

Introducción a los Sistemas Distribuidos de Tiempo Real

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



1. Planteamiento General

- ✓ La capacidad de procesamiento está distribuida entre varios computadores interconectados.
- ✓ Las actividades del sistema tienen requisitos de tiempo.
- ✓ Necesidad de sistemas distribuidos:
 - Requisitos de procesamiento.
 - Distribución física del sistema.
 - Fiabilidad: Tolerancia a fallos.
- ✓ Los sistemas distribuidos de tiempo real (SDTR) son complicados de realizar.

11/01/02

©Alejandro Alonso Muñoz

3

Contenido

1. Planteamiento general
2. Arquitectura de SDTR
3. Comunicación en SDTR
4. Planificación de SDTR
5. POSIX 1003.21: comunicaciones en STDR
6. Conclusiones

11/01/02

©Alejandro Alonso Muñoz

2



Planteamiento general (Cont.)

- ✓ Se consideran sistemas débilmente acoplados.
- ✓ Comunicación mediante mensajes
- ✓ El tiempo de comunicación es significativo.
- ✓ Ej. Sistemas multimedia, SCADA, aviónica, fabricación integrada, robótica.
- ✓ Distintos tipos de requisitos temporales
- ✓ Se consideran, fundamentalmente, sistemas críticos

11/01/02

©Alejandro Alonso Muñoz

4



Problemas específicos de SDTR

- ✓ Planificación del medio de comunicación.
- ✓ Asignación de tareas a procesadores.
- ✓ Plazos de respuesta globales (end-to-end).
- ✓ Sincronización de relojes.
- ✓ Tolerancia a fallos.

11/01/02

©Alejandro Alonso Muñoz

5

Medio de Comunicación

- ✓ En sistemas monoprocesador, el procesador suele ser el único recurso a planificar.
- ✓ En SDTR, hay que planificar el medio de comunicación.
- ✓ Los mensajes suelen tener un plazo desde que se solicita su envío hasta que se reciben.
- ✓ Para garantizar estos plazos y acotar el tiempo de envío: protocolos de comunicación **deterministas**.
- ✓ En muchos casos la arquitectura de niveles OSI introduce demasiada sobrecarga.

11/01/02

©Alejandro Alonso Muñoz

6

Distribución de carga

- ✓ Un problema adicional es asignar tareas a procesadores.
- ✓ Puede ser estática o dinámica.
- ✓ Una distribución no adecuada puede infrautilizar recursos e impedir el cumplimiento de plazos.
- ✓ Criterios de asignación:
 - Los procesadores tienen recursos limitados.
 - Replicación de tareas.
 - Requisitos de utilización de recursos específicos.
 - Distribución geográfica.

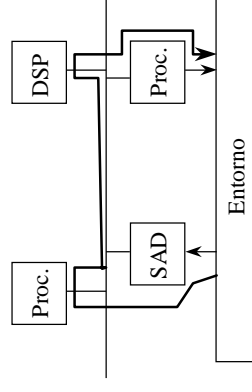
11/01/02

©Alejandro Alonso Muñoz

7

Plazos de respuesta globales

- ✓ En SDTR: plazo de respuesta asociado a una transacción.
- ✓ Transacción: conjunto de actividades relacionadas que se sincronizan/comunican mediante mensajes.
- ✓ Ejemplo de plazo global.



11/01/02

©Alejandro Alonso Muñoz

8

Sincronización de relojes

- ✓ Granularidad del tiempo: suficiente para la aplicación.
- ✓ Los procesadores deben tener la misma visión del tiempo.
- ✓ Los relojes presentan variaciones que obligan a sincronizarlos.
- ✓ La diferencia entre los valores locales del tiempo de observación del mismo evento en diferentes procesadores debe estar **acotada**.
- ✓ La sincronización de los relojes no debe degradar el rendimiento del sistema.

