

INTRODUCCIÓN A LA TOLERANCIA A AVERÍAS

Definición

Un sistema es tolerante a averías si puede enmascarar la presencia de averías en el sistema usando redundancia

Objetivo de las técnicas de tolerancia a averías

Evitar que falle el sistema, incluso ante la presencia de averías en el mismo

Un sistema ...

NO PUEDE hacerse tolerante a averías contra sus propios fallos
(Cuando el sistema global ya ha fallado no se puede hacer nada para evitar el fallo)

SÍ PUEDE hacerse tolerante a averías contra el fallo de sus componentes
(El sistema global enmascara el fallo de un subsistema a los niveles superiores)

Sistema tolerante a averías

Su comportamiento externo (servicios, propiedades, etc.) es consistente con las especificaciones, aún en presencia de averías

REDUNDANCIA

Son las partes de un sistema que son innecesarias para su correcto funcionamiento, si no se desea tener tolerancia a averías

**El sistema trabaja correctamente sin redundancia
SI NO se producen averías**

Tipos de redundancia {
1) Hardware
2) Software
3) Tiempo
4) Información

**En general se usan de forma combinada
los diversos tipos de redundancia**

FASES EN LOS MECANISMOS DE TOLERANCIA A AVERÍAS

La implementación de un mecanismo de tolerancia a averías en un sistema es muy dependiente del sistema, su arquitectura y su diseño

- NO HAY una técnica general para “añadir” tolerancia a averías a un sistema
- SÍ HAY principios básicos útiles para diseñar sistemas tolerantes a averías

Cualquier mecanismo o esquema para soportar tolerancia a averías debe realizar cuatro actividades sucesivas:

- 1) Detección del error
- 2) Confinamiento (localización, aislamiento) de los daños
- 3) Recuperación del error
- 4) Tratamiento de la avería y continuación del servicio

FASE DE DETECCIÓN DEL ERROR

Introducción

Las averías y los fallos no se pueden observar directamente sino que tienen que deducirse de la presencia de errores

El error aparece en el estado del sistema (o subsistema)
Hay que realizar pruebas sobre el estado para ver si hay error o no

La eficacia de un esquema de tolerancia a averías depende en gran medida de la efectividad del mecanismo de detección de errores empleado

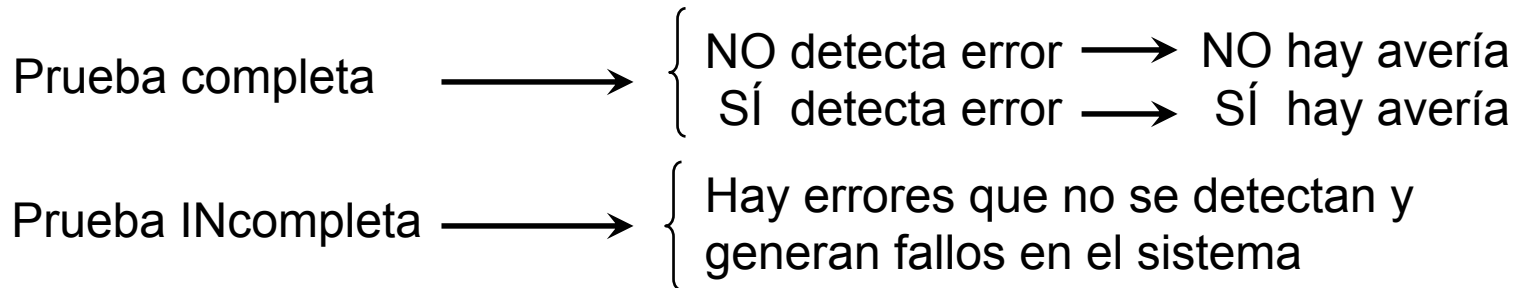
IDEAL → Detecta cualquier error causado por las averías que pretende manejar el esquema de tolerancia a averías

PERO ES IMPOSIBLE → Concepto de prueba ideal para la detección de errores

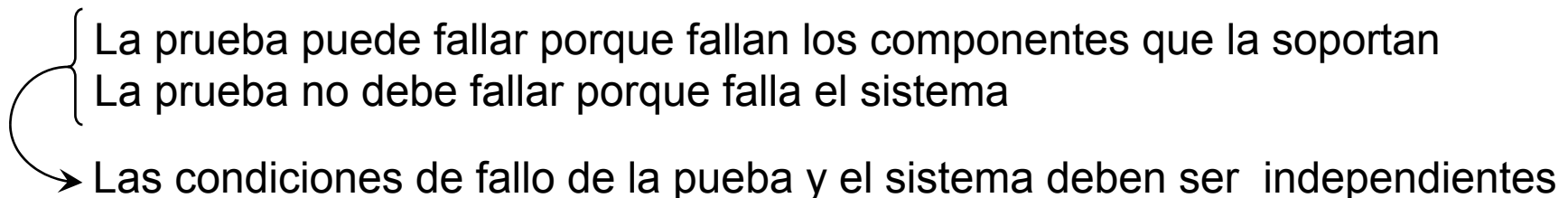
CONCEPTO DE PRUEBA IDEAL PARA LA DETECCIÓN DE ERRORES

