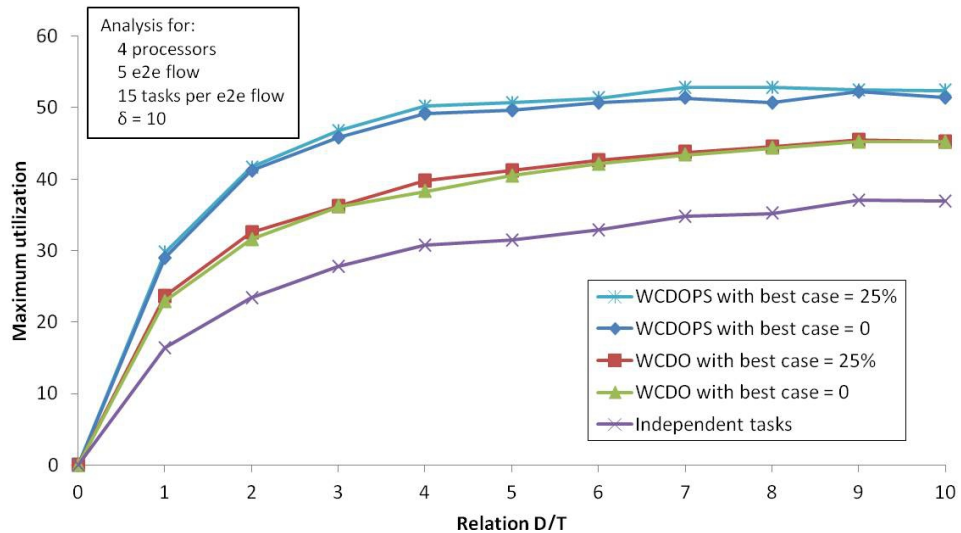


Figura 4.9: Utilización máxima planificable, 65 tareas.

puede observar que para niveles de utilización alrededor de 40-50 %, el tiempo de respuesta R_{indep} es entre 1.1 y 1.3 veces más grande que el calculado mediante el algoritmo con *offsets* dinámicos (WCDO-EDF), $R_{WCDO-EDF}$. Este resultado es mayor, entre 2.2 y 4.3 veces, para el análisis con relaciones de precedencia (WCDOPS-EDF), $R_{WCDOPS-EDF}$.

La Figura 4.8 muestra los resultados de un caso similar, pero en este caso el sistema cuenta con 4 procesadores. Las tareas son asignadas a los procesadores aleatoriamente por la herramienta GEN4MAST. Las técnicas de análisis WCDO-EDF y WCDOPS-EDF se ven beneficiadas por el alto número de tareas en el mismo procesador, y como en este caso han decrecido respecto al de la Figura 4.7, los beneficios de estos algoritmos también decrecen. De todas formas, se puede observar que los beneficios continúan siendo significativos, con tiempos de respuesta entre 1.1 y 1.2 veces mejores para WCDO-EDF y entre 2.1 y 3.3 para WCDOPS-EDF en utilizaciones del 40-50 %. Esto supone una ventaja significativa para la técnica basada en *offsets* que reduce el pesimismo del tiempo de respuesta entre un 10 % y un 20 %. El beneficio de la técnica de precedencias es destacable, dividiendo a la mitad o incluso a un tercio el tiempo de respuesta.

El segundo conjunto de gráficos (Figura 4.9 y Figura 4.10) representa la utilización máxima planificable que el análisis puede asegurar para un conjunto de tareas. En los gráficos podemos ver cinco líneas de puntos diferentes:

4.4. Resultados de simulación

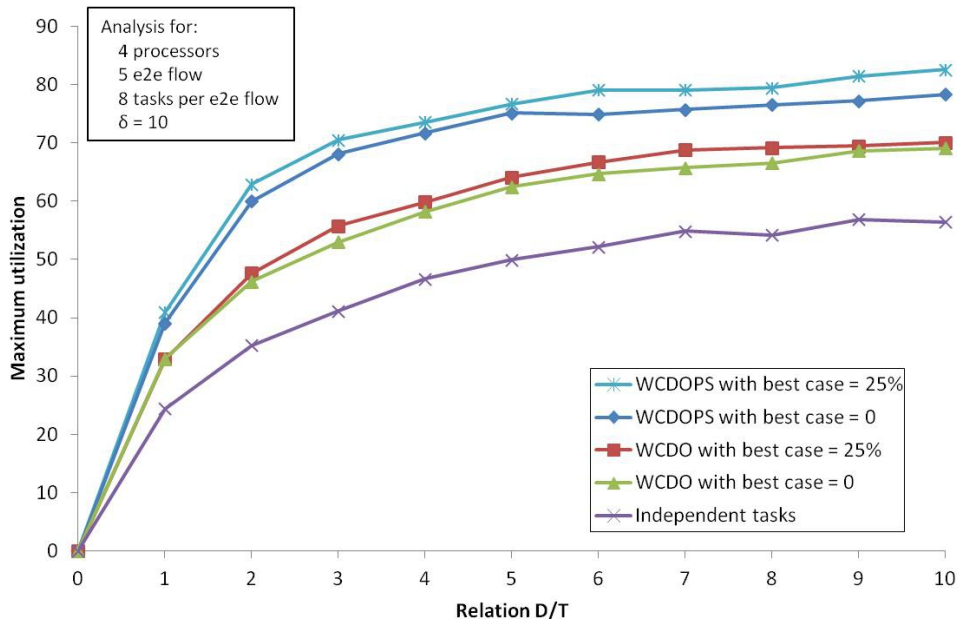


Figura 4.10: Utilización máxima planificable, 40 tareas.

una para el análisis holístico con tareas independientes; dos para el algoritmo WCDO-EDF, con tiempos de ejecución de mejor caso igual a cero e igual al 25 % del tiempo de ejecución de peor caso; y otras dos para el algoritmo WCDOPS-EDF, con los mismos casos de tiempos de ejecución de mejor caso.

La utilización máxima se calcula iniciando el análisis con una utilización baja, y subiéndola hasta que el sistema deje de ser planificable. La máxima utilización es la última que hace que el sistema cumpla los plazos de las tareas y, por lo tanto, da un resultado del análisis de planificación positivo. La simulación se ha realizado con una ratio $\delta = T_{max}/T_{min} = 10$ entre el máximo y el mínimo periodo y con diferentes ratios entre el plazo y el periodo (D/T).

La Figura 4.9 muestra los resultados para un sistema con 4 procesadores, 5 flujos de tareas y 15 tareas por cada flujo de tareas. La Figura 4.10 muestra los resultados para un sistema similar, pero con 8 tareas por flujo de tareas en vez de 15. En la Figura 4.9, podemos ver que para valores $D/T = 2$ y mayores, se puede conseguir un incremento en la utilización alrededor del 9 % para WCDO-EDF y 16 % para WCDOPS-EDF. En la Figura 4.10, podemos ver una mejora de alrededor de un 11 % para WCDO-EDF y 24 % para WCDOPS-EDF. En la Figura 4.9, si tomamos $D/T = 4$, WCDO-EDF

4. Relaciones de precedencia en el análisis de planificabilidad para sistemas distribuidos de tiempo real con planificación EDF

mejora los resultados de un 30 % a un 38 % y WCDOPS-EDF de un 30 % a 49 %. Estos datos implican una significativa mejora relativa en la utilización máxima planificable de un 27 % y un 63 % respectivamente.

También es importante mencionar que la utilización máxima planificable mejora aún considerando tiempos de ejecución de mejor caso iguales a cero, pero mejora incluso más cuando los tiempo de ejecución de mejor caso son mayores que cero.

4.5. Conclusiones

En este capítulo hemos desarrollado dos técnicas de análisis de planificabilidad para flujos de tareas planificados con la política LC-EDF, para sistemas distribuidos o sistemas multiprocesador particionados. Al contrario que la política GC-EDF, LC-EDF no requiere de mecanismos de sincronización de relojes y, por lo tanto, es aplicable a sistemas que no ofrecen esta funcionalidad.

La primera de las contribuciones en este capítulo introduce el análisis basado en *offsets* para LC-EDF. Este análisis es capaz de reducir el pesimismo del análisis anterior para este tipo de planificación, que está basado en la técnica holística donde se supone que las tareas se activan de forma independiente a las demás tareas. Añadir *offsets* al modelo de tareas ayuda a modelar las relaciones temporales entre las activaciones de las diferentes tareas de un flujo de tareas y, usándolos en el análisis, el pesimismo se reduce significativamente. Los resultados de nuestros experimentos usando casos extensos muestran que la utilización máxima planificable de la CPU (*Central Processing Unit*, Unidad de Procesamiento Central) puede incrementarse entre un 9 % y un 11 % adicional.

La segunda contribución consiste en explotar los beneficios de analizar las relaciones de precedencia entre las tareas para minimizar el pesimismo del algoritmo de análisis aún más. Los resultados de nuestra simulación muestran que los beneficios son destacables. Con esta nueva técnica, las estimaciones de tiempos de respuesta de peor caso decrecen considerablemente. Los valores de tiempos de respuesta obtenidos se reducen a la mitad o a un tercio de los obtenidos por la técnica existente. Además la utilización máxima planificable aumenta entre un 16 % y un 24 %.

4.5. Conclusiones

Es importante mencionar que estas mejoras no suponen ninguna penalización a la ejecución y comportamiento del sistema, ya que únicamente afectan a la técnica de análisis y no a la política de planificación, ni a la arquitectura del sistema. Aún así es posible alcanzar la máxima mejora cuando se miden los tiempos de ejecución de mejor caso y se introducen en el modelo de análisis.

Tras estas contribuciones, el cuadro de técnicas para el cálculo de tiempos de respuesta de peor caso en sistemas distribuidos queda como se puede ver en la Tabla 4.2.

Tabla 4.2: Técnicas para cálculo de tiempos de respuesta de peor caso (II).

Técnica	FP	GC-EDF	LC-EDF
Holística	[THW94]	[Spu96b]	[RGPGH10]
WCDO	[PGH98]	[PGH03]	[DdCUGHP14a]
WCDOPS	[PGH99]	No aplicable	[DdCUGHP14a]
Slanted	[MTN08]	-	-
Slanted+PS	-	No aplicable	-

Los algoritmos se han implementado en una copia privada de la herramienta de análisis MAST [MAS], que será publicada en futuras versiones públicas de la herramienta.

Como trabajo futuro se plantea hacer una comparación extensa entre las nuevas técnicas presentadas aquí y la técnica basada en *offsets* para GC-EDF.

Por otra parte, en [MTN08] Mäki-Turja y Nolin introdujeron una nueva mejora a la técnica basada en *offsets* para FP con objeto de reducir el pesimismo del algoritmo. Como esta mejora no ha sido todavía desarrollada para EDF, no la hemos tenido en cuenta. Dejamos como trabajo futuro la extensión de este trabajo a EDF y la adaptación al método basado en relaciones de precedencia. Las técnicas desarrolladas en esta sección podrán servir como punto de partida de estas extensiones. Además, en [Red04] Redell introdujo una pequeña mejora al análisis basado en relaciones de precedencia para FP. Esta mejora no la hemos tenido en cuenta y, ya que podría suponer una mejora del algoritmo, se plantea también como trabajo futuro.

Por último, como se ha comentado durante el capítulo los test exactos desarrollados requieren de mucha carga de computación para analizar un sis-

4. Relaciones de precedencia en el análisis de planificabilidad para sistemas distribuidos de tiempo real con planificación EDF

tema complejo. Por ello se aplican aproximaciones que reducen ese tiempo de computo. Sería interesante como trabajo futuro la búsqueda de test exactos más rápidos que los actuales.