11/01/02

©Alejandro Alonso Muñoz

9

Tolerancia a fallos

- ✓ El sistema produce resultados correctos, aún en presencia de fallos hardware y/o software.
- ✓ En SDTR las actividades se completan en el plazo adecuado.
- ✓ Las técnicas de tolerancia a fallos para sistemas de tiempo real están muy poco maduras.
- ✓ Se deben considerar fallos en procesadores y en los medios de comunicación.

11/01/02

©Alejandro Alonso Muñoz

10

2. Arquitectura de SDTR

- ✓ Compuesta por un conjunto de nodos conectados por redes de comunicación.
- ✓ Cada nodo puede ser un multiprocesador conectado por buses.
- ✓ Sólo consideraremos sistemas débilmente acoplados.
- ✓ Es frecuente que estos sistemas tengan gran complejidad.
- ✓ Se suelen estructurar jerárquicamente, para tratar con:
 - Plazos de respuesta de diversas escalas.
 - Tipo de información.
 - Requisitos de procesador

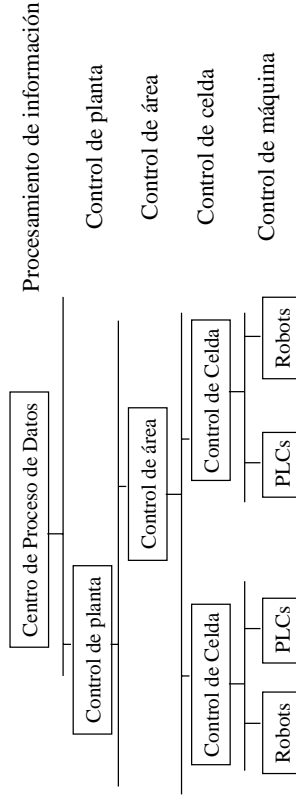
11/01/02

©Alejandro Alonso Muñoz

11

Sistemas de Fabricación

- ✓ Sistema integrado de fabricación: Automatización, Integración y Flexibilidad.
- ✓ Modelo simplificado de un sistema integrado de fabricación.



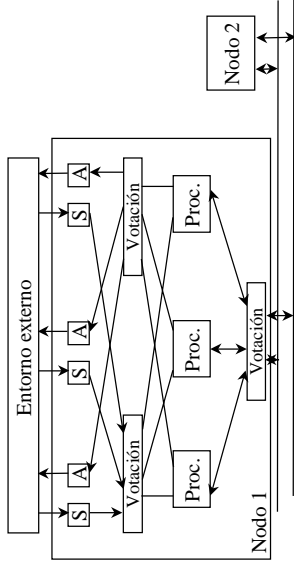
11/01/02

©Alejandro Alonso Muñoz

12

Sistemas de aviónica

- ✓ Requisitos de seguridad muy exigentes.
- ✓ Necesidad de tolerar fallos hardware y software.
- ✓ Se utilizan replicas y métodos de votación.



11/01/02

©Alejandro Alonso Muñoz

13

3. Comunicación en SDTR.

- ✓ El tiempo de comunicación es significativo
- ✓ Los mensajes tienen requisitos temporales.
- ✓ El medio debe ser determinista.
- ✓ Hay pocos protocolos orientados a tiempo real.
- ✓ El protocolo CSMA/CD no es determinista.
- ✓ En realizaciones prácticas hay que considerar las actividades necesarias para mandar/recibir los mensajes:
 - Generación
 - Encolamiento
 - Distribución
 - Transmisión.

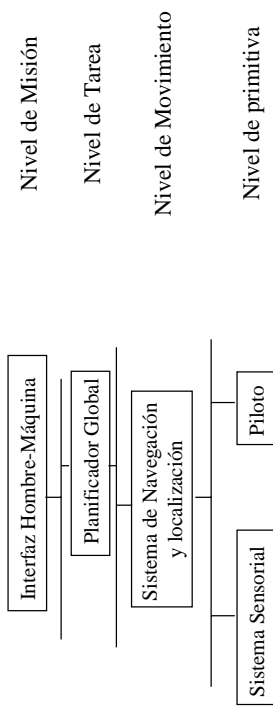
11/01/02

©Alejandro Alonso Muñoz

15

Sistemas de robótica

- ✓ La arquitectura refleja los niveles de control involucrados.
- ✓ Cada nivel trata información con granularidad diferente.
- ✓ Esquema de una arquitectura del sistema de control de un robot.