Una prueba ideal es la que satisface las siguientes propiedades:

- 1** Es determinada exclusivamente por las especificaciones del sistema
NO debe estar influenciada por el diseño interno del sistema
- 2** Es completa y correcta: debe detectar TODOS los errores posibles en el sistema que se pueden originar por las averías que hay que manejar



- 3** Es independiente del sistema



PRUEBAS DE DETECCIÓN DE ERRORES EN SISTEMAS REALES

Las pruebas en sistemas reales ...

- Raramente pueden satisfacer las propiedades de una “prueba ideal”
- Tan sólo pueden aproximarse a una “prueba ideal”

Problema 1

Las pruebas prácticas no se realizan en todos los $\left\{ \begin{matrix} \text{estados} \\ \text{instantes} \end{matrix} \right\}$ posibles

Se usa información sobre la estructura interna y el diseño del sistema para reducir el espacio de observación

Problema 2

Realizar una prueba completa ...

NO suele ser factible debido a restricciones de $\left\{ \begin{matrix} \text{Complejidad} \\ \text{Coste} \\ \text{Prestaciones} \end{matrix} \right\}$

Problema 3

Es muy difícil conseguir una independencia total entre prueba y sistema pues comparten elementos: alimentación, caja, entorno ...

CONCEPTO DE PRUEBA ACEPTABLE PARA LA DETECCIÓN DE ERRORES

Una prueba aceptable para la detección de errores ...

- Es una aproximación de la prueba ideal
- Objetivos {
 - Mantener el coste de la prueba a un nivel aceptable
 - Maximizar los errores que son detectados
- Problema: NO garantizan que no haya errores no detectados
Intentan capturar la “mayoría” de los errores de interés
(sobre todo los que ocurren más frecuentemente)

Tipos de pruebas de detección de errores = f [Tipo de sistema
Fallos de interés]

- 1) De replicación
- 2) De temporización
- 3) Estructurales y de codificación
- 4) De razonabilidad
- 5) De diagnóstico

PRUEBAS DE DETECCIÓN DE ERRORES BASADAS EN REPLICACIÓN (1)

Consisten en replicar algún componente del sistema

Los resultados de los componentes son comparados para detectar errores

El tipo y número de réplicas depende de cada aplicación

Requisitos para aplicar la replicación

- 1) El diseño del componente replicado es correcto
- 2) Los fallos de los componentes se deben a causas físicas
- 3) Los componentes fallan independientemente unos de otros

Aplicación típica

En el diseño de un subsistema hardware

El esquema más típico es el denominado de redundancia modular triple (TMR, Triple Modular Redundancy)

PRUEBAS DE DETECCIÓN DE ERRORES BASADAS EN REPLICACIÓN (2)

Ventajas

Es una prueba que puede ser completa

Puede realizarse sin conocer la estructura interna del componente replicado

Es potente y eficiente

Inconveniente

Es muy cara debido a la replicación

DETECCIÓN DE AVERÍAS EN EL DISEÑO

- La técnica de replicación usando copias **IDÉNTICAS** de un componente no funciona si el diseño del componente es incorrecto (todas las réplicas funcionarán mal)
- La técnica de replicación puede detectar averías de diseño si el diseño de cada componente es independiente de los otros

PRUEBAS DE DETECCIÓN DE ERRORES BASADAS EN TIEMPO

Si las especificaciones de un componente incluyen restricciones temporales
ENTONCES