Capítulo 5

Análisis de planificabilidad de flujos de tareas con secciones atómicas

RESUMEN:

La mayoría de los algoritmos propuestos para el análisis de planificabilidad con la técnica de tiempos de respuesta están basados en un modelo de flujo de tareas lineal. Cada flujo de tareas está compuesto por un grupo de tareas con una relación de precedencia uno-a-uno, lo que significa que cada tarea tiene como mucho una tarea predecesora y una sucesora. Por otra parte, una instancia del flujo de tareas no tiene relación de precedencia con la siguiente instancia del flujo de tareas. Por tanto, dos instancias de una tarea pueden estar activas al mismo tiempo. Esto supone una restricción para los diseñadores de sistemas de tiempo real a la hora de modelar sus sistemas, ya que hay sistemas donde no es posible esta simultaneidad en la activación de tareas. Por ejemplo, una llamada remota a procedimiento causa que el thread de la tarea invocante se bloquee en espera de la finalización de la llamada en el nodo remoto y, por tanto, que este thread no esté disponible para procesar la próxima instancia de la tarea. El uso de las técnicas de análisis convencionales para sistemas con llamadas remotas a procedimientos conduciría a resultados de tiempos de respuesta incorrectos. En [Gut95] y [HHJ⁺05], el modelo lineal es extendido para permitir relaciones de precedencia más flexibles. En este capítulo se presenta un nuevo concepto llamado sección atómica de tareas, que permitirá a los diseñadores un modelo más amplio para diseñar sus sistemas de tiempo real. Una sección atómica de tareas permite a los diseñadores

representar un fragmento de un flujo de tareas, en el que dos instancias de cualquier tarea de dicha sección no pueden estar activas simultáneamente. Además, se van a introducir algunos cambios en el análisis de tiempos de respuesta presentado en el capítulo anterior, teniendo en cuenta las relaciones de precedencia originadas por las secciones atómicas, con el objetivo de reducir el pesimismo del análisis lo máximo posible. Aplicaremos estas mejoras a sistemas distribuidos con planificación LC-EDF, pero la técnica aquí presentada puede ser fácilmente extensible a otras políticas de planificación.

El capítulo se organiza de la siguiente manera. En la sección 5.1 introduciremos el nuevo concepto de sección atómica de tareas y estudiaremos el nuevo modelo resultante. En segundo lugar, en la sección 5.2 veremos modelos alternativos al modelo lineal presentados en otros trabajos. Después, en la sección 5.3 describiremos el análisis de planificación teniendo en cuenta el nuevo modelo con secciones atómicas. En la sección 5.4, mostraremos un ejemplo al que le aplicaremos el análisis con secciones atómicas para comparar los resultados con la técnica de análisis sin secciones atómicas. Por último en la sección 5.5 analizaremos las conclusiones obtenidas durante el desarrollo de este trabajo.

5.1. Modelo del sistema con flujo de tareas con secciones atómicas

En este capítulo se introduce una modificación sobre el modelo general de tareas presentado en la sección 1.2. El modelo general permite que dos instancias de una tarea puedan estar activas al mismo tiempo. Si el flujo de tareas tiene un tiempo de respuesta superior al periodo, antes de que la última tarea del flujo complete su ejecución, la primera tarea puede haber sido activada por el evento correspondiente al siguiente periodo. Esto se tiene en cuenta en las técnicas de análisis de tiempos de respuesta, pero hay algunos casos en los que este modelo no refleja el patrón de ejecución real. Esto ocurre cuando varias tareas del flujo comparten el mismo thread de ejecución.

Considerar, por ejemplo, el caso común de un thread que se suspende a sí mismo en una operación de espera en medio de su ejecución. Algunas veces esto se hace para adaptar la ejecución a los requerimientos temporales del

entorno externo o para controlar el *jitter* de salida [BW09]. En el modelo de flujos de tareas se dividiría este thread en dos tareas de análisis y se modelaría el retraso mediante un *offset* entre ambas tareas. En el modelo de análisis convencional es posible que ambas tareas estén activas en el mismo instante, y por lo tanto el análisis tendría en cuenta un potencial efecto de desalajo entre ambas tareas. En la práctica este desalajo no puede darse porque ambas tareas son un artefacto usado para modelar el mismo thread. Generalmente, el thread no puede iniciar la ejecución hasta que la instancia anterior haya finalizado su ejecución. El modelo de análisis no refleja la realidad y reportará resultados erróneos.

Otro ejemplo muy común de este problema es una llamada bloqueante (una llamada remota bloqueante) en un sistema distribuido. Suponer un thread que realiza una llamada a un procedimiento remoto para invocar un servicio en un procesador remoto. El thread ejecuta una parte inicial antes de la llamada, luego realiza la llamada suspendiéndose a sí mismo a la espera de la respuesta del nodo remoto, y cuando recibe la respuesta, continúa ejecutando la parte final. Este thread y la llamada remota se modelan como en la Figura 1.4 del capítulo 1, usando un flujo de tareas con 5 elementos: una tarea para la parte inicial del thread, un mensaje de salida, una tarea en el procesador remoto, un mensaje de entrada y otra tarea para modelar la parte final del thread. El problema es que las dos tareas ejecutadas en el procesador 1 pueden activarse en el mismo instante, y la técnica de análisis convencional tendría en cuenta la interferencia entre ellas. Esto genera que el análisis sea incorrecto debido a que en la práctica la tarea inicial no se puede ejecutar hasta que la tarea final haya finalizado la ejecución de la instancia anterior.

Es posible usar construcciones de programación que permiten a las tareas realizar operaciones de espera y llamadas a procedimientos remotos para que se comporten como en el modelo usado para el análisis. Por tanto, el modelo no es incorrecto pero es incompleto. Es necesario poder especificar en el modelo de análisis que dos tareas que pertenecen al mismo flujo de tarea y que en la práctica se ejecutan en el mismo thread y son atómicas en el sentido de que la primera tarea no puede ejecutar una nueva instancia hasta que la segunda no haya completado la ejecución de la instancia actual. La Figura 1.5 del capítulo 1 muestra dos maneras alternativas de programar una llamada a procedimiento remoto en un thread de tiempo real. La construcción de la Figura 1.5-a del capítulo 1 sería correctamente analizada por las técnicas de análisis convencionales, pero la construcción más simple de la Figura 1.5-b del capítulo 1 daría resultados incorrectos.

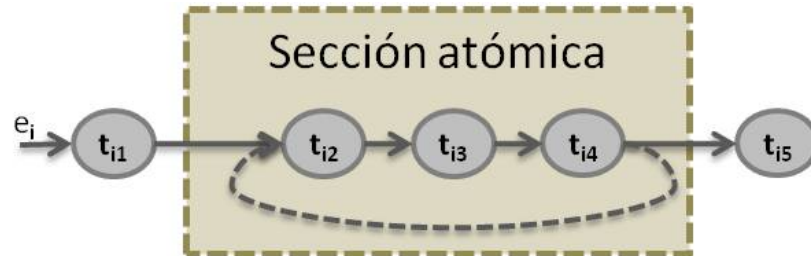


Figura 5.1: Nueva relación de precedencia en el flujo de tareas debido al efecto de la sección atómica de tareas

En esta sección introducimos un nuevo término, llamado sección atómica de tareas, que permite modelar este concepto.

Definición 5.1. Una sección atómica de tareas representa un fragmento de un flujo de tareas (compuesto por una o varias tareas contiguas), en el cual dos instancias de cualquier tarea de la sección no pueden estar activas en el mismo instante de tiempo. En otras palabras, una instancia de la primera tarea del flujo de tareas no puede activarse antes de que todas las tareas de la instancia anterior del flujo de tareas que están en la sección atómica hayan terminado su ejecución.

En la Figura 5.1 se puede ver un ejemplo de un flujo de tareas con cinco tareas, en el que tres de ellas son parte de una sección atómica de tareas (desde τ_{i2} a τ_{i4}). En este ejemplo, la instancia p de la tarea τ_{i2} no puede activarse hasta que la instancia $p - 1$ de la tarea τ_{i4} haya finalizado. Esta nueva característica genera una nueva relación de precedencia entre la última y la primera tarea de la sección atómica. Esta nueva relación se puede ver representada en la Figura 5.1, con una flecha discontinua desde τ_{i4} a τ_{i2} .

Esta extensión del modelo permite cubrir aplicaciones con flujos de tareas, en las cuales una tarea (τ_{i1} en la Figura 5.1) recoge instancias del evento externo y a su vez encola eventos internos, que son procesados de forma atómica por las siguientes tareas (desde τ_{i2} a τ_{i4} en la Figura 5.1). Además, puede haber otras tareas después de la sección atómica de tareas que no tienen por qué ser ejecutadas atómicamente (por ejemplo, τ_{i5} en la Figura 5.1).

Debido a la dificultad que supone analizar las dependencias de un flujo de tareas en el que hay varias secciones atómicas, el análisis propuesto se limita a sistemas con flujos de tareas en los que como mucho haya una sección

atómica de tareas. Esto es una limitación de nuestra técnica de análisis que se propone eliminar como parte del trabajo posterior a esta tesis. Aún con esta limitación, el modelo propuesto es útil para analizar muchos sistemas de tiempo real prácticos.

5.2. Alternativas al modelo lineal

Generalmente, la mayor parte de trabajos sobre análisis de planificabilidad presentan un modelo de flujo de tareas lineal como el mostrado en la sección 1.2. Esto significa que una tarea es activada por un solo evento y solo puede activar otro evento. Pero hay algunos trabajos que extienden este modelo para romper con esta linealidad. En [Gut95], se presentan diferentes patrones de eventos de activación que permiten modelar sistemas más complejos que lo que permite el modelo de flujo de tareas lineal. Además, se demostró que estos patrones pueden modelarse mediante un modelo lineal equivalente.

Aunque estos nuevos patrones extienden el modelo lineal, no permiten modelar el concepto de sección atómica de tareas dentro del flujo de tareas, presentado en la sección anterior.

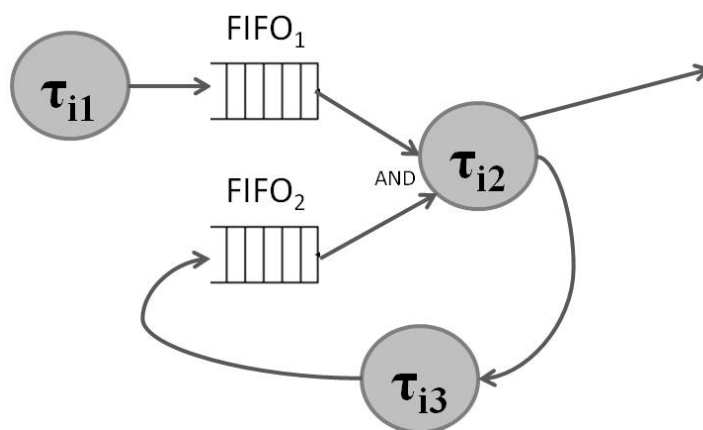


Figura 5.2: Patrón cíclico

En Henia et al.[HHJ⁺05], se presentan algunos patrones similares a los presentados en [Gut95], pero además se presenta un patrón de dependencia de

tareas cíclicas, como el representado en la Figura 5.2. Este patrón es similar al introducido en este capítulo. En este trabajo establecen cómo calcular el *jitter* de la tarea que sufre la restricción de precedencia múltiple (τ_{i2}) con la siguiente ecuación:

$$J_{AND} = \max(JP_i); i = 1..k \quad (5.1)$$

Siendo JP_i el *jitter* producido por una de las tareas precedentes ($i = 1..k$). En el ejemplo de la Figura 5.2 el *jitter* de la tarea τ_{i2} es igual a:

$$J_{i2} = \max(JP_{i1}, JP_{i3}) \quad (5.2)$$

Esta propuesta para calcular el *jitter* es similar a la que se propone en este trabajo en las ecuaciones (5.3) y (5.4) de la siguiente sección, aunque en el trabajo de Henia et al. no se tenían en cuenta los *offsets* de las tareas. No obstante, la mayor diferencia entre esta propuesta y la que se hace en nuestro trabajo consiste en que en nuestra propuesta la técnica de análisis tiene en cuenta las relaciones de precedencia generadas por la sección atómica de tareas para reducir el pesimismo del análisis, lo que puede conducir a resultados más ajustados.

