Sistemas Informáticos II

Tema 3: Aspectos operacionales de los sistemas distribuidos: Disponibilidad

Antonio E. Martínez, Irene Rodríguez,
Daniel Hernández (daniel.hernandez@uam.es)
Álvaro Ortigosa (alvaro.ortigosa@uam.es)
Manuel Sánchez-Montañés (manuel.smontanes@uam.es)

Contenido

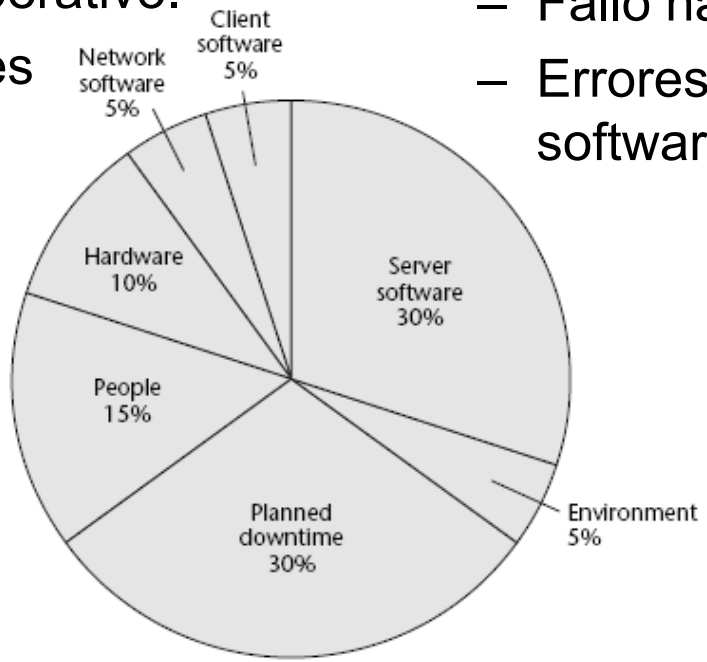
- Introducción
- Teoría de la Disponibilidad.
  - Disponibilidad, MTT**F**, MTTR y MTBF.
  - Fiabilidad.
  - Mantenibilidad.
  - Disponibilidad de asociaciones de elementos.
- Arquitecturas que incrementan la disponibilidad.
  - Comunicaciones.
  - Balanceo de carga.
  - Almacenamiento.
  - Servidores.
- Recuperación ante desastres.
- Bibliografía especial del tema

Introducción

- **La disponibilidad *(availability)* de un sistema es la probabilidad de que un sistema se encuentre operativo en un instante de tiempo arbitrario.**
- La no disponibilidad de un sistema se puede deber a dos causas:
  - Paradas no planificadas: Producidas por fallos en los equipos o los programas que implementan los servicios.
    - Requieren tratamiento estadístico: Teoría de la Fiabilidad de componentes y sistemas.
    - Los sistemas que las minimizan se llaman de Alta Disponibilidad (*High Availability, HA*).
  - Paradas planificadas: Requeridas por la aplicación para su funcionamiento: Rearranques programados, copias de seguridad, cambios de configuración...
    - Por ser previsibles permiten un tratamiento sistemático.
    - Sistemas que las minimizan se denominan de Operación Continua (*Continous Operation, CO*): Sistemas 24x7 o 24x7x365.
- El sistema ideal sería de Alta Disponibilidad + Operación Continua.
  - A mayor disponibilidad, mayor es el coste del sistema.
  - Valorar el nivel de disponibilidad requerido para invertir lo necesario para conseguirlo, y no más de eso.

Causas más frecuentes de interrupción del servicio

- Planificado
  - Copias de seguridad.
  - Reemplazar o actualizar hardware.
  - Reemplazar o actualizar aplicaciones.
  - Actualizar sistema operativo.
  - Instalación de parches
- No planificado
  - Extensión del tiempo destinado a operaciones planificadas.
  - Error humano.
  - Fallo en aplicación.
  - Fallo del sistema operativo.
  - Fallo hardware.
  - Errores de configuración del software.



Definiciones relacionadas con la disponibilidad

- **Fiabilidad *(Reliability)***: Probabilidad de que un componente o sistema continúe funcionando en un determinado instante en el tiempo.
- **Elasticidad o Resiliencia *(Resiliency)***: Capacidad de un sistema para adaptarse a condiciones externas imprevistas (fallos, aumento de carga...) para continuar cumpliendo sus parámetros de calidad.
- **Mantenibilidad *(Serviceability)*:** Es la probabilidad de realizar una reparación satisfactoria en un tiempo determinado.
- **Sistemas tolerantes a fallos *(Fault-Tolerant Systems)*:** Sistemas que contienen componentes hardware dobles de modo que el fallo de uno de ellos no suspende su operación.
- ***Clusters* de alta disponibilidad *(High Availability Clusters)*:** Conjunto de nodos de servicio que comparten conexiones externas (red, discos...) y gestionados por un software especial que permite proporcionar servicio sin interrupciones ante el fallo de alguno de sus componentes.
- ***Clusters* de alto rendimiento *(High Performance Clusters or Parallel Computing Clusters)*:** Conjunto de nodos de servicio que comparten una misma carga de trabajo.
- **Recuperación ante desastres *(Disaster Recovery)*:** Capacidad de una instalación de recuperar la operatividad tras un evento de gran magnitud, bien de tipo local (edificio), urbano (ciudad) o regional (área extendida con infraestructuras comunes).