Se pueden usar pruebas de tiempo para ver si se cumplen las restricciones

Implementación

Se basa en el uso de temporizadores

Si el temporizador expira → **ERROR**: Se viola la restricción temporal

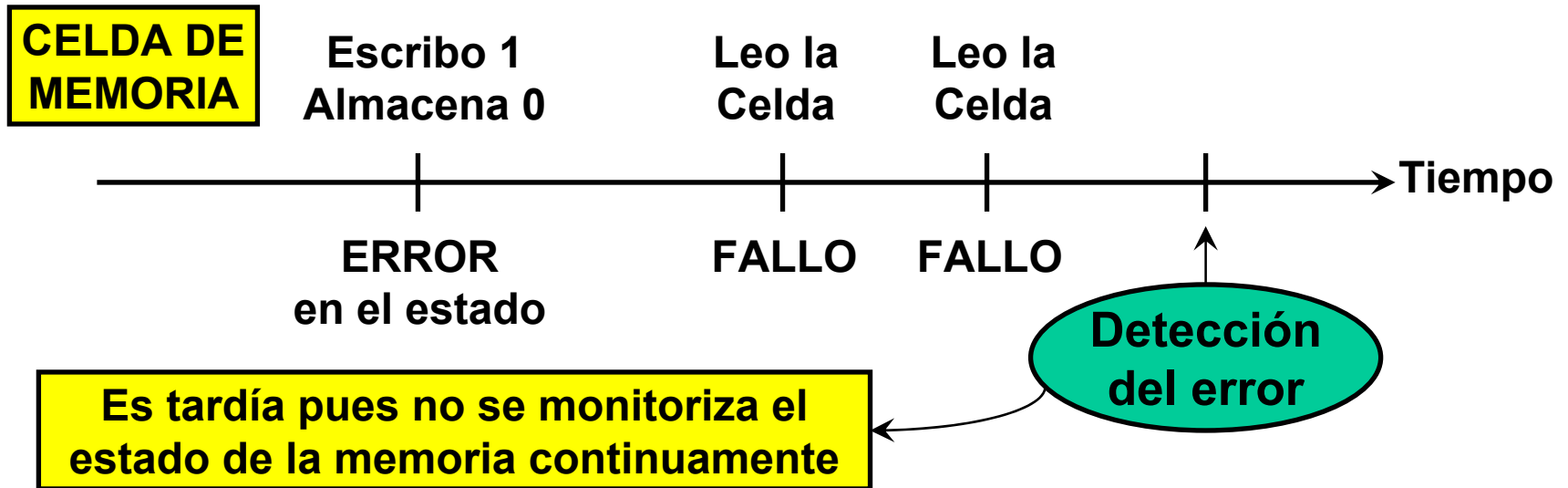
Aplicaciones típicas

- ① En los subsistemas hardware se suelen usar temporizadores para detectar situaciones problemáticas (suelen denominarse “wachtdog timers”)
- ② En los sistemas operativos se usan frecuentemente temporizadores para detectar situaciones de bloqueo indefinido
- ③ En sistemas distribuidos cada nodo debe responder dentro de un plazo de tiempo razonable (Si no lo hace se considera que ha fallado)

FASE DE CONFINAMIENTO Y VALORACION DE LOS DAÑOS (1)

Al detectar un error en el estado del sistema $\xrightarrow[\text{DEDUCE}]{\text{SE}}$ $\left\{ \begin{array}{l} \text{Que hay averías en algún componente} \\ \text{Que se han producido fallos} \end{array} \right.$

Ejemplo:



El error puede haberse propagado a $\left\{ \begin{array}{l} \text{Otros componentes del sistema} \\ \text{y/o} \\ \text{Otras partes del estado del sistema} \end{array} \right.$

FASE DE CONFINAMIENTO Y VALORACIÓN DE LOS DAÑOS (2)

Antes de corregir el estado \longrightarrow Determinar exactamente las partes corruptas

MÉTODO

- ① Hacer suposiciones sobre $\left\{ \begin{array}{l} \text{La fuente del error} \longrightarrow \text{Aspecto espacial} \\ \text{y} \\ \text{Cuándo se generó} \longrightarrow \text{Aspecto temporal} \end{array} \right. +$
- ② Analizar todos los flujos de información entre los componentes del sistema para determinar hasta dónde se ha propagado el error

RESULTADO

- Se obtienen los límites del estado que puede estar afectado por el error
- Los posibles daños quedan confinados a esos límites

FASE DE CONFINAMIENTO Y VALORACIÓN DE LOS DAÑOS (3)

TÉCNICAS PARA OBTENER LOS LÍMITES

DINÁMICAS

Basadas en registrar y examinar todo del flujo de información
Son técnicas muy complejas

ESTÁTICAS

Basadas en la incorporación de “cortafuegos” en el sistema
Estos aseguran que el error no se puede propagar cruzándolos
Son las técnicas más usadas

FRECUENTEMENTE ...