5.3. Análisis de planificabilidad

En esta sección describiremos cómo afectan las secciones atómicas de tareas al análisis de tiempo de respuesta de peor caso de las tareas y cómo se aprovecha este efecto para reducir el pesimismo del análisis. En primer lugar, en la sección 5.3.1, se analizan las diferencias en el cálculo del *offset* y del *jitter* de las tareas para el nuevo modelo. Después, en la sección 5.3.2, presentaremos un método para calcular la interferencia que un flujo de tareas puede generar en una tarea bajo análisis teniendo en cuenta las secciones atómicas de tareas contenidas en ese flujo de tareas.

5.3.1. Calculando *offset* y *jitter* de las tareas

En métodos previos basados en *offsets*, como [PGH98], [PGH99], [PGH05] y [DdCUGHP14a], en los que el modelo de flujo de tareas es lineal, el *offset* y el *jitter* de una tarea τ_{ij} dependen del tiempo de respuesta de peor caso

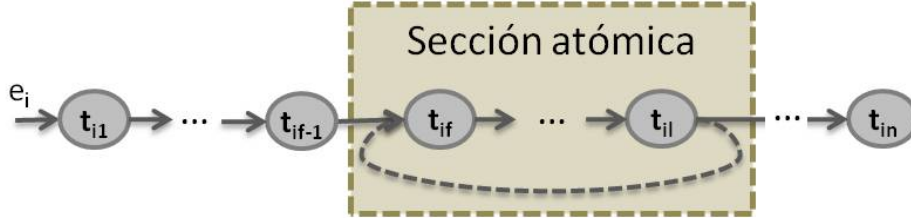


Figura 5.3: Modelo genérico de flujo de tareas con una sección atómica de tareas

y de mejor caso de la tarea precedente (R_{ij-1} y R_{ij-1}^b respectivamente) y se calculan mediante las siguientes ecuaciones:

$$\phi_{ij} = \max(\Phi_{ij}, R_{ij-1}^b) \quad (5.3)$$

$$J_{ij} = R_{ij-1} - \phi_{ij} \quad (5.4)$$

siendo Φ_{ij} el *offset* mínimo de la tarea τ_{ij} .

En el nuevo modelo con secciones atómicas de tareas (En la Figura 5.3 se muestra el modelo genérico), las ecuaciones para calcular los *offsets* y *jitters* cambian. En el modelo genérico podemos ver las tareas τ_{i1} (primera tarea del flujo de tareas), τ_{in} (última tarea del flujo de tareas), τ_{if} (primera tarea en la sección atómica de tareas), τ_{il} (última tarea en la sección atómica de tareas) y τ_{if-1} (la tarea anterior a la primera en la sección atómica de tareas). Como indican los puntos suspensivos, puede haber más tareas entre estas. La única tarea cuyas relaciones de precedencia varían, debido al efecto de la sección atómica de tareas, es la tarea τ_{if} . Por lo tanto, para el resto de tareas continuaremos usando las ecuaciones (5.3) y (5.4).

A continuación, definimos las ecuaciones para calcular el *offset* y *jitter* de la primera tarea en la sección atómica de tareas (τ_{if}).

Teorema 5.1. *En un sistema compuesto por flujos de tareas con un máximo de una sección atómica de tareas por cada flujo, el offset y el jitter de la primera tarea en la sección atómica de tareas τ_{if} se calculan mediante las siguientes ecuaciones:*

$$\phi_{if} = \max(\Phi_{if}, R_{if-1}^b) \quad (5.5)$$

$$J_{if} = \max(R_{if-1}, R_{il} - T_i) - \phi_{if} \quad (5.6)$$

Demostración. Inicialmente supondremos que el *offset* mínimo Φ_{if} es igual a cero, ya que su utilización no es necesaria para la demostración. Posteriormente eliminaremos esta restricción. Como se puede ver en la Figura 5.3, la tarea τ_{if} tiene dos eventos de entrada: uno que viene de la tarea τ_{if-1} y otro que viene de la instancia anterior de la tarea τ_{il} . La tarea solo puede activarse cuando lleguen ambos eventos de entrada. Por lo tanto, para calcular el *offset* y el *jitter* de la instancia p de la tarea τ_{if} tenemos que fijarnos en los efectos generados por los tiempos de respuesta de la instancia p de la tarea τ_{if-1} y de la instancia $p-1$ de la tarea τ_{il} . En la Figura 5.4, podemos ver dos ejemplos de representación de tiempos de respuesta de estas dos tareas.

Primero, para calcular el *offset*, tenemos que fijarnos en el tiempo de respuesta de mejor caso de estas dos instancias, siendo el mínimo entre ellas el *offset* de la tarea. En la Figura 5.4, tenemos las dos situaciones posibles que pueden ocurrir en el sistema. En la situación de la Figura 5.4a, el tiempo de respuesta de mejor caso de la instancia p de la tarea τ_{if-1} finaliza antes que el tiempo de respuesta de mejor caso de la instancia $p-1$ de la tarea τ_{il} , $R_{if-1}^{b(p)} + T_i < R_{il}^{b(p-1)}$. Por contra, en la situación de la Figura 5.4b, el tiempo de respuesta de mejor caso de la instancia $p-1$ de la tarea τ_{il} finaliza antes que el de la instancia p de la tarea τ_{if-1} , $R_{il}^{b(p-1)} < R_{if-1}^{b(p)} + T_i$. Por las características de la sección atómica, sabemos que la instancia p de la tarea τ_{if} no puede activarse antes de que ambas tareas hayan finalizado. Por lo tanto, el *offset* mínimo en la Figura 5.4b es:

$$\phi_{if}^{(p)} = R_{if-1}^{b(p)} \quad (5.7)$$

Por otra parte, en la Figura 5.4a, si aplicamos el mismo criterio, el *offset* mínimo es:

$$\phi_{if}^{(p)} = R_{il}^{b(p-1)} - T_i \quad (5.8)$$

Los *offsets* negativos no se permiten, pues ello implicaría empezar a ejecutar la tarea antes de que llegue el evento que la activa. Si en el caso de la Figura 5.4a obtuviésemos un *offset* positivo ello significaría que el tiempo de respuesta de mejor caso de la tarea τ_{il} sería superior al periodo, con lo que el procesador no podría soportar la carga de trabajo y el análisis sería innecesario, por lo que desestimamos este caso. En consecuencia el *offset* generado tanto por la tarea precedente como por la última tarea de la sección atómica

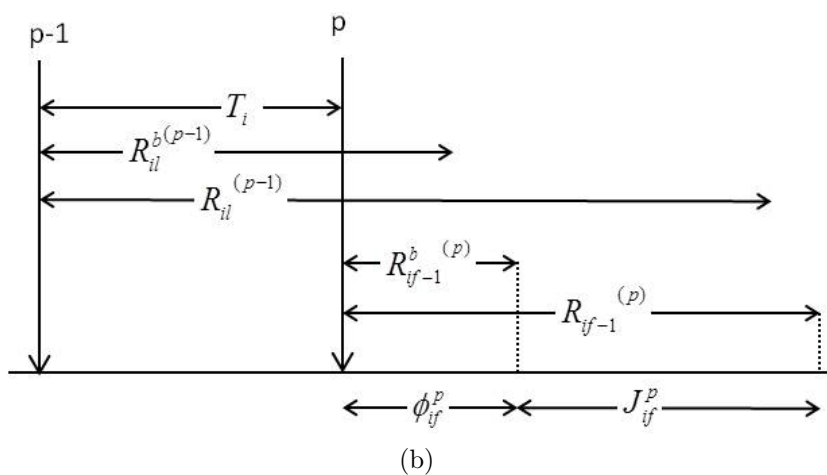
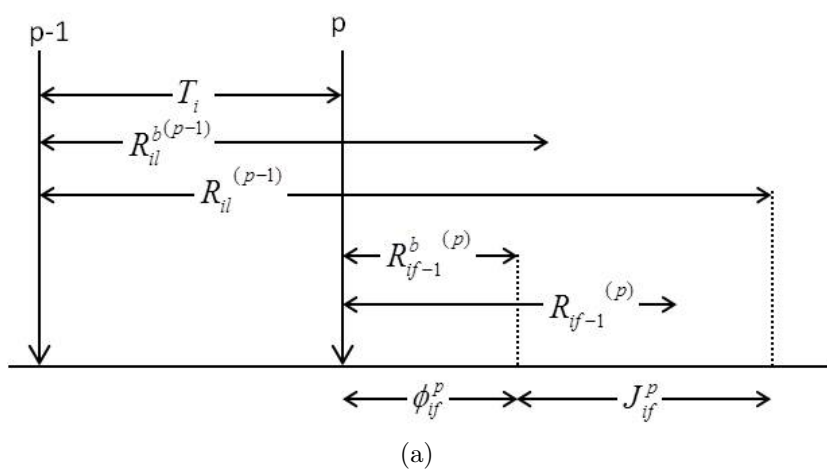


Figura 5.4: Representación de tiempos del *offset* y *jitter* de la tarea τ_{if}

se calcularía con la ecuación (5.7). Al inicio de la demostración hemos obviado el *offset* mínimo Φ_{if} de la tarea debido a que este concepto no interviene en el *offset* provocado por una tarea precedente, únicamente impone un valor mínimo al valor final del *offset* ϕ_{if} de la tarea. Por lo tanto, si le añadimos el concepto de *offset* mínimo, obtenemos la ecuación (5.5).