11/01/02

©Alejandro Alonso Muñoz

14

TDMA

- ✓ TDMA: Time Division Multiple Access.
- ✓ El envío de los mensajes es síncrono.
- ✓ El acceso al medio se divide en rodajas de tiempo.
- ✓ Los mensajes se envían en rodajas específicas.
- ✓ El inicio se indica mediante un reloj global.
- ✓ La planificación del medio es estática.
- ✓ Este protocolo es predecible y determinista.
- ✓ Sin embargo, es poco flexible, difícil de mantener y se puede desaprovechar la red.

11/01/02

©Alejandro Alonso Muñoz

16

Paso de testigo

- ✓ **Priority-Driven Protocol.**
 - Política de arbitraje basada en prioridades.
 - El testigo tiene un campo con prioridad.
 - Emite el nodo con mensaje más prioritario.
 - Se emplea sobre IEEE 802.5.
 - Planificación basada en prioridades.
 - Pocos niveles de prioridad.
 - Útil en redes con velocidades 1-10 Mbits.

11/01/02

©Alejandro Alonso Muñoz

17

Paso de testigo (cont.)

- ✓ **Timed Token Protocol.**
 - El ancho de banda se asigna cíclicamente.
 - Cada estación envía durante cierto tiempo cuando tiene el testigo.
 - El tiempo de envío debe ser suficiente para enviar los mensajes en su plazo.
 - Se usa con redes rápidas: FDDI.

11/01/02

©Alejandro Alonso Muñoz

18

CAN Bus

- ✓ CAN: Controller Area Network.
- ✓ Los mensajes tienen un identificador.
- ✓ El identificador indica la prioridad.
- ✓ Resolución dinámica de colisiones.
- ✓ Es posible aplicar el análisis de planificabilidad de sistemas con prioridades estáticas.
- ✓ La velocidad de la red es 1Mbit/sec y el mensaje más largo ocupa 130ms.

11/01/02

©Alejandro Alonso Muñoz

19

6. Planificación de SDTR.

- ✓ Problema muy complejo.
- ✓ Es necesario considerar la planificación de un conjunto de actividades.
- ✓ No hay un enfoque generalmente aceptado.
- ✓ Hay diversas aproximaciones al problema orientadas a dominios concretos.
- ✓ Un objetivo común consiste en integrar gestión de tareas y recursos y mecanismos de tolerancia a fallos.
- ✓ Gran variedad de sistemas operativos en desarrollo: MARS, Spring, CHAOS, RT-Mach, Chimera, DEDOS, etc.

11/01/02

©Alejandro Alonso Muñoz

20

Síncrona-Cíclica

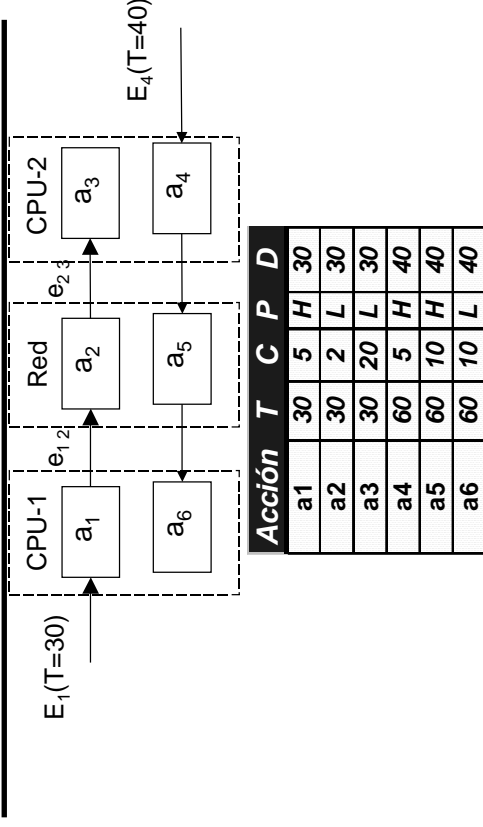
- ✓ Planifica estáticamente la red y los procesadores.
- ✓ En los procesadores se usan planificadores cíclicos.
- ✓ La red es del tipo TDMA.
- ✓ Se determina estáticamente el instante de envío de los mensajes.
- ✓ Este enfoque se aplica en MARS.
- ✓ El método es predecible, determinista, inflexible y no optimiza el uso de recursos.
- ✓ MARS incorpora mecanismos avanzados de tolerancia a fallos.