- La fase de confinamiento+valoración no se realiza explícitamente
- En la fase siguiente (recuperación del error) se hacen suposiciones sobre el confinamiento usando información sobre la estructura del sistema

FASE DE RECUPERACIÓN DEL SISTEMA DEL ERROR (1)

En esta fase se eliminan los errores del estado del sistema, restaurando el estado a unos valores consistentes

Técnicas {
 Recuperación hacia atrás (backward recovery)
 Recuperación hacia adelante (forward recovery)

Recuperación hacia atrás

El estado del sistema es restaurado a un estado previo que se supone libre de errores

- ① El estado del sistema es controlado y almacenado periódicamente en algún almacenamiento que no es afectado por los fallos
- ② Cuando se detecta un error, el sistema es devuelto al último estado almacenado

(En inglés 1+2 se denominan: Checkpoint + Rollback)

FASE DE RECUPERACIÓN DEL SISTEMA DEL ERROR (2)

VENTAJAS: Técnica muy general que no depende de la naturaleza de la avería
Si la avería es transitoria tan solo precisa continuar el funcionamiento

PROBLEMA: Genera una gran sobrecarga en el sistema

Recuperación hacia adelante

NO hay un estado previo libre de errores al que se pueda devolver el sistema

Se intenta “ir hacia adelante” construyendo un estado libre de errores a base de hacer correcciones en el estado actual

VENTAJA: Genera poca sobrecarga en el sistema

PROBLEMAS: Requiere una diagnosis sobre el confinamiento (localización) y valoración de los daños en el estado muy exacta

La diagnóstico (y esta técnica) es totalmente dependiente del sistema y de la aplicación

TRATAMIENTO DEL FALLO Y CONTINUIDAD DEL SERVICIO (1)

Las tres primeras fases se centran en los errores

Con el estado del sistema libre de errores se hace ...

Si el error fue causado por una avería de tipo:

{
Detección
Propagación
Eliminación

Transitorio

Basta continuar funcionando a partir del estado libre de errores

Como la avería ha desaparecido, no se producirán nuevos errores

Permanente

Si se continúa funcionando, la avería generará nuevos fallos

Es necesario reparar el componente averiado antes de seguir funcionando

Esta fase tan sólo tiene sentido con averías de tipo permanente

Hay 2 subfases importantes {
Localización de la avería
Reparación del sistema

TRATAMIENTO DEL FALLO Y CONTINUIDAD DEL SERVICIO (2)

① Subfase de LOCALIZACIÓN DE LA AVERÍA

Consiste en identificar el componente que se ha averiado

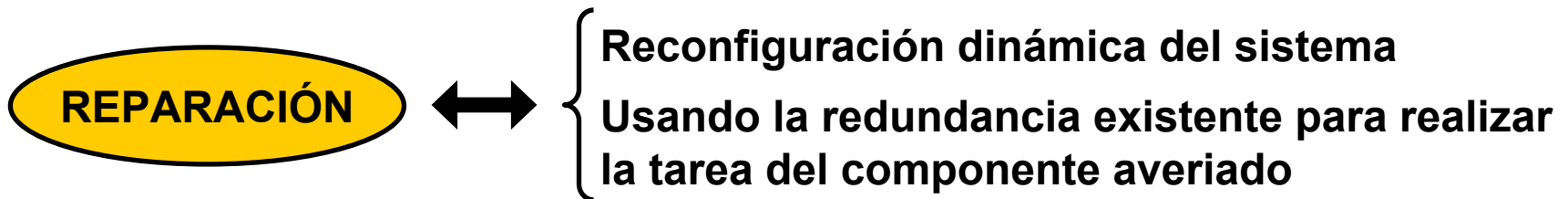
Normalmente, se supone que el componente averiado es el más relacionado con la fuente del error

Si NO puedo identificar el componente averiado la reparación es imposible

② Subfase de REPARACIÓN DEL SISTEMA

Consiste en: NO usar el componente averiado o
usarlo con una configuración diferente

Importante: La reparación se hace on-line sin intervención manual
SI NO el sistema NO puede considerarse tolerante a averías



TRATAMIENTO DEL FALLO Y CONTINUIDAD DEL SERVICIO (3)


ESTRATEGIA DE REPARACIÓN MÁS SIMPLE

Repuesto siempre preparado para dar servicio (Standby Spare)

Consiste en disponer de un componente de repuesto siempre listo para entrar en servicio cuando falle el componente primario, sustituyéndolo

Tras la reparación, el sistema continúa dando el servicio

**Efecto global en el servicio
(gracias al sistema de tolerancia a averías)**



El sistema nunca queda indisponible para los usuarios

Aunque puede haber {
Discontinuidades mínimas del servicio
Alguna degradación de prestaciones

COBERTURA DE AVERÍAS (1)

También denominada { Tasa de cobertura
Coeficiente de cobertura (fault coverage)

Definición intuitiva (concepto) →

Basada en las fases típicas de los mecanismos de tolerancia a averías

Es una medida de la capacidad que tiene un sistema para:

- Detectar
 - Localizar
 - Contener los efectos de
 - Recuperarse de
- } Averías

La cobertura de averías se puede desglosar en diferentes coberturas según la fase que se considere:

Es la fracción (porcentaje) de averías ...

- que pueden ser detectadas
- que pueden ser localizadas
- cuyos efectos pueden ser contenidos
- de las que el sistema es capaz de recuperarse

COBERTURA DE AVERÍAS (2)

Definición matemática

Cobertura en la recuperación de fallos ...