Por otro lado, para calcular el *jitter* de la tarea τ_{if} , tenemos que fijarnos en el tiempo de peor caso de las tareas τ_{if-1} y τ_{il} . En la Figura 5.4a, teniendo en cuenta el tiempo de ejecución de peor caso, la instancia $p - 1$ de la tarea τ_{il} finaliza más tarde que la instancia p de la tarea τ_{if-1} . Por lo tanto el *jitter* máximo en este caso es:

$$J_{if}^{(p)} = R_{il}^{(p-1)} - T_i - R_{if-1}^b{}^{(p)} \quad (5.9)$$

Por otra parte, en la Figura 5.4b, es la instancia p de la tarea τ_{if-1} la que finaliza más tarde. Por lo tanto, el *jitter* máximo que puede sufrir la tarea τ_{if} es:

$$J_{if}^{(p)} = R_{if-1}^{(p)} - R_{if-1}^b{}^{(p)} \quad (5.10)$$

En consecuencia, para calcular el *jitter* máximo de la tarea τ_{if} , tenemos que calcular el máximo de ambas situaciones. Esto es, el máximo entre las ecuaciones (5.9) y (5.10):

$$\begin{aligned} J_{if}^{(p)} &= \max \left(R_{il}^{(p-1)} - T_i - R_{if-1}^b{}^{(p)}, R_{if-1}^{(p)} - R_{if-1}^b{}^{(p)} \right) \\ &= \max \left(R_{il}^{(p-1)} - T_i, R_{if-1}^{(p)} \right) - R_{if-1}^b{}^{(p)} \end{aligned} \quad (5.11)$$

Con lo que la ecuación resultante es la misma que en la ecuación (5.6) propuesta en el teorema 5.1. \square

Como se ve en estas ecuaciones tanto el *jitter* como el *offset* de la primera tarea de la sección atómica dependen del tiempo de respuesta de la tarea precedente. Puede ocurrir que la primera tarea de la sección atómica sea también la primera tarea del flujo de tareas y, por lo tanto, no tener tarea precedente. En este caso, el tiempo de respuesta de mejor y peor caso de la tarea precedente (R_{if-1}^b y R_{if-1}) se pueden sustituir por 0.

Con estas ecuaciones, podemos calcular los efectos de *offset* y *jitter* derivados de las nuevas relaciones de precedencia introducidas por la sección atómica de tareas.

5.3.2. Interferencia de peor caso

Al igual que en las técnicas de capítulos anteriores, necesitamos calcular la interferencia de peor caso que un flujo de tareas Γ_i puede generar en la tarea bajo análisis τ_{ab} . Para ello, tenemos que identificar qué instancias de las tareas en el flujo de tareas Γ_i pueden ejecutarse antes que la instancia de la tarea τ_{ab} . Las técnicas basadas en *offsets* con relaciones de precedencia [PGH99] [DdCUGH14a] establecen que la interferencia de algunas de estas instancias pueden ser incompatibles debido a su relación de precedencia. En esta nueva técnica, seguiremos el mismo proceso para el cálculo de la interferencia que en la técnica presentada en la sección 4.3. Es decir, crearemos una tabla de conflictos con todas las instancias que pueden interferir en el tiempo de respuesta de la tarea bajo análisis y resolveremos dichos conflictos. Tomaremos como ejemplo el flujo de las tareas de la Figura 5.5. En la Figu-

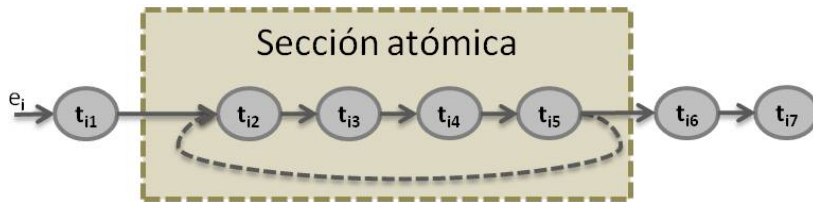


Figura 5.5: Ejemplo de flujo de tareas con sección atómica

ra 5.6a podemos ver un ejemplo de una tabla de interferencia generada por dicho flujo de tareas, en la que las columnas representan las tareas del flujo de tareas Γ_i que se ejecutan en el mismo procesador que la tarea bajo análisis τ_{ab} , y las filas representan las diferentes instancias del flujo de tareas. Cada celda define si esa instancia puede interferir (valor igual a C_n) o no puede (valor igual a cero). Las columnas con sombreado más oscuro representan la sección atómica de tareas. También podemos ver cajas con borde negro señalando las diferentes *H-sections*, descritas en la definición 2.2. Como se ha explicado anteriormente, por cada instancia del flujo de tareas, esto es, por cada fila de la tabla, solo una *H-section* puede interferir en el mismo periodo de ocupación.

Para tener en cuenta las relaciones de precedencia creadas por la inclusión de la sección atómica de tareas, es necesario añadir un nuevo concepto, llamado *atomic H-section (AH-section)*. Como hemos visto en la Figura 5.1, la sección atómica de tareas tiene una relación de precedencia cíclica. Por es-

	t_{i1}	t_{i2}	t_{i3}	t_{i4}	t_{i5}	t_{i6}	t_{i7}
p_n	C_1	C_2	0	C_4	C_5	C_6	0
p_{n+1}	C_1	C_2	0	C_4	C_5	0	C_7
p_{n+2}	C_1	C_2	0	C_4	C_5	0	C_7
p_{n+3}	0	C_2	C_3	C_4	C_5	0	C_7

(a)

	t_{i1}	t_{i2}	t_{i3}	t_{i4}	t_{i5}	t_{i6}	t_{i7}
p_n	C_1	C_2	0	C_4	C_5	C_6	0
p_{n+1}	C_1	C_2	0	C_4	C_5	0	C_7
p_{n+2}	C_1	C_2	0	C_4	C_5	0	C_7
p_{n+3}	0	C_2	C_3	C_4	C_5	0	C_7

(b)

Figura 5.6: Tabla de interferencia del flujo de tareas Γ_i para el cálculo del tiempo de respuesta de la tarea τ_{ab} con las *H-sections* marcadas (a) y las *AH-sections* marcadas (b).

te motivo, una *AH-section* puede contener instancias de tareas de diferentes instancias del flujo de tareas.

Definición 5.2. Dos instancias τ_{ij}^p y τ_{ik}^q pertenecen a la misma *AH-section*, para el análisis de la tarea τ_{ab} , si ambas tareas son parte de la sección atómica de tareas, se ejecutan en el mismo procesador que τ_{ab} con un plazo absoluto más cercano o igual que el de τ_{ab} , y no hay instancias intermedias (que sean parte de la sección atómica de tareas) entre ellas, en el mismo procesador y con un plazo absoluto más lejano que el de τ_{ab} . $AH_{ij}^p(\tau_{ab})$ identifica la *AH-section* a la que pertenece la instancia τ_{ij}^p y está compuesto por un conjunto de instancias de las tareas de la sección atómica de tareas del flujo de tareas Γ_i que verifican que:

$$AH_{ij}^p(\tau_{ab}) = \left\{ \tau_{il}^e \in \Gamma_i \mid (\exists \tau_{ix}^y [p * n + j \leq y * n + x \leq e * n + l \vee e * n + l \leq y * n + x \leq p * n + j] \mid \text{proc}(\tau_{ix}) = \text{proc}(\tau_{ab}) \wedge \text{dead}(\tau_{ix}^y) > D \wedge \text{dead}(\tau_{ij}^p) > D \wedge \tau_{ix} \in \text{atomic task-section}) \right\} \quad (5.12)$$

donde D es el plazo absoluto de la tarea bajo análisis τ_{ab} , $\text{dead}(\tau_{ix}^p)$ es el plazo absoluto de la instancia p de la tarea τ_{ix} y n es el número de tareas en el flujo de tareas Γ_i . Además, $p * n + x$ es la posición absoluta en el flujo de tareas de la instancia p de la tarea τ_{ix} (τ_{ix}^p).

En la Figura 5.6b se puede ver el mismo ejemplo de una tabla de interferencias que en la Figura 5.6a, pero en este caso, las cajas con bordes negros señalan las diferentes *AH-sections*. Considerando la naturaleza de las secciones atómicas de tareas, solo una única *AH-section* de toda la tabla puede interferir en el mismo periodo de ocupación. Por este motivo, si la tarea τ_{ik} (creadora del instante crítico) es parte de la sección atómica, solo la *AH-section*, en la que está la instancia $p'_{0,ikk}$ de la tarea τ_{ik} ($AH_{ik}^{p'_{0,ikk}}(\tau_{ab})$), puede interferir con la tarea bajo análisis. Por ejemplo, imaginemos que la instancia generadora del instante crítico ($p'_{0,ikk}$) es $\tau_{i4}^{p_n}$, es decir, la instancia p_n de la tarea τ_{i4} . Solo las instancias de esa misma tarea o de otras pertenecientes a la sección atómica que estén en la misma *AH-section* pueden causar interferencia en la tarea bajo análisis:

$$AH_{i4}^{p_n}(\tau_{ab}) = \{ \tau_{i4}^{p_n}, \tau_{i5}^{p_n}, \tau_{i2}^{p_{n+1}} \} \quad (5.13)$$

Por lo tanto, las demás *AH-sections* se pueden reducir. Teniendo en cuenta esta reducción, el nuevo algoritmo para crear la tabla de conflictos se muestra en el Algoritmo 5.1.

Algoritmo 5.1: construir_tabla_conflictos_con_seccion_atmica_EDF

Input: τ_{ab} , D , t , Γ_i , τ_{ik}

Result: Tabla

```

1 begin
2   Inicializar_Tabla;
3   for  $\tau_{ij} \in \Gamma_i | proc(\tau_{ij}) = proc(\tau_{ab})$  do
4     if  $D > d_{ijk}$  then
5       |  $p'_0 := p'_{0,ijk}$ ;
6     else
7       |  $p'_0 := p'_{f,ijk}$ ;
8     for  $p' \in p'_0 \cdot p'_{t,D,ijk}$  do
9       |  $Table[j, p'] := C_{ij}$  ;
10      if  $p' \geq 1$  and  $j \notin MP_i^{p'}(\tau_{ab})$  then
11        |  $Table[j, p'] := 0$ ;
12      if  $p' \geq p'_{0,ikk}$  and  $j > k$  and  $H_{ij}^{p'}(\tau_{ab}) \neq H_{ik}^{p'}(\tau_{ab})$  then
13        |  $Table[j, p'] := 0$ ;
14      if  $\tau_{ik}, \tau_{ij} \in atomic\ task\ -\ section$  and
15        |  $AH_{ij}^{p'}(\tau_{ab}) \neq AH_{ik}^{p'_{0,ikk}}(\tau_{ab})$  then
        |  $Table[j, p'] := 0$ ;

```

5.3. Análisis de planificabilidad

Como se ha mencionado antes, solo una *H-section* por cada fila de la tabla y una *AH-section* por toda la tabla pueden interferir simultáneamente en el tiempo de respuesta de peor caso de la tarea bajo análisis. Para calcular la interferencia de peor caso, se repite el análisis de la tabla tantas veces como *AH-sections* hay en la tabla. Cada vez haremos el análisis teniendo en cuenta una única *AH-section* y descartando las demás. Una vez que hayamos hecho esto, el análisis se realiza como en el Algoritmo 4.2. La interferencia total será el máximo de todas las calculadas con las diferentes *AH-sections*. El proceso completo se puede ver en el Algoritmo 5.2.