11/01/02

©Alejandro Alonso Muñoz

21

Ejemplo



11/01/02

©Alejandro Alonso Muñoz

22

Ejemplo (Cont.)

CPU-1			
Tiempo	Acción	WCET	
0	Inicio a1	5	
4	Envío a2	1	
5	Fin a1		
20	Inicio a6	10	
30	Fin a6		
30	Inicio a1	5	
34	Envío a2	1	
35	Fin a1		

Tiempo	Dirección	D
4	m1'address	S
6	m5'address	E
34	m1'address	S

CPU-2			
Tiempo	Acción	WCET	
0	Inicio a4	5	
4	Envío a5	1	
5	Fin a4		
10	Inicio a3	20	
30	Fin a3		
40	Inicio a3	20	
60	Fin a3		

Tiempo	Dirección	D
4	m1'address	E
6	m5'address	S
34	m1'address	E

Basada en prioridades estáticas

- ✓ Mismos principios para planificar el medio de comunicación y los procesadores.
- ✓ La red es de paso de testigo o CAN-Bus.
- ✓ Se determinan los tiempos de respuesta más desfavorables de tareas y mensajes.
- ✓ El tiempo de respuesta global se calcula como la suma.
- ✓ Los periodos de las tareas/mensajes deben ser armónicos.
- ✓ Permite un mejor aprovechamiento de recursos.
- ✓ No hay métodos de tolerancia a fallos suficientemente maduros.

11/01/02

©Alejandro Alonso Muñoz

23

11/01/02

©Alejandro Alonso Muñoz

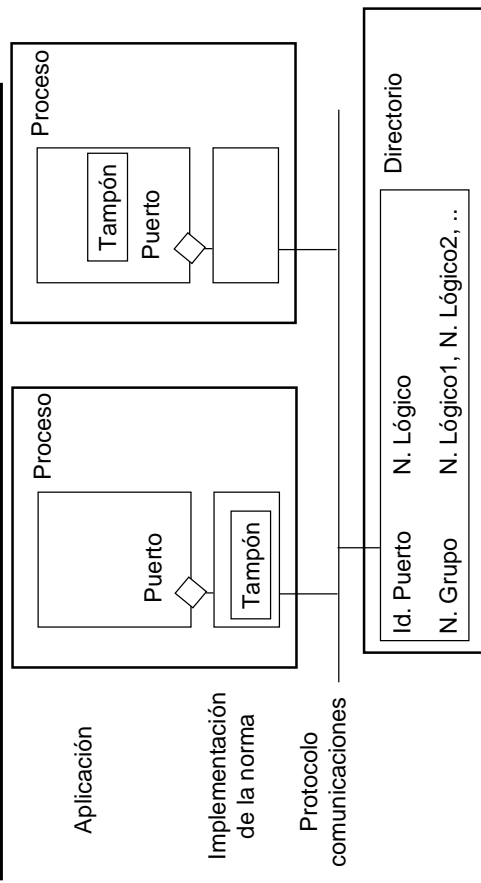
24

- ✓ Las decisiones de planificación (parte) se realizan en tiempo de ejecución.
- ✓ Spring es un sistema operativo de acuerdo con este enfoque.
- ✓ Planificación integrada de procesos y recursos.
- ✓ Se considera información semántica de las tareas para la planificación.
- ✓ Planificación a cuatro niveles.
 - Despachador.
 - Planificador local.
 - Planificador distribuido.
 - Controlador metanivel.