Es la probabilidad condicional de que ocurrido un fallo el sistema se recupere

Suelen considerarse 2 coberturas básicas

- 1) Cobertura de detección de averías (fault detection coverage)**
- 2) Cobertura de tolerancia de averías (fault tolerance coverage)**

**Para obtener un valor numérico de estas coberturas,
del número total de averías que pueden producirse se estima:**

- 1) El N° de averías que puede detectar el sistema**
- 2) El N° de averías que puede tolerar el sistema**

COBERTURA DE AVERÍAS (3)

Definición intuitiva (concepto) —————→

Basada en las técnicas y métodos usados para obtener una dependabilidad elevada

Métodos para CONSEGUIR una dependabilidad elevada

- Prevención de averías (fault avoidance)
- Tolerancia a averías (fault tolerance)

Métodos para ANALIZAR y VALIDAR la dependabilidad

- Eliminación de averías (fault removal)
- Predicción de averías (fault prediction)

SEGÚN ESTOS MÉTODOS SE HABLA DE ...

Cobertura en la prevención de averías	Es muy difícil de estimar
Cobertura en la tolerancia a averías	Fracción de averías que el sistema puede tolerar
Cobertura en la eliminación de averías	Fracción de averías detectables en fase de test
Cobertura en la predicción de averías	No se suele considerar

TÉCNICAS DE TOLERANCIA A AVERÍAS BASADAS EN HARDWARE (1)

Las técnicas más comunes para alcanzar una determinada tolerancia a averías en el hardware se basan en el uso de hardware redundante

Tipos de redundancia hardware { Estática
Dinámica
Híbrida

Sistemas Estáticos

Se diseñan para tolerar averías enmascarándolas (sus efectos)
NO requieren acciones específicas del propio sistema

Sistemas Dinámicos

Se diseñan para tolerar averías detectándolas y reconfigurando el sistema
Sí requieren acciones específicas del propio sistema

Sistemas Híbridos

Combinan las dos técnicas anteriores

- Enmascaramiento: prevenir que se propaguen los errores
- Detección y reconfiguración: sustituir los componentes averiados

TOLERANCIA A AVERÍAS HARDWARE

Diseños Basados en Redundancia Estática

Se basan en un mecanismo de votación que compara las salidas de varios módulos redundantes para seleccionar la correcta

DISEÑOS CLÁSICOS:

(Diseño mínimo)



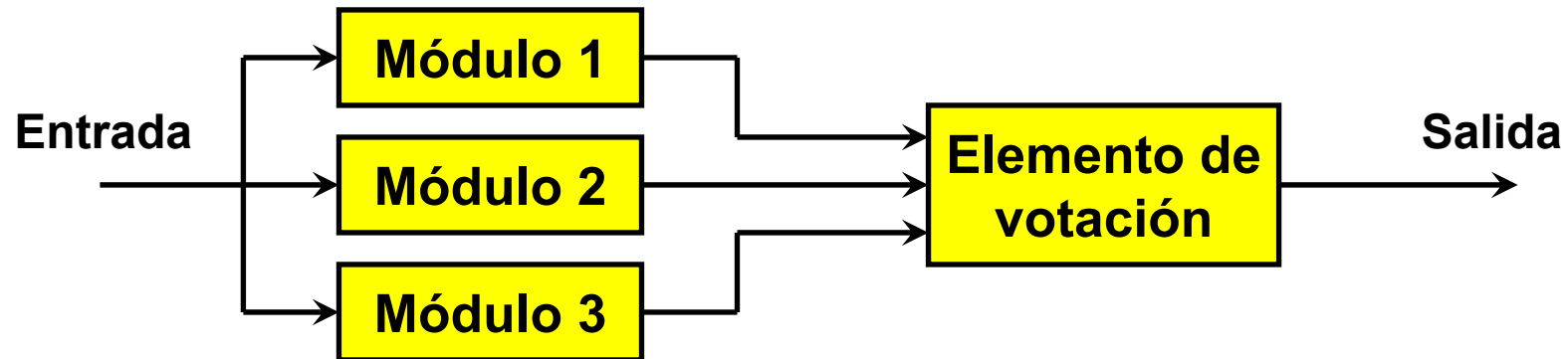
Triple Modular Redundancy (TMR)

N-Modular Redundancy

TOLERANCIA A AVERÍAS HARDWARE

Redundancia Modular Triple (TMR)

Un diseño basado en redundancia modular triple tiene esta arquitectura



El mecanismo de votación por mayoría es válido para enmascarar el fallo de un módulo, pero no protege contra el fallo simultáneo de dos módulos

ESTA TÉCNICA SE BASA EN ...