Algoritmo 5.2: resolver_confl._con_sec._atom._EDF

Input: $\tau_{ab}, D, t, \Gamma_i, \tau_{ik}$
Result: Interferencia: time

```

1 begin
2   construir_tabla_confl._con_seccion_atomica_EDF( $\tau_{ab}, D, t, \Gamma_i,$ 
    $\tau_{ik}$ ) ;
3   Interferencia := 0;
4   foreach  $x$ : AH-section en la tabla do
5     Total := 0;
6     for  $p' \in p'_{min} \cdot p'_{max}$  do
7       max_section,sum:= 0;
8       for  $\tau_{ij} \in \Gamma_i | proc(\tau_{ij}) = proc(\tau_{ab})$  do
9         if  $Table[j, p'] = 0$  or ( $\tau_{ij} \in atomic\ task\ section$  and
           $\tau_{ij}^{p'} \notin x$ ) then
10          sum := 0 ;
11          else
12            sum := sum + Table[j, p'] ;
13            if sum > max_section then
14              max_section := sum;
15          Total := Total + max_section ;
16        if Total > Interference then
17          Interferencia := Total;

```

Por lo tanto, la interferencia del flujo de tareas Γ_i sobre la tarea bajo análisis τ_{ab} con un plazo D , en un periodo de ocupación de longitud t iniciado en el instante crítico creado con la tarea τ_{ik} es:

$$W_{ik}(\tau_{ab}, t, D) = resolver_confl._con_sec._atom._EDF(\tau_{ab}, D, t, \Gamma_i, \tau_{ik}) \quad (5.14)$$

Y el límite superior de la interferencia máxima del flujo de tareas Γ_i :

$$W_i^*(\tau_{ab}, t, D) = \max(W_{ik}(t, D)), \forall k \in \Gamma_i | \text{proc}(\tau_{ik}) = \text{proc}(\tau_{ab}) \quad (5.15)$$

5.3.3. Aplicación al flujo de tareas bajo análisis

En esta sección analizaremos también las relaciones de precedencia, pero en este caso para calcular la interferencia del flujo de tareas Γ_a , al que pertenece la tarea bajo análisis τ_{ab} . Para calcular la interferencia máxima tenemos que estudiar los diferentes instantes críticos creados por las tareas del flujo de tareas Γ_a . La interferencia del flujo de tareas Γ_a a la tarea τ_{ab} en un periodo de ocupación creado a partir de la tarea τ_{ac} , se calcula construyendo una tabla de conflictos con un algoritmo similar al Algoritmo 5.1. Pero en el caso del flujo de tareas Γ_a , además se pueden aplicar las tres reducciones presentadas en las ecuaciones 4.30 a 4.32 en la sección 4.3.2. Además se puede añadir una nueva reducción debido al uso de la sección atómica de tareas. Si la instancia bajo análisis τ_{ab}^p está dentro de la sección atómica de tareas, otras instancias que están dentro de la sección atómica de tareas, pero no en la misma *AH-section*, no pueden interferir con la tarea bajo análisis. Por lo que podemos aplicar la siguiente regla:

4ª regla de reducción. Siendo la instancia bajo análisis τ_{ab}^p parte de la sección atómica de tareas, todas las instancias en la sección atómica de tareas que están en conflicto con dichas instancias pueden ser eliminadas. Por tanto se pueden eliminar las celdas (j, p') que cumplan:

$$\tau_{ab}, \tau_{aj} \in \text{atomic task-section} \wedge AH_{aj}^{p'}(\tau_{ab}) \neq AH_{ab}^p(\tau_{ab}) \quad (5.16)$$

Aplicando estas restricciones, la tabla de conflictos se construye mediante el Algoritmo 5.3, similar al Algoritmo 5.1, pero introduciendo las reglas de reducción.

La resolución de conflictos se realiza siguiendo el mismo proceso que en el Algoritmo 5.2, pero construyendo la tabla mediante el Algoritmo 5.3. Con ello obtenemos el Algoritmo 5.4.

Y por lo tanto, la contribución total del flujo de tareas Γ_a al periodo de ocupación de longitud t creado a partir de la tarea τ_{ac} para el cálculo

Algoritmo 5.3: construir_tabla_confl._con_sec._atom._en_ Γ_a _EDF

Input: $\tau_{ab}^p, D, t, \Gamma_i, \tau_{ac}$
Result: Tabla

```

1 begin
2   Inicializar_Tabla;
3   for  $\tau_{aj} \in \Gamma_i | \text{proc}(\tau_{aj}) = \text{proc}(\tau_{ab})$  do
4     if  $D > d_{ajc}$  then
5       |  $p'_0 := p'_{0,ajc}$ ;
6     else
7       |  $p'_0 := p'_{f,ajc}$ ;
8     for  $p' \in p'_0 \cdot p'_{t,D,ajc}$  do
9       |  $\text{Table}[j, p'] := C_{aj}$  ;
10      if  $p' \geq 1$  and  $j \notin MP_i^{p'}(\tau_{ab})$  then
11        |  $\text{Table}[j, p'] := 0$ ;
12      if  $p' \geq p'_{0,acc}$  and  $j > c$  and  $H_{aj}^{p'}(\tau_{ab}) \neq H_{ac}^{p'}(\tau_{ab})$  then
13        |  $\text{Table}[j, p'] := 0$ ;
14      if  $p' \leq p'_{ab}$  and  $j < b$  and  $H_{aj}^{p'}(\tau_{ab}) \neq H_{ab}^{p'}(\tau_{ab})$  then
15        |  $\text{Table}[j, p'] := 0$ ;
16      if  $(p' \geq p'_{ab}$  and  $j > b)$  or  $(p' > p'_{ab}$  and  $j = c)$  then
17        |  $\text{Table}[j, p'] := 0$ ;
18      if  $\tau_{ac}, \tau_{aj} \in \text{atomic task-section}$  and
19         $AH_{aj}^{p'}(\tau_{ab}) \neq AH_{ac}^{p_0,acc}(\tau_{ab})$  then
20        |  $\text{Table}[j, p'] := 0$ ;
21      if  $\tau_{ab}, \tau_{aj} \in \text{atomic task-section}$  and  $AH_{aj}^{p'}(\tau_{ab}) \neq AH_{ab}^p(\tau_{ab})$ 
22      then
23        |  $\text{Table}[j, p'] := 0$ ;

```

Algoritmo 5.4: resolver_conflictos_con_seccion_atmica_en_Gamma_a_EDF

Input: $\tau_{ab}, D, t, \Gamma_i, \tau_{ik}$

Result: Interferencia: time

```

1 begin
2   construir_tabla_conflictos_con_seccion_atmica_en_Gamma_a_EDF( $\tau_{ab}^p, D, t,$ 
    $\Gamma_i, \tau_{ik}$ );
3   Interferencia := 0;
4   foreach  $x$ : AH-section en la tabla do
5     Total := 0;
6     for  $p' \in p'_{min} \cdot p'_{max}$  do
7       max_section, sum := 0;
8       for  $\tau_{ij} \in \Gamma_i | proc(\tau_{ij}) = proc(\tau_{ab})$  do
9         if  $Table[j, p'] = 0$  or ( $\tau_{ij} \in atomic\ task\ section$  and
10           $\tau_{ij}^{p'} \notin x$ ) then
11           sum := 0;
12         else
13           sum := sum + Table[j, p'];
14           if sum > max_section then
15             max_section := sum;
16           Total := Total + max_section;
17       if Total > Interferencia then
18         Interferencia := Total;

```

5.4. Evaluación del análisis con secciones atómicas

del tiempo de respuesta de la instancia $\tau_{ab}^{p'}$ se obtiene mediante la siguiente ecuación:

$$W_{ac}(\tau_{ab}^{p'}, t, D) = \text{resolver_confl._con_seccion_atomica_} \\ \text{_en_}\Gamma_a\text{_EDF}(\tau_{ab}^{p'}, D, t, \Gamma_a, \tau_{ac}) \quad (5.17)$$

El tiempo de finalización $w_{abc}(p'_{ab})$ de la instancia p'_{ab} de la tarea τ_{ab} se puede calcular mediante la ecuación:

$$w_{abc}(p'_{ab}) = B_{ab} + W_{ac}(\tau_{ab}^{p'}, w_{abc}(p'_{ab}), D) + \sum_{\forall i \neq a} W_i^*(\tau_{ab}, w_{abc}(p'_{ab}), D) \quad (5.18)$$

Obteniendo W_{ac} y W_i mediante las ecuaciones 5.17 y 5.15 respectivamente.

El proceso para calcular el tiempo de respuesta es el mismo que el utilizado en la sección 4.2 para la técnica de análisis basada en *offsets*. Y por lo tanto, se pueden utilizar las ecuaciones desde (4.10) a (4.20).

5.4. Evaluación del análisis con secciones atómicas

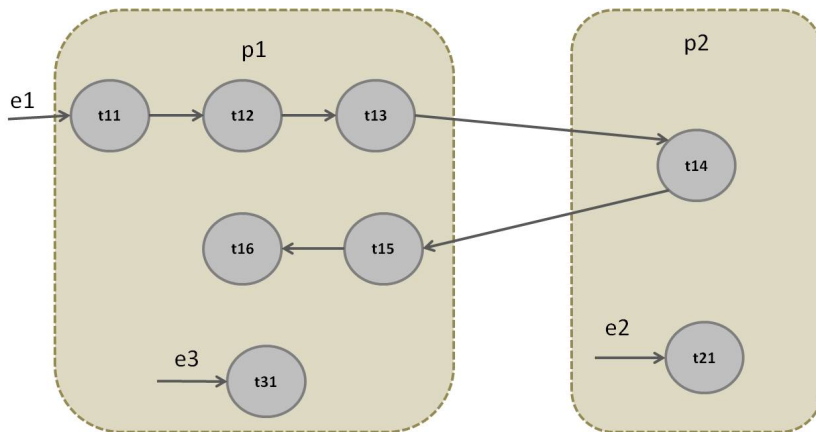


Figura 5.7: Ejemplo de aplicación de un flujo de tareas con sección atómica: Composición de flujos de tarea y distribución en procesadores

En esta sección se presenta un ejemplo de aplicación con secciones atómicas a la que se le aplica tanto la técnica con una sección atómica, presentada

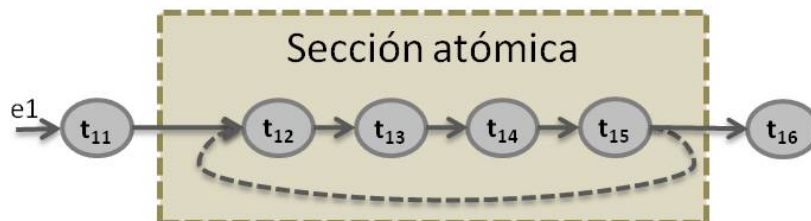


Figura 5.8: Flujo de tareas con sección atómica, e_1

en la sección anterior, como la técnica sin secciones atómicas. Los resultados de estos dos análisis se compararan para mostrar la diferencia en el pesimismo de ambas técnicas. Este sistema cuenta con tres flujos de tareas periódicos distribuidos en dos procesadores. La composición de tareas y su distribución en procesadores se pueden ver en la Figura 5.7. El flujo e_1 representa una sección de código en la que se realiza una llamada remota entre los procesadores p_1 y p_2 . Como se ha descrito en la sección 5.1 esta es una de las estructuras de programación en la que el modelo de las secciones atómicas es especialmente útil. En la Figura 5.8 se puede observar el modelo del flujo de tareas e_1 con las tareas pertenecientes a la sección atómica (τ_2 - τ_5) recogidas en un recuadro.

Las características temporales de los flujos de tareas y de las tareas se pueden ver en las Tablas 5.1 y 5.2.

Tabla 5.1: Características temporales de los flujos de tareas.

Flujo de tareas	Tipo de flujo	T/MIT
e_1	periódico	145
e_2	periódico	2000
e_3	periódico	2000

A este sistema le hemos aplicado dos técnicas: en la primera técnica se tienen en cuenta las relaciones de precedencia generadas por las secciones atómicas, mientras que en la segunda técnica no se tienen en cuenta estas relaciones y se realiza el análisis clásico con *offsets*. En la Tabla 5.3 se pueden ver los tiempos de respuesta obtenidos para las tareas con ambos métodos. Como se puede observar los tiempos de respuesta son inferiores para la técnica que tiene en cuenta las secciones atómicas. Con lo que se puede concluir que

5.5. Conclusiones

Tabla 5.2: Características temporales de las tareas del sistema.