- ✓ Gestión de tampones (buffers)
- ✓ Bloques de control de mensajes
- ✓ Operaciones asíncronas y síncronas
- ✓ Bloqueos limitados por la implementación
- ✓ Mensajes con prioridades
- ✓ Mensajes con etiqueta

4. POSIX 1003.21:
comunicaciones en STDR

- ✓ Definición de interfaz para las operaciones necesarias para la comunicación en SDTR.
- ✓ Portabilidad del código fuente.
- ✓ Interfaz independiente del protocolo.
- ✓ Especificación independiente del lenguaje.
- ✓ La norma no ha sido aún aprobada.



Componentes de interfaz

- ✓ Puertos de comunicación (*endpoints*):
 - Las aplicaciones los usan para enviar y recibir.
 - Permiten identificar a la implementación el origen y destino de los mensajes.
 - Local al proceso que lo crea.
 - Se obtiene un nombre lógico asociado para enviar.
- ✓ Nombres lógicos: Identifica a uno o varios puertos.
- ✓ Grupos de radiado múltiple:
 - Conjunto de puertos con un nombre lógico.

11/01/02

©Alejandro Alonso Muñoz

29

Componentes de interfaz (Cont.)

- ✓ Mensajes:
 - Contenedor de datos.
 - Incluye una cabecera con atributos de la implementación.
 - Un atributo es la prioridad.
- ✓ Tampones (*Buffer*):
 - Sirven para almacenar mensajes.
 - La aplicación o la implementación pueden gestionar la memoria asociada.

11/01/02

©Alejandro Alonso Muñoz

30

Operaciones de transferencia de datos

- ✓ Servicios de directorio: Gestión de los nombres.
- ✓ Establecimiento de la comunicación:
 - Definición de atributos de los puertos y mensajes
- ✓ Operaciones sincronas y asincronas.
- ✓ Modelos de comunicación.
 - Modelo sencillo.
 - Modelo radiado.
 - Modelo de radiado múltiple.
 - Modelo de mensajes con etiqueta.
 - Combinación de modelos.

11/01/02

©Alejandro Alonso Muñoz

31

Operaciones de transferencia de datos (Cont.)

- ✓ Operaciones de recepción:
 - Con/sin tampón.
 - De uno/varios puertos.
- ✓ Operaciones de envío: Se indica:
 - Puerto origen y destino.
 - Longitud del mensaje.
 - Dirección del mensaje a enviar.
- ✓ Operaciones de registro, según los modelos:
 - Puerto para recibir los mensajes radiados
 - Etiquetas que se quieren recibir.
 - Puerto que se incluye en un grupo de radiado.
 - Nombre lógico de un puerto.

11/01/02

©Alejandro Alonso Muñoz

32

Características para tiempo real.

- ✓ Control de la memoria de los tampones (buffers)
- ✓ Interacción asíncrona y síncrona con la implementación
- ✓ Tiempos de bloqueo acotados.
- ✓ Mensajes con prioridad.
- ✓ Mensajes con etiqueta.
- ✓ Se puede emplear un protocolo de comunicación predecible.

11/01/02

©Alejandro Alonso Muñoz

33

7. Conclusiones

- ✓ SDTR: Hay soluciones aplicables en sistemas críticos:
 - Planificación síncrona y basada en prioridades
- ✓ Aún hay problemas, pendientes de resolución.
 - Tolerancia ante fallos
 - Protocolos de comunicación
 - Distribución de carga
- ✓ Es necesario investigar métodos:
 - con mayor nivel de abstracción.
 - que integren las soluciones a los diversos problemas.

11/01/02

©Alejandro Alonso Muñoz

34