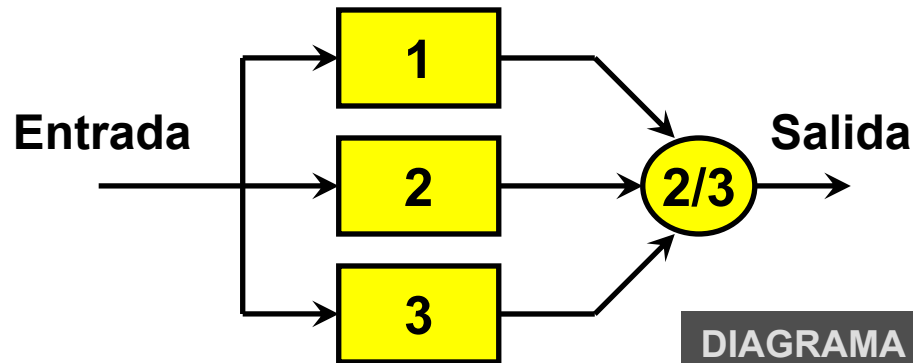
- 1) El fallo simultáneo de 2 módulos es mucho menos probable que el de 1 módulo
- 2) El elemento de votación NO debe fallar: se puede replicar
- 3) Una vez ha fallado un módulo el sistema no es tolerante a otra avería
Debe notificar la avería y se debe reparar cuanto antes

TOLERANCIA A AVERÍAS HARDWARE

Fiabilidad de la Arquitectura TMR

Suponiendo totalmente fiable el mecanismo de votación ...

La probabilidad de que el sistema funcione correctamente se expresa así:



Prob funcionamiento TMR =

= Prob [Ninguno falle]
+ Prob [Sólo falle el 1]
+ Prob [Sólo falle el 2]
+ Prob [Sólo falle el 3]

La probabilidad de que un módulo funcione durante un período t es: $R_m(t)$

$$\begin{aligned} R_{TMR}(t) = & R_1(t) \cdot R_2(t) \cdot R_3(t) \\ & + [1 - R_1(t)] \cdot R_2(t) \cdot R_3(t) \\ & + R_1(t) \cdot [1 - R_2(t)] \cdot R_3(t) \\ & + R_1(t) \cdot R_2(t) \cdot [1 - R_3(t)] \end{aligned}$$

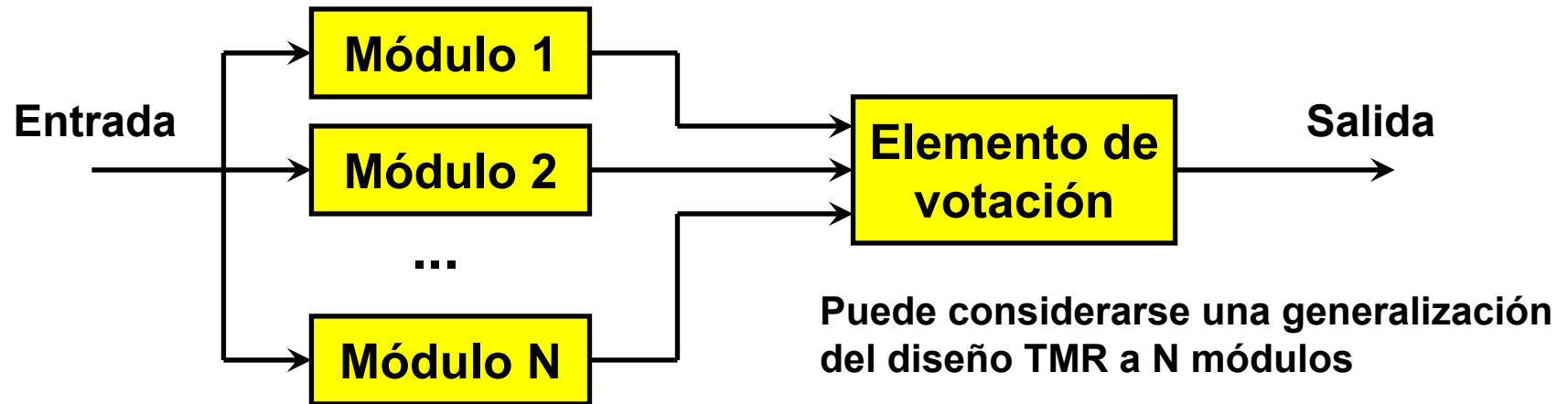
Si la fiabilidad de los 3 módulos es idéntica = $R_m(t)$