Tarea	C	Cb	d	proc
t11	5	2	500	p1
t12	6	2	700	p1
t13	5	1	500	p1
t14	4	1	700	p2
t15	7	3	800	p1
t16	2	2	200	p1
t21	50	20	80	p2
t31	30	20	80	p1

en este sistema, y gracias a la toma en consideración de las nuevas relaciones de precedencia, los resultados son menos pesimistas con esta técnica.

Tabla 5.3: Comparativa de tiempos de respuesta de las tareas del sistema para las técnicas con y sin secciones atómicas.

Tarea	R (con análisis de secciones atómicas)	R (sin análisis de secciones atómicas)
t11	30	30
t12	60	71
t13	90	101
t14	140	151
t15	170	186
t16	200	216
t21	50	50
t31	30	30

5.5. Conclusiones

En este capítulo hemos extendido el modelo lineal de flujo de tareas para soportar secciones atómicas de tareas. Esto permite a los diseñadores disponer de un mayor número de elementos de modelado con los que modelar sus sistemas de tiempo real de una forma más precisa. Además, se ha adaptado

5. Análisis de planificabilidad de flujos de tareas con secciones atómicas

la técnica de análisis basada en *offsets* con relaciones de precedencia para sistemas LC-EDF para tener en cuenta las relaciones de precedencia que surgen con el uso de secciones atómicas de tareas. Esto reduce el pesimismo del análisis, permitiendo incrementar la ratio de utilización del sistema.

Aunque solo se ha aplicado el concepto de sección atómica de tareas a sistemas planificados mediante LC-EDF, parece sencilla su adaptación a otros planificadores, como GC-EDF y FP. Por tanto este punto podría ser una futura línea de trabajo. También dejamos como trabajo futuro la aplicación del nuevo modelo de flujo de tareas a la herramienta de análisis MAST [MAS] y la implementación de la nueva técnica de análisis. Para ello, será necesario un rediseño del modelo de tareas utilizado en la herramienta.

Por otra parte queda como trabajo futuro ampliar el análisis para que admita múltiples secciones atómicas de tareas en un mismo flujo, o incluso secciones atómicas anidadas. Esto aumentará las posibilidades de los diseñadores de sistemas de tiempo real para el modelado de dichos sistemas.

Capítulo 6

Conclusiones y trabajo futuro

RESUMEN: Para concluir con esta tesis vamos a revisar los logros conseguidos en este trabajo. Para ello, primero, en la sección 6.1, vamos a revisar los objetivos establecidos en el capítulo 1. Posteriormente, en la sección 6.2, resumiremos las contribuciones aportadas en la presente tesis. Por último, en la sección 6.3, indicaremos las posibles líneas de trabajo que surgen a partir de este trabajo.

6.1. Revisión de objetivos

El objetivo principal de esta tesis ha consistido en validar y mejorar el análisis de planificación para sistemas distribuidos de tiempo real con planificación EDF. El modelo de sistema que se contempla en esta tesis tiene las siguientes características:

- Sistema compuesto por un conjunto de procesadores conectados entre sí por sistemas de comunicación de tiempo real. Además el sistema puede contar con sensores y actuadores que permiten la interacción con el entorno físico.
- Sistema compuesto por un conjunto de tareas y mensajes estáticamente alojadas en los procesadores o redes de comunicación del sistema.

Dichas tareas y mensajes se agrupan en cadenas o flujos de tareas. La activación del flujo de tareas, y por lo tanto, de la primera tarea, viene dada por la llegada de un evento externo. Cada llegada del evento externo generará una instancia del flujo de tareas. Las siguientes tareas/mensajes se activan por eventos internos generados a la finalización de la tarea/mensaje precedente.

- La ejecución de las tareas se planifica mediante un modelo de planificador dinámico expulsor basado en la planificación EDF.

Para conseguir validar el análisis de planificación, el primer objetivo que se ha planteado en esta tesis consiste en validar una de las técnicas de análisis para sistemas distribuidos de tiempo real con planificación EDF que se utiliza como base para una gran mayoría de las técnicas existentes. Con ello, se han querido afianzar las bases sobre las que se asienta el objetivo principal de la tesis. Para ello, en el capítulo 3, analizamos la validez de la técnica holística desarrollada por Spuri [Spu96b].

Como hemos mencionado anteriormente, el interés principal de esta tesis ha consistido en validar y mejorar las técnicas de análisis para sistemas distribuidos de tiempo real con planificación EDF. Con ello, se pretende poder acercarse a las técnicas desarrolladas para sistemas con planificaciones FP. En el capítulo 4, se han desarrollado dos nuevos algoritmos que permiten reducir el pesimismo en el cálculo de tiempos de respuesta de sistemas distribuidos planificados mediante EDF.

Para finalizar, se ha planteado un último objetivo que consiste en ampliar el modelo de tareas para permitir mayor flexibilidad a la hora de modelar sistemas de tiempo real. Esta ampliación ha consistido en introducir un concepto de *thread* atómico, en el cual la activación de varias instancias (de una o varias tareas diferentes) no puede ser simultánea. Además, parte de este objetivo ha consistido en adaptar el análisis del capítulo 4 a este nuevo modelo, reduciendo al máximo el pesimismo. En el capítulo 5, se recoge dicha adaptación.

6.2. Contribuciones de este trabajo

La primera contribución de este trabajo ha supuesto probar la validez de la técnica holística de Spuri para el análisis de planificabilidad de sistemas dis-

tribuidos de tiempo real con planificación EDF, demostrando que las fórmulas iterativas usadas son no-decrescentes, como se afirma en [Spu96a, Spu96b]. Además, se ha aplicado dicha técnica al subsistema de control de apertura/-cierre de puertas de un ascensor. Los resultados obtenidos verifican la planificabilidad del sistema y el cumplimiento de las restricciones temporales. Con ello, hemos demostrado la utilidad de este tipo de análisis en sistemas industriales.

Una conclusión obtenida a partir de esta primera contribución es que, debido a que los niveles de expulsión y los techos de prioridad de los recursos compartidos dependen de los términos de *jitter*, cuyos valores definitivos únicamente se conocen tras finalizar el análisis. Dichos niveles y techos deberán ir variando durante el análisis y los valores definitivos se obtendrán como resultado del proceso de análisis. Esto no se mencionaba en el trabajo original de Spuri, por lo que supone una pequeña modificación del mismo.

La segunda contribución de esta tesis consiste en el desarrollo de dos nuevos algoritmos para el análisis de planificabilidad para sistemas distribuidos de tiempo real con el modelo de flujos de tareas lineales planificados con la política LC-EDF (EDF de relojes locales). El primero de estos algoritmos introduce el análisis basado en *offsets* para LC-EDF. Este análisis es capaz de reducir el pesimismo del análisis anterior para este tipo de planificación, que está basado en la técnica holística, donde se asume que las tareas se activan de forma independiente a las demás. Añadir *offsets* al modelo de tareas ayuda a modelar las relaciones temporales entre las activaciones de las diferentes tareas de un flujo de tareas y, usándolos en el análisis, el pesimismo se reduce significativamente. En los experimentos, usando casos extensos, hemos obtenido un aumento en la utilización máxima planificable de la CPU entre un 9% y un 11%.

El segundo algoritmo consiste en explotar los beneficios de analizar las relaciones de precedencia entre las tareas para minimizar el pesimismo del algoritmo de análisis aun más. Las simulaciones muestran que los beneficios son destacables. Con esta nueva técnica, la utilización máxima planificable aumenta entre un 16% y un 24% sobre la técnica holística previa.

La última contribución consiste en la extensión del modelo lineal de flujo de tareas para soportar secciones atómicas de tareas que permitan a los diseñadores modelar sus sistemas de tiempo real de una forma más precisa. Además, se ha adaptado la técnica de análisis basada en *offsets* con relaciones de precedencia para sistemas LC-EDF para tener en cuenta las relaciones de

precedencia que surgen con el uso de secciones atómicas de tareas. Esto reduce el pesimismo del análisis, permitiendo incrementar la ratio de utilización del sistema.

6.3. Trabajo futuro

El trabajo desarrollado en esta tesis genera un nuevo conjunto de líneas de trabajo a abordar.

Técnica holística de Spuri

En este trabajo se ha demostrado la validez de esta técnica. Durante el proceso se han implementado herramientas prototipo con las técnicas descritas que han permitido analizar el caso de estudio del capítulo 3. Aunque el método holístico está implementado en la herramienta MAST [MAS] está restringido a sistemas sin recursos compartidos, debido a que antes de este trabajo la aplicabilidad de la técnica holística a sistemas con recursos compartidos planificados bajo EDF estaba cuestionada. Al haberse demostrado su correcto funcionamiento en este trabajo, en el futuro MAST debería permitir el uso de este algoritmo para sistemas con recursos compartidos. El algoritmo holístico extendido, presentado en esta tesis, supone un cambio en la arquitectura de la herramienta de análisis MAST, que trata los niveles de expulsión y los techos de los recursos como *inputs* del análisis. Mientras que, en este nuevo algoritmo, los niveles y los techos deben cambiar en cada iteración del sistema, obteniéndose los valores definitivos a la finalización del proceso de análisis.

Nuevos algoritmos de análisis con *offsets* y precedencias

Como se ha comentado antes, los dos nuevos algoritmos presentados en el capítulo 4 son técnicas aproximadas y, por lo tanto, son susceptibles de ser mejoradas reduciendo su pesimismo. Existen varios trabajos, como [MTN08] y [Red04], que introducen mejoras a las técnicas basadas en *offsets* y precedencias para FP. Estas técnicas, que consiguen reducir el pesimismo del análisis, no han sido tenidas en cuenta a la hora de desarrollar las técnicas para sistemas planificados con EDF. Por lo tanto, una posible línea de trabajo podría ser analizar la posibilidad de aplicar estas mejoras a las técnicas de análisis para EDF.

6.3. Trabajo futuro

Además sería interesante el desarrollar un test exacto que permitiese cuantificar en un tiempo razonable el pesimismo absoluto de todas las técnicas aproximadas que se han presentado.

Por otra parte, con el desarrollo de estos dos nuevos algoritmos se ha completado gran parte del mapa de técnicas de análisis mostrado en la tabla 4.2. Con esto, se nos abre la posibilidad de realizar una comparación extensa entre las técnicas para LC-EDF y GC-EDF, así como las técnicas para FP.

Nuevo modelo de tareas

El nuevo modelo de tareas con las secciones atómicas de tareas solo se ha aplicado a sistemas planificados mediante LC-EDF extendiendo los algoritmos de análisis desarrollados en el capítulo 4. A pesar de ello, parece sencilla su aplicación a otros planificadores, como GC-EDF o FP. Por lo que este punto podría ser otra futura línea de trabajo.

Por último, también dejamos como trabajo futuro la aplicación del nuevo modelo de flujo de tareas a la herramienta de análisis MAST [MAS] y la implementación de la nueva técnica de análisis. Para ello, será necesario un rediseño del modelo de tareas utilizado en la herramienta.

