

Sistemas Distribuidos de Tiempo Real con Prioridades Fijas

Alejandro Alonso Muñoz
Dept. Ingeniería de Sistemas Telemáticos
ETSI de Telecomunicación
Universidad Politécnica de Madrid

1. Planteamiento General

- ◆ **Objetivo:**
 - ✓ Planificación de SDTR basados en planificación con prioridades fijas
 - ✓ Presentar técnicas de análisis de tiempos de respuesta
- ◆ **Tipo de sistemas:**
 - ✓ Sistemas distribuidos débilmente acoplados
 - ✓ Nodos: planificación con prioridades fijas
 - ✓ Redes: acotado el tiempo de respuesta de los mensajes

Contenido

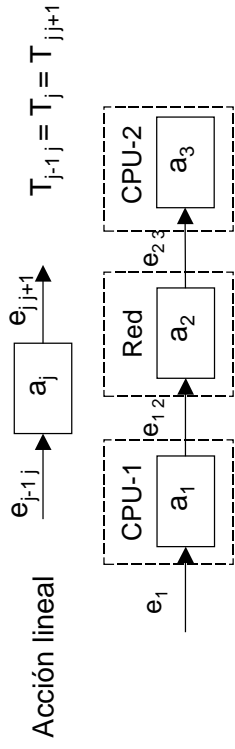
1. Planteamiento general
2. Análisis holístico de SDTR
3. Comunicación con CAN
4. Ejemplo

Modelo de sistema

- ◆ **Sistema hardware compuesto por:**
 - ✓ Conjunto de procesadores
 - ✓ Unidos por redes de comunicaciones
- ◆ **Sistema software compuesto por:**
 - ✓ Tareas asignadas estáticamente a procesadores
 - ✓ Mensajes asignados estáticamente a redes
 - ✓ Comunicación entre tareas:
 - » Si en el mismo nodo: memoria compartida
 - » Si en distintos nodos: paso de mensajes

Modelo lineal de sistema distribuido

- ◆ Un evento genera una secuencia de acciones
- ◆ Una acción es una tarea o un mensaje, en secuencia.
- ◆ Las acciones sólo pueden ser activadas por un evento y sólo pueden activar a una acción.



11/01/02

© 2000 Alejandro Alonso

5

Tareas periódicas con jitter

- ◆ El instante crítico no es el mismo si hay jitter
- ◆ Las tareas periódicas no suelen tener jitter
- ◆ Puede haberlo si la planificación se hace con una granularidad apreciable
- ◆ El tiempo de respuesta se calcula con

$$w_i^{p+1} = C_i + B_i + \sum_{j \in hp(i)} \left\lceil \frac{w_i^p + J_j}{T_j} \right\rceil \cdot C_j$$

$$R_i = w_i^p + J_i \text{ para } w_i^{p+1} = w_i^p$$

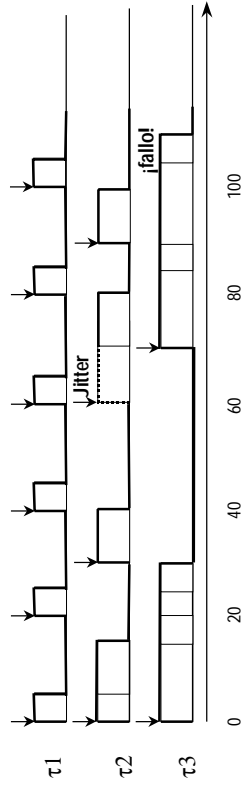
11/01/02

© 2000 Alejandro Alonso

6

Ejemplo de tareas con Jitter

Tarea	T	C	P	D	J
τ_1	20	5	3	20	0
τ_2	30	10	2	30	10
τ_3	70	10	1	35	0



11/01/02

© 2000 Alejandro Alonso

7

Plazos arbitrarios

- ◆ Si el plazo es mayor que el período, puede haber varias activaciones pendientes en un ciclo
- ◆ Para analizar esta situación construimos una sucesión de ventanas $w_i^p(q)$, donde $q + 1$ es el número de activaciones de τ_i .
- ◆ Para cada ventana se obtiene un valor de R_i con

$$w_i^{p+1}(q) = (q + 1) \cdot C_i + B_i + \sum_{j \in hp(i)} \left\lceil \frac{w_i^p(q) + J_j}{T_j} \right\rceil \cdot C_j$$

$$R_i(q) = w_i^p(q) - q \cdot T_i + J_i$$

11/01/02

© 2000 Alejandro Alonso

8

Análisis de planificabilidad (Cont.)

- ◆ Tiempo de respuesta de un mensaje (tarea)

$$w_i^{n+1}(q) = (q + 1) \cdot C_i + B_i + \sum_{j \in hp(i)} \left[\frac{w_j^n(q) + J_j}{T_j} \right] \cdot C_j$$

$$R_i(q) = w_i^n(q) - q \cdot T_i + J_i$$

- ◆ Esta ecuación se resuelve por iteración simple.
- ◆ Si $R_i \leq D_i$, entonces el mensaje M_i (Tarea τ_i) es planificable.