$$R_{TMR}(t) = 3 \cdot R_m(t)^2 - 2 \cdot R_m(t)^3$$

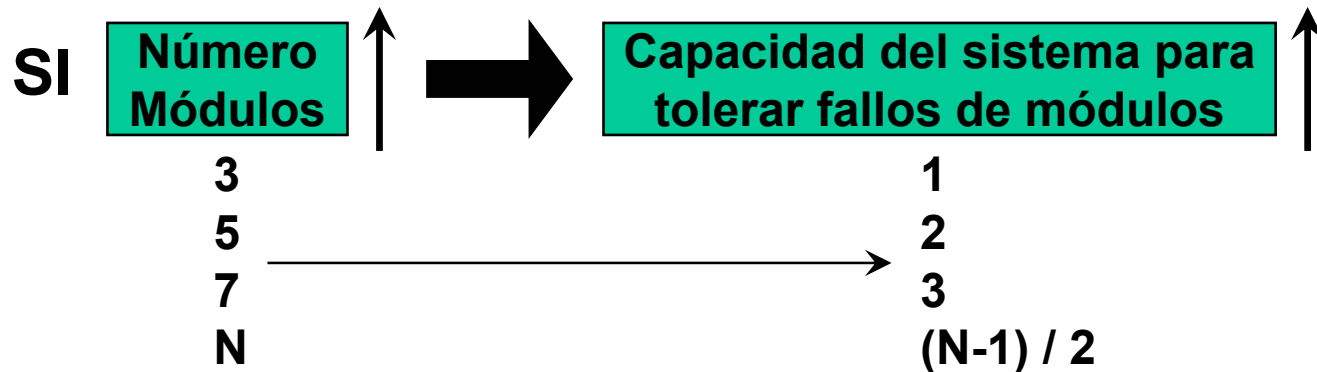
TOLERANCIA A AVERÍAS HARDWARE

Redundancia Modular Múltiple

Un diseño basado en redundancia modular múltiple tiene esta arquitectura



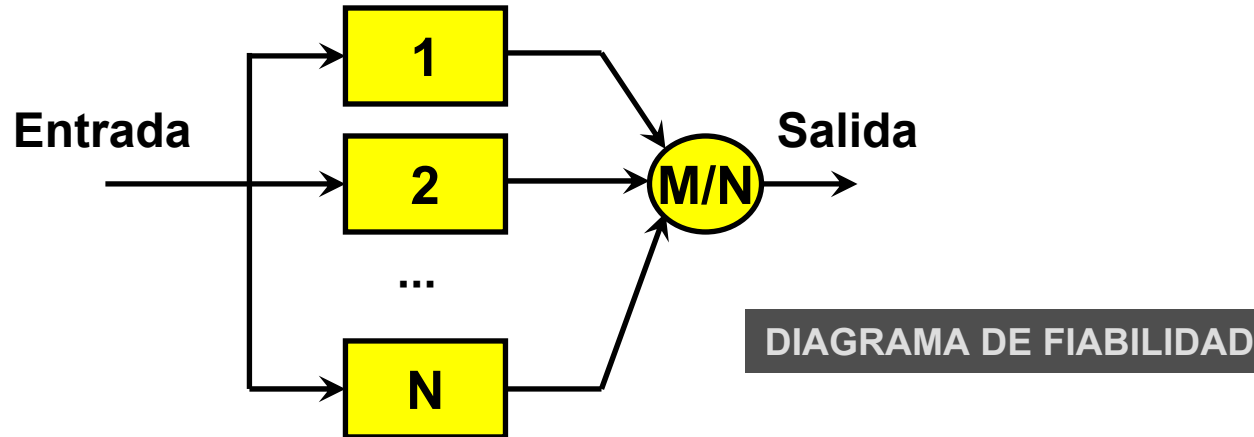
En muchos casos se usa un número impar de módulos para permitir el uso de un mecanismo de votación por mayoría



TOLERANCIA A AVERÍAS HARDWARE

Fiabilidad de la Arquitectura NMR

Suponiendo totalmente fiable el mecanismo de votación ...



La probabilidad de que un módulo funcione durante un período t es: $R_m(t)$

Si la fiabilidad de los N módulos es idéntica = $R_m(t)$

$$R_{MdeN}(t) = \sum_{i=0}^{N-M} \left(\frac{N!}{(N-i)!i!} \right) R_m^{N-i}(t) [1 - R_m(t)]^i$$

TOLERANCIA A AVERÍAS HARDWARE

Diseños Basados en Redundancia Dinámica

Basados en una combinación de dos mecanismos:

- Uno para detectar los errores generados por las averías
- Otro para sustituir on-line el módulo averiado por otro de respuesto

El enfoque dinámico requiere una redundancia menor que el estático:

Nº fallos tolerables (módulos averiados)	Nº total de módulos	
	T. Estática	T. Dinámica
1	3	2
2	5	3