Bibliografía

- [ABB96] Neil C. Audsley, I.J. Bate, and Alan Burns. Putting fixed priority scheduling theory into engineering practice for safety critical applications. In *Proceedings of Real-Time Technology and Applications Symposium*, pages 2–10. IEEE, 1996.
- [ABCD91] Gopal Agrawal, Wei Zhao Baio Chen, and Sadegh Davari. Architecture impact of FDDI network on scheduling hard real-time traffic. In *Workshop on Architectural Aspects of Real-Time Systems*. Citeseer, 1991.
- [ABR⁺93] Neil Audsley, Alan Burns, Mike Richardson, Ken Tindell, and Andy J. Wellings. Applying new scheduling theory to static priority pre-emptive scheduling. *Software Engineering Journal*, 8(5):284–292, 1993.
- [ARGH01] Mario Aldea Rivas and Michael González Harbour. *MaR-TE OS: An Ada kernel for real-time embedded applications*, pages 305–316. Springer, 2001.
- [Aud91] N. C. Audsley. Optimal priority assignment and feasibility of static priority tasks with arbitrary start times. *Real-Time Systems*, 1991.
- [AUM⁺11] Ekain Azketa, Juan P. Uribe, Marga Marcos, Luis Almeida, and J. Javier Gutiérrez. Permutational genetic algorithm for the optimized assignment of priorities to tasks and messages in distributed real-time systems. In *Proceedings of the 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 958–965. IEEE, 2011.

- [B⁺06] Giorgio Buttazzo et al. Real-time scheduling and resource management. 2006.
- [Bak90] Theodore P. Baker. A stack-based resource allocation policy for realtime processes. In *Proceedings of the 11th Real-Time Systems Symposium*, pages 191–200. IEEE, 1990.
- [Bak91] Theodore P. Baker. Stack-based scheduling of realtime processes. *Real-Time Systems*, 3(1):67–99, 1991.
- [Bar06] Sanjoy K. Baruah. Resource sharing in EDF-scheduled systems: A closer look. In *Proceedings of the 27th IEEE International Real-Time Systems Symposium RTSS'06*, pages 379–387. IEEE, 2006.
- [BB97] Iain Bate and Alan Burns. Schedulability analysis of fixed priority real-time systems with offsets. In *Proceedings of the Ninth Euromicro Workshop on Real-Time Systems*, pages 153–160. IEEE, 1997.
- [BB98] Iain Bate and Alan Burns. Investigation of the pessimism in distributed systems timing analysis. In *Proceedings of the 10th Euromicro Workshop on Real-Time Systems*, pages 107–114. IEEE, 1998.
- [BCGG09] Reinder J. Bril, Liliana Cucu-Grosjean, and Joël Goossens. Exact best-case response time analysis of real-time tasks under fixed-priority pre-emptive scheduling for arbitrary deadlines. In *Proceedings of the 21st Euromicro Conference on Real-Time Systems - ECRTS'09, WIP session*. IEEE, 2009.
- [BCP03] Guillem Bernat, Antoine Colin, and Stefan Petters. pWCET: A tool for probabilistic worst-case execution time analysis of real-time systems. Technical Report 0963-620X, University of York, 2003.
- [BHR90] Sanjoy K. Baruah, Rodney R. Howell, and Louis Rosier. Algorithms and complexity concerning the preemptive scheduling of periodic, real-time tasks on one processor. *Real-Time Systems*, 2:301–324, 1990.

- [BLM13] Reinder J. Bril, Johan J. Lukkien, and Rudolf H. Mak. Best-case response times and jitter analysis of real-time tasks with arbitrary deadlines. In *Proceedings of the 21st International conference on Real-Time Networks and Systems*, pages 193–202. ACM, 2013.
- [BMR90] Sanjoy K. Baruah, Aloysius K. Mok, and Louis E. Rosier. Preemptively scheduling hard-real-time sporadic tasks on one processor. In *Proceedings of the 11th Real-Time Systems Symposium*, pages 182–190, 1990.
- [BS89] Theodore P. Baker and Alan Shaw. The cyclic executive model and Ada. *Real-Time Systems*, 1(1):7–25, 1989.
- [Bur12] Alan Burns. A deadline-floor inheritance protocol for EDF scheduled embedded real-time systems with resource sharing. Technical report, Department of Computer Science, University of York, UK, 2012.
- [But05] Giorgio C. Buttazzo. Rate monotonic vs. EDF: judgment day. *Real-Time Systems*, 29(1):5–26, 2005.
- [BW89] Alan Burns and Andrew J. Wellings. *Real-time systems and programming languages*. Addison-Wesley, 1989.
- [BW09] Alan Burns and Andy Wellings. *Real-Time Systems and Programming Languages: Ada, Real-Time Java and C/Real-Time POSIX*. Addison-Wesley Educational Publishers Inc, USA, 4th edition, 2009.
- [DBBL07] Robert I. Davis, Alan Burns, Reinder J. Bril, and Johan J. Lukkien. Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised. *Real-Time Systems*, 35(3):239–272, 2007.
- [DdCUGHPa] Unai Díaz-de Cerio, Juan P. Uribe, Michael González Harbour, and J. C. Palencia. On the convergence of the holistic analysis for EDF distributed systems. *Remitido a Journal of Systems Architecture*.
- [DdCUGHPb] Unai Díaz-de Cerio, Juan P. Uribe, Michael González Harbour, and J. C. Palencia. Schedulability analysis for EDF distributed real-time systems with atomic task sections. *Remitido a Journal of Systems Architecture*.

-
- [DdCUGHP14a] Unai Díaz-de Cerio, Juan P. Uribe, Michael González Harbour, and J. C. Palencia. Adding precedence relations to the response-time analysis of EDF distributed real-time systems. In *Proceedings of the 22Nd International Conference on Real-Time Networks and Systems, RTNS '14*, pages 129:129–129:138, New York, NY, USA, 2014. ACM.
- [DdCUGHP14b] Unai Díaz-de Cerio, Juan P. Uribe, Michael González Harbour, and J. Carlos Palencia. Summary: On the convergence of the holistic analysis for edf distributed systems. In *XVII Jornadas de Tiempo Real*, 2014.
- [DdCUUGHPG14] Unai Díaz de Cerio Urain, Juan P Uribe, Michael González Harbour, and José Carlos Palencia Gutiérrez. On the convergence of the holistic analysis for edf distributed systems. Technical report, Universidad de Cantabria, 2014.
- [DM89] Michael L. Dertouzos and Aloysius K. Mok. Multiprocessor online scheduling of hard-real-time tasks. *IEEE Transactions on Software Engineering*, 15(12):1497–1506, 1989.
- [DMSW08] C. Diederichs, U. Margull, F. Slomka, and G. Wirrer. An application-based edf scheduler for osek/vdx. In *Design, Automation and Test in Europe, 2008. DATE '08*, pages 1045–1050, March 2008.
- [DMTW99] Chen Deji, A. K. Mok, and Kuo Tei-Wei. Utilization bound re-visited. In *Proceedings of the Sixth International Conference on Real-Time Computing Systems and Applications RTCSA '99*, pages 295–302. IEEE, 1999.
- [DN00] M. Di Natale. Scheduling the can bus with earliest deadline techniques. In *Real-Time Systems Symposium, 2000. Proceedings. The 21st IEEE*, pages 259–268, 2000.
- [DNM01] Marco Di Natale and Antonio Meschi. Scheduling messages with earliest deadline techniques. *Real-Time Systems*, 20(3):255–285, 2001.
- [DP05] Jean-François Deverge and Isabelle Puaut. Safe measurement-based WCET estimation. In *Proceedings*

- of the 5th International Workshop on Worst-Case Execution Time (WCET) Analysis*, volume 1, pages 13–16, 2005.
- [Eri] ERIKA Enterprise. Evidence home page, <http://www.evidence.eu.com/>.
- [FRB05] Santiago Fuster, Francisco Rodríguez, and Alberto Bonastre. Software-based EDF message scheduling on CAN networks. In *Proceedings of the Second International Conference on Embedded Software and Systems*, page 6 pp. IEEE, 2005.
- [GAGB01] Paolo Gai, Luca Abeni, Massimiliano Giorgi, and Giorgio Buttazzo. A new kernel approach for modular real-time systems development. In *Proceedings of the 13th Euro-micro Conference on Real-Time Systems*, pages 199–206. IEEE, 2001.
- [GARBGH14] Marina Gutiérrez, Mario Aldea Rivas, Alan Burns, and Michael González Harbour. A deadline-floor inheritance protocol for EDF scheduled embedded real-time systems with resource sharing. *IEEE Transactions on Computers*, 99(PrePrints):1, 2014.
- [GB10] David Griffin and Alan Burns. Realism in statistical analysis of worst case execution times. In *Proceedings of the 8th International Workshop on Worst-Case Execution Time (WCET) Analysis*, volume 15, pages 44–53, 2010.
- [GGH95] J. Javier Gutiérrez and Michael González Harbour. Optimized priority assignment for tasks and messages in distributed hard real-time systems. In *Proceedings of the Third Workshop on Parallel and Distributed Real-Time Systems*, pages 124–132. IEEE, 1995.
- [GHGD⁺13] Michael González Harbour, J. Javier Gutiérrez, José M. Drake, Patricia López Martínez, and J. C. Palencia. Modeling distributed real-time systems with mast 2. *Journal of Systems Architecture - Embedded Systems Design*, 59(6):331–340, 2013.

-
- [GHKL91] M González Harbour, Mark H Klein, and John P Lehoczky. Fixed priority scheduling periodic tasks with varying execution priority. In *Proceedings of 12th IEEE Real-Time Systems Symposium*, pages 116–128, 1991.
- [GPGH13] J. Javier Gutiérrez, J. C. Palencia, and Michael González Harbour. Holistic schedulability analysis for multi-packet messages in AFDX networks. *Real-Time Systems*, pages 1–40, 2013.
- [GRS96] Laurent George, Nicolas Rivierre, and Marco Spuri. Preemptive and non-preemptive real-time uniprocessor scheduling. 1996.
- [GSYY10] Nan Guan, Martin Stigge, Wang Yi, and Ge Yu. Fixed-priority multiprocessor scheduling with liu and layland’s utilization bound. In *Proceedings of the 19th Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 165–174. IEEE, 2010.
- [Gut95] J. Javier Gutiérrez. *Planificación, análisis y optimización de sistemas distribuidos de tiempo real estricto*. PhD thesis, Universidad de Cantabria. Supervisor: Michael González Harbour, 1995.
- [HBJK06] Hoai Hoang, Giorgio Buttazzo, Magnus Jonsson, and Stefan Karlsson. Computing the minimum edf feasible deadline in periodic systems. In *Proceedings of 12th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, 2006.*, pages 125–134. IEEE, 2006.
- [HHJ⁺05] Rafik Henia, Arne Hamann, Marek Jersak, Razvan Racu, Kai Richter, and Rolf Ernst. System level performance analysis-the SymTA/S approach. *IEE Proceedings of Computers and Digital Techniques*, 152(2):148–166, 2005.
- [HJ87] Paul K. Harter Jr. Response times in level-structured systems. *ACM Transactions on Computer Systems (TOCS)*, 5(3):232–248, 1987.
- [Jav] RTSJ (Real-Time Specification for Java). Home page, <http://www.rtsj.org/>.