11/01/02

© 2000 Alejandro Alonso

9

2. Análisis holístico de SDTR

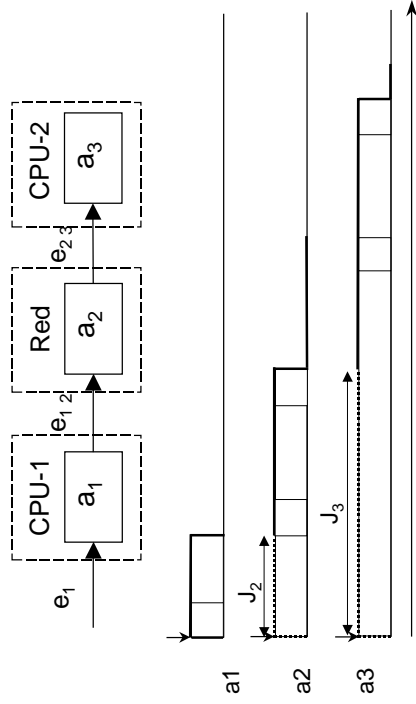
- ◆ Desarrollado por Ken Tindell, 1994
- ◆ El análisis integra los procesadores y las redes.
- ◆ Análisis:
 - ✓ Todos los mensajes y tareas de una transacción se activan al mismo tiempo.
 - ✓ Las relaciones de precedencia se modelan como *jitter*
 - ✓ El retardo en la activación de una tarea (mensaje) depende del tiempo de respuesta del mensaje (tarea) anterior
 - ✓ Se continúa hasta que se complete una activación

11/01/02

© 2000 Alejandro Alonso

10

Análisis holístico de SDTR (Cont.)



11/01/02

© 2000 Alejandro Alonso

11

3. Comunicación con CAN

- ◆ CAN (*Controller Area Network*) es un protocolo de comunic. en serie, radiado y sensible a portadora.
- ◆ Transmisión y recepción de mensajes pequeños, en ambientes ruidosos.
- ◆ Velocidad máxima: 1Mbit/s. Distancia máxima: 40 m.
- ◆ La resolución de la contención está basada en la prioridad de los mensajes.
- ◆ El identificador describe el contenido del mensaje y es su prioridad.
- ◆ Consistencia de los datos: todos los nodos reciben los mismos datos.

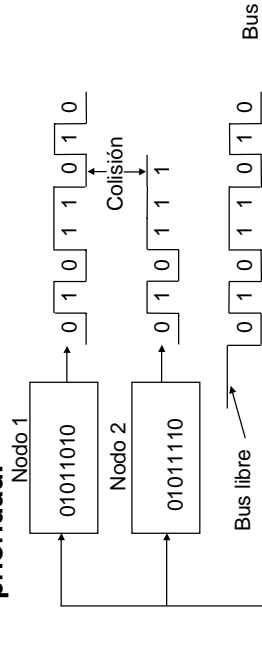
11/01/02

© 2000 Alejandro Alonso

12

Arbitraje del bus

- ◆ Para resolver las colisiones se emplea el campo de arbitraje: identificador y bit de petición remota.
- ◆ Si un nodo envía un bit recesivo (1) y lee un bit dominante (0), hay un nodo transmitiendo con más prioridad.



11/01/02

© 2000 Alejandro Alonso

13

Análisis de planificabilidad

- ◆ El acceso al bus CAN está basado en prioridades.
- ◆ Condición: El controlador del bus transmite siempre que haya mensajes pendientes.
- ◆ No hay expulsión durante la transmisión. El mensaje más largo ocupa 0'13ms (B).
- ◆ Se supone que los mensajes se generan al principio del periodo y no hay errores de transmisión.

Tareas		Mensajes	
T	Periodo	T	Periodo
D	Plazo de respuesta	D	Plazo de respuesta
B	Bloqueo por región crítica ocupada	B	Bloqueo por región crítica ocupada

11/01/02

© 2000 Alejandro Alonso

15

Prioridad de los mensajes

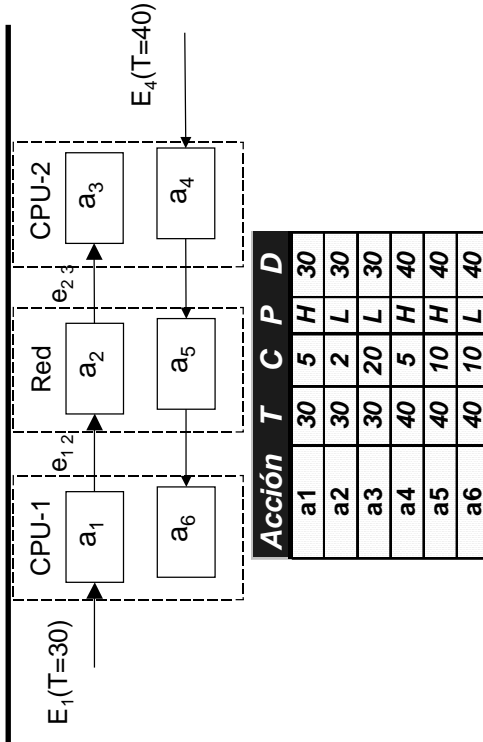
- ◆ El identificador es la prioridad del mensaje
- ◆ Cuanto menor es, mayor es la prioridad.
- ◆ Para que este método funcione se debe cumplir:
 - ✓ Los mensajes tienen identificador único.
 - ✓ No hay dos nodos transmitiendo mensajes con el mismo identificador.

11/01/02

© 2000 Alejandro Alonso

14

4. Ejemplo



11/01/02

© 2000 Alejandro Alonso

16

Lecturas adicionales

- ♦ J. Javier Gutiérrez García
Planificación, análisis y optimización de sistemas distribuidos de tiempo real estricto.
Tesis doctoral, Universidad de Cantabria, 1995
(Notación empleada en la presentación)
- ♦ K. Tindell, J. Clark
"Holistic Schedulability Analysis for Distributed Hard Real-Time Systems
Microprocessing & Microprogramming, V. 50, N 2-3, Abril de 1994

11/01/02

© 2000 Alejandro Alonso

17

Solución al problema

Acción	T	C	P	D	R
a1	30	5	H	30	5
a2	30	2	L	30	17
a3	30	20	L	30	42
a4	40	5	H	40	5
a5	40	10	H	40	15
a6	40	10	L	40	30

11/01/02

© 2000 Alejandro Alonso

18