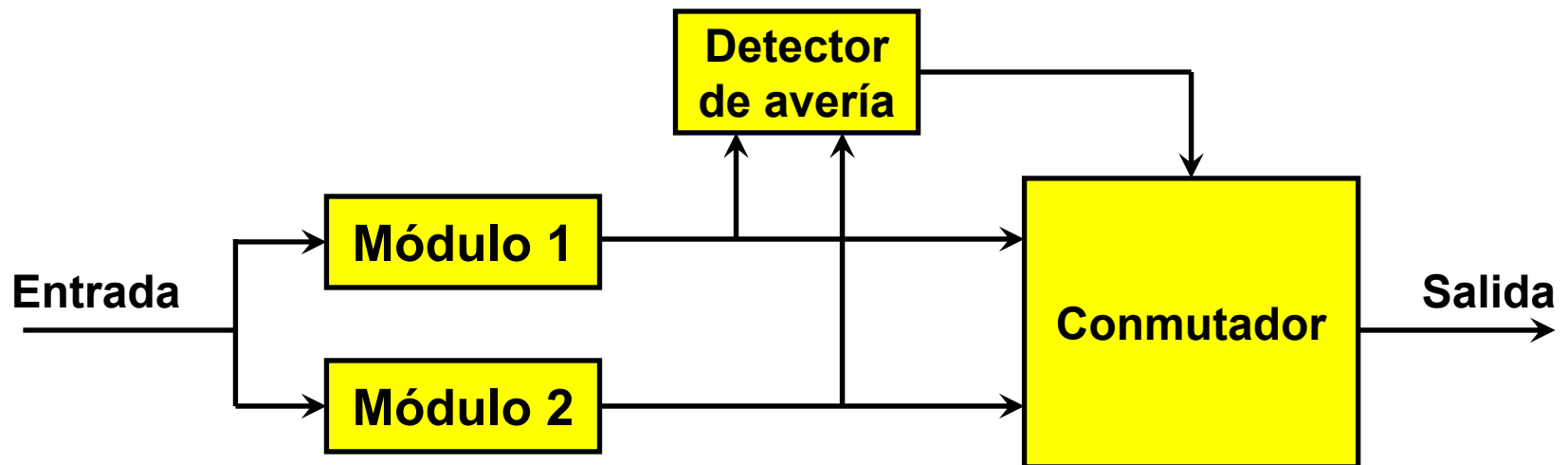
Éxito del enfoque dinámico $\xrightarrow[\text{MUCHO DE}]{\text{DEPENDE}}$ Proceso de detección de averías

- { Las técnicas dinámicas NO enmascaran las averías
- { SINO que las detectan y reconfiguran el sistema
- { Como la detección y reconfiguración puede llevar algún tiempo,
son adecuadas para sistemas que pueden soportar errores temporales

TOLERANCIA A AVERÍAS HARDWARE

Técnica Dinámica del Repuesto Preparado

Dos módulos idénticos funcionan en paralelo (el primario y el repuesto)
Cuando falla el primario el sistema conmuta al de repuesto



El esquema básico de 2 modulos se puede extender a N módulos

TOLERANCIA A AVERÍAS HARDWARE

Fiabilidad de la Técnica del Repuesto Preparado

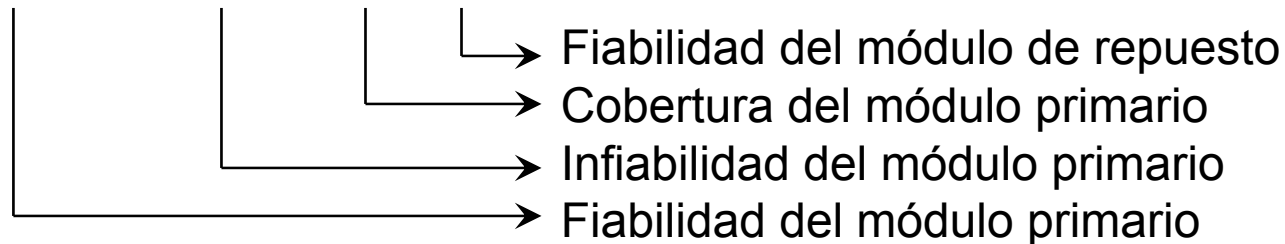
El sistema funciona bien si ...

- Funciona bien el modulo primario
- Falla el módulo primario Y

El sistema detecta el fallo
Y
El repuesto funciona bien

En términos de fiabilidad se expresa así:

$$R(t) = R_p(t) + [1-R_p(t)] C_p R_r(t)$$



Para módulos con idéntica fiabilidad:

$$R(t) = R_m(t) + [1-R_m(t)] C_m R_m(t)$$

Si la cobertura de fallos es perfecta, $C_m=1$

$$R(t) = 1 - [1-R_m(t)]^2$$

Expresión de la fiabilidad
de módulos en paralelo