- [Jef92] Kevin Jeffay. Scheduling sporadic tasks with shared resources in hard-real-time systems. In *Proceedings of Real-Time Systems Symposium, 1992*, pages 89–99. IEEE, 1992.
- [JLT85] E. Douglas Jensen, C. Douglass Locke, and Hideyuki Tokuda. Time-driven scheduling model for real-time operating systems. In *Proceedings of Real-Time Systems Symposium*, pages 112–122, 1985.
- [JP86] Mathai Joseph and P. Pandya. Finding response times in a real-time system. *The Computer Journal*, 29(5):390–395, 1986.
- [KDK⁺89] Hermann Kopetz, Andreas Damm, Christian Koza, Marco Mulazzani, Wolfgang Schwabl, Christoph Senft, and Ralph Zainlinger. Distributed fault-tolerant real-time systems: The Mars approach. *Micro, IEEE*, 9(1):25–40, 1989.
- [Kop11] Hermann Kopetz. *Real-time systems: design principles for distributed embedded applications*. Springer, 2011.
- [KRP⁺93] Mark H. Klein, Thomas Ralya, Bill Pollak, Ray Obenza, and Michael González Harbour. *A practitioner's handbook for real-time analysis*. Kluwer Academic Publishers, 1993.
- [KZF⁺91] Hermann Kopetz, Ralph Zainlinger, Gerhard Fohler, Holger Kantz, Peter Puschner, and Werner Schutz. The design of real-time systems: From specification to implementation and verification. *Software Engineering Journal*, 6(3):72–82, 1991.
- [LBJ⁺95] Sung-Soo Lim, Young Hyun Bae, Gyu Tae Jang, Byung-Do Rhee, Sang Lyul Min, Chang Yun Park, Heonshik Shin, Kunsoo Park, Soo-Mook Moon, and Chong Sang Kim. An accurate worst case timing analysis for RISC processors. *IEEE Transactions on Software Engineering*, 21(7):593–604, 1995.
- [Leh90] John P. Lehoczky. Fixed priority scheduling of periodic task sets with arbitrary deadlines. In *Proceedings of*

-
- the 11th Real-Time Systems Symposium*, pages 201–209. IEEE, 1990.
- [Lin] Linux. Home page, <http://www.linux.com/>.
- [Liu00] J.W.S. Liu. *Real-Time Systems*. Prentice Hall, 2000.
- [LL73] Chung Laung Liu and James W. Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal of the ACM (JACM)*, 20(1):46–61, 1973.
- [LM80] Joseph Y-T Leung and ML Merrill. A note on preemptive scheduling of periodic, real-time tasks. *Information processing letters*, 11(3):115–118, 1980.
- [Loc92] C. Douglass Locke. Software architecture for hard real-time applications: cyclic executives vs. fixed priority executives. *Real-Time Systems*, 4(1):37–53, 1992.
- [LR80] Butler W Lampson and David D Redell. Experience with processes and monitors in mesa. *Communications of the ACM*, 23(2):105–117, 1980.
- [LSST91] John P. Lehoczky, Liu Sha, J. K. Strosnider, and Hide Tokuda. *Fixed priority scheduling theory for hard real-time systems*, pages 1–30. Springer, 1991.
- [LW82] Joseph Y. T. Leung and Jennifer Whitehead. On the complexity of fixed-priority scheduling of periodic, real-time tasks. *Performance evaluation*, 2(4):237–250, 1982.
- [Mar] MaRTE OS. Home page, <http://martel.unican.es/>.
- [MAS] MAST: modelling and analysis suite for real-time systems. <http://mast.unican.es>.
- [MGHG03] José María Martínez, Michael González Harbour, and J. Javier Gutiérrez. RT-EP: Real-time ethernet protocol for analyzable distributed applications on a minimum real-time POSIX kernel. In *Proceedings of the 2nd International Workshop on Real-Time LANs in the Internet Age, RTLIA*, volume 2003, 2003.

- [Mok83] A. K. Mok. *Fundamental Design Problems of Distributed Systems for the Hard Real-Time Environment*. PhD thesis, MIT, Cambridge, Massachusetts. Supervisor: Stephen A. Ward, 1983.
- [MTN08] Jukka Mäki-Turja and Mikael Nolin. Efficient implementation of tight response-times for tasks with offsets. *Real-Time Systems*, 40(1):77–116, 2008.
- [OMG] OMG (Object Management Group). Real-Time Corba Specification. v1.2, 2005, <http://www.omg.org/spec/RT/1.2/>.
- [PAG02] Paulo Pedreiras, Luís Almeida, and Paolo Gai. The FTT-Ethernet protocol: Merging flexibility, timeliness and efficiency. In *Proceedings of the 14th Euromicro Conference on Real-Time Systems*, page 152. IEEE Computer Society, 2002.
- [PB00] Peter Puschner and Alan Burns. Guest editorial: A review of worst-case execution-time analysis. *Real-Time Systems*, 18(2):115–128, 2000.
- [PGGH98] J. C. Palencia, J. Javier Gutiérrez, and Michael González Harbour. Best-case analysis for improving the worst-case schedulability test for distributed hard real-time systems. In *Proceedings of the 10th Euromicro Workshop on Real-Time Systems*, pages 35–44. IEEE, 1998.
- [PGH98] J. C. Palencia and Michael González Harbour. Schedulability analysis for tasks with static and dynamic offsets. In *Proceedings of the 19th IEEE Real-Time Systems Symposium*, pages 26–37. IEEE, 1998.
- [PGH99] J. C. Palencia and Michael González Harbour. Exploiting precedence relations in the schedulability analysis of distributed real-time systems. In *Proceedings of the 20th IEEE Real-Time Systems Symposium*, pages 328–339. IEEE, 1999.
- [PGH03] J. C. Palencia and Michael Gonzalez Harbour. Offset-based response time analysis of distributed systems scheduled under EDF. In *Proceedings of the 15th Euromi-*

- cro Conference on Real-Time Systems*, pages 3–12. IEEE, 2003.
- [PGH05] J. C. Palencia and Michael González Harbour. Response time analysis of EDF distributed real-time systems. *Journal of Embedded Computing*, 1(2):225–237, 2005.
- [PL05] Rodolfo Pellizzoni and Giuseppe Lipari. Improved schedulability analysis of real-time transactions with earliest deadline scheduling. In *Proceedings of the 11th Real Time and Embedded Technology and Applications Symposium RTAS'05*, pages 66–75. IEEE, 2005.
- [RCM96] Ismael Ripoll, Alfons Crespo, and Aloysius K Mok. Improvement in feasibility testing for real-time tasks. *Real-Time Systems*, 11(1):19–39, 1996.
- [Red04] O. Redell. Analysis of tree-shaped transactions in distributed real time systems. In *Proceedings of the 16th Euro-micro Conference on Real-Time Systems*, pages 239–248, 2004.
- [RGGH13] Juan M. Rivas, J. Javier Gutiérrez, and Michael González Harbour. Fixed priorities or edf for distributed real-time systems? *SIGBED Rev.*, 10(2):21–21, 2013.
- [RGGH14] Juan M Rivas, J Javier Gutiérrez, and Michael González Harbour. Gen4mast: A tool for the evaluation of real-time techniques using a supercomputer. In *REACTION 2014. 3rd International Workshop on Real-time and Distributed Computing in Emerging Applications. Rome, Italy.*, 2014.
- [RGPGH10] Juan Maria Rivas, J. Javier Gutiérrez, J. C. Palencia, and Michael González Harbour. Optimized deadline assignment and schedulability analysis for distributed real-time systems with local edf scheduling. In *Proceedings of the 8th International Conference on Embedded Systems and Applications*, 2010.
- [RGPGH11] Juan Maria Rivas, J. Javier Gutiérrez, J. C. Palencia, and Michael González Harbour. Schedulability analysis and optimization of heterogeneous EDF and FP distributed

- real-time systems. In *Proceedings of the 23rd Euromicro Conference on Real-Time Systems (ECRTS)*, pages 195–204. IEEE, 2011.
- [RS02] O. Redell and M. Sanfridson. Exact best-case response time analysis of fixed priority scheduled tasks. In *Proceedings of the 14th Euromicro Conference on Real-Time Systems*, pages 165–172. IEEE, 2002.
- [RSL88] Rangunathan Rajkumar, Lui Sha, and John P Lehoczky. Real-time synchronization protocols for multiprocessors. In *RTSS*, volume 88, pages 259–269, 1988.
- [Sha] S.Ha.R.K. Home page, <http://shark.sssup.it/>.
- [SML88] Jay K. Strosnider, Tom Marchok, and John Lehoczky. Advanced real-time scheduling using the IEEE 802.5 token ring. In *Proceedings of Real-Time Systems Symposium*, pages 42–52. IEEE, 1988.
- [Spu96a] Marco Spuri. Analysis of deadline scheduled real-time systems. Technical report, Institut national de recherche en informatique et en automatique (INRIA), 1996.
- [Spu96b] Marco Spuri. Holistic analysis for deadline scheduled real-time distributed systems. Technical report, Institut national de recherche en informatique et en automatique (INRIA), 1996.
- [SR88] John A. Stankovic and Krithi Ramamritham. *Hard real-time systems*. IEEE Computer Society Press, 1988.
- [SRL90] Lui Sha, Rangunathan Rajkumar, and John P. Lehoczky. Priority inheritance protocols: An approach to real-time synchronization. *IEEE Transactions on Computers*, 39(9):1175–1185, 1990.
- [SSH⁺09] H Shokry, M Shedeed, S Hammad, M Shalan, and A Wahdan. Hardware EDF scheduler implementation on controller area network controller. In *4th International Design and Test Workshop (IDT)*, pages 1–6. IEEE, 2009.

-
- [SSL89] Brinkley Sprunt, Lui Sha, and John Lehoczky. Aperiodic task scheduling for hard-real-time systems. *Real-Time Systems*, 1(1):27–60, 1989.
- [SSRB98] John A Stankovic, Marco Spuri, Krithi Ramamritham, and Giorgio C Buttazzo. *Deadline Scheduling for Real-Time Systems*. Springer, 1998.
- [TBW92] Ken W. Tindell, Alan Burns, and Andy J. Wellings. Allocating hard real-time tasks: an NP-hard problem made easy. *Real-Time Systems*, 4(2):145–165, 1992.
- [TBW94] Ken W. Tindell, Alan Burns, and Andy J. Wellings. An extendible approach for analyzing fixed priority hard real-time tasks. *Real-Time Systems*, 6(2):133–151, 1994.
- [TBW95] Ken Tindell, Alan Burns, and Andy J. Wellings. Calculating controller area network (CAN) message response times. *Control Engineering Practice*, 3(8):1163–1169, 1995.
- [TC94] Ken Tindell and John Clark. Holistic schedulability analysis for distributed hard real-time systems. *Microprocess. Microprogram.*, 40(2-3):117–134, 1994.
- [TDB⁺07] S. Tucker Taft, Robert A. Duff, Randall L. Brukardt, Erhard Ploedereder, and Pascal Leroy. *Ada 2005 Reference Manual. Language and Standard Libraries: International Standard ISO/IEC 8652/1995 (E) with Technical Corrigendum 1 and Amendment 1 (Lecture Notes in Computer Science)*. 2007.
- [THW94] K. W. Tindell, Hans Hansson, and Andy J. Wellings. Analysing real-time communications: controller area network (CAN). In *Proceedings of Real-Time Systems Symposium*, pages 259–263. IEEE, 1994.
- [Tin94] Ken Tindell. Adding time-offsets to schedulability analysis. Technical report, University of York, 1994.
- [YZ08] Jun Yan and Wei Zhang. WCET analysis for multi-core processors with shared L2 instruction caches. In *Proceedings of Real-Time and Embedded Technology and Applications Symposium RTAS'08.*, pages 80–89. IEEE, 2008.

- [ZB09] Fengxiang Zhang and Alan Burns. Improvement to quick processor-demand analysis for EDF-scheduled real-time systems. In *Proceedings of the 21st Euromicro Conference on Real-Time Systems ECRTS'09*, pages 76–86. IEEE, 2009.