Estimación de la disponibilidad

- La disponibilidad de un sistema se estima a partir de la medida del tiempo que ha estado operativo, T_{op}, en un intervalo de tiempo T_{tot} (suficientemente grande):

$$A = \frac{T_{op}}{T_{tot}} = \frac{T_{op}}{T_{op} + T_{inact}} \qquad \textbf{(6.1)}$$

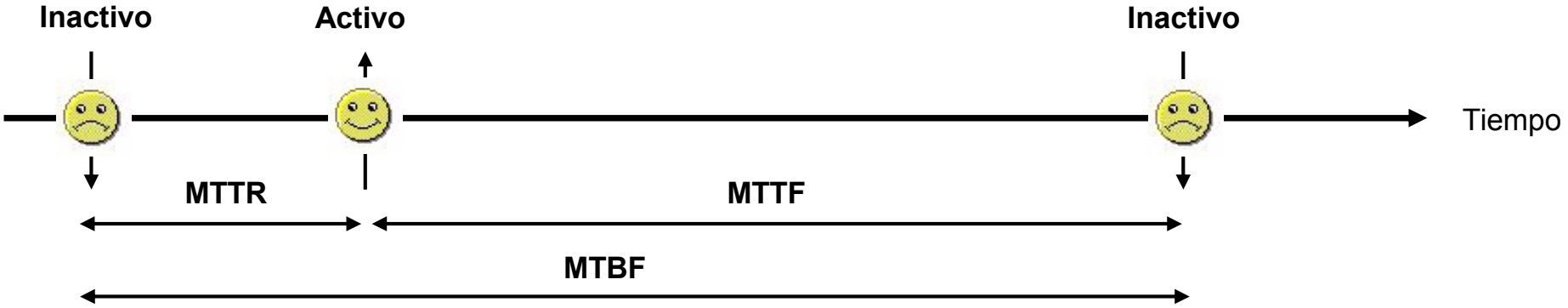
- Esta medida no da una idea global de la disponibilidad del sistema, pues un mismo valor puede obtenerse de diversas maneras:

T_{op}	T_{inact}	Escala	A
990	10	Horas	0.99
99	1	Horas	0.99
99	1	Segundos	0.99

- Para evitar esta indeterminación se utiliza el tiempo medio entre fallos.

MTBF, MTTF y MTTR

- Tiempo medio entre fallos *(Mean Time Between Failures, MTBF)* es valor esperado del tiempo que transcurre entre dos fallos consecutivos de un equipo.
- Tiempo medio hasta el fallo *(Mean Time To Failure, MTTF)* es el valor esperado del tiempo de vida de un equipo o sistema, medida de su Fiabilidad (*Reliability*).
- Tiempo medio de reparación/recuperación *(Mean Time To Repair/Restore, MTTR)* es el valor esperado del tiempo que se tarda en sustituir un equipo averiado o recuperar un fallo de software, medida de su Mantenibilidad (*Serviceability*)
- De forma gráfica, por tanto:



- En función de estos parámetros y (6.107), se puede calcular la disponibilidad como:

$$A = \frac{MTTF}{MTBF} = \frac{MTTF}{MTTF + MTTR} \qquad \textbf{(6.2)}$$

Fiabilidad

- **Fiabilidad (*reliability*)** de un componente o sistema en el tiempo, es la probabilidad de que el sistema continúe funcionando en un instante de tiempo. Si denominamos T al tiempo de vida del componente, su fiabilidad viene dada por la expresión:

$$R(t) = P\{T > t\} \qquad \textbf{(6.3)}$$

- La vida del componente, T, es una variable aleatoria, cuya función de distribución es:

$$F_T(t) = P\{T \leq t\} \qquad \textbf{(6.4)}$$

y la relación entre ambas es, por tanto:

$$R(t) = 1 - P\{T \leq t\} = 1 - F_T(t) \qquad \textbf{(6.5)} \qquad R'(t) = -F_T'(t) = -f_T(t) \qquad \textbf{(6.6)}$$

- La vida media de un componente, o tiempo medio hasta el fallo (*mean time to failure, MTTF*) es el valor esperado de T:

$$MTTF = \mathrm{E}[T] = \int_{-\infty}^{\infty} t\, f_T(t)\, dt = \int_0^{\infty} R(t)\, dt \qquad \textbf{(6.7)}$$

Función de tasa de fallo

- Suponiendo que un sistema está funcionando en un instante *t*, la probabilidad de que falle antes de un instante *x* posterior viene dada por la expresión:

$$F_T\left(x\middle|T>t\right)=P\left\{T\le x\middle|T>t\right\}=\frac{P\left\{\left\{T\le x\right\}\bigcap\left\{T>t\right\}\right\}}{P\left\{T>t\right\}}=$$

$$=\frac{P\left\{t<T\le x\right\}}{P\left\{T>t\right\}}=\frac{F_T\left(x\right)-F_T\left(t\right)}{1-F_T\left(t\right)},\ \ x\ge t \qquad \textbf{(6.8)}$$

y diferenciando con respecto a x se obtiene:

$$f_T\left(x\middle|T>t\right)=\frac{f_T\left(x\right)}{1-F_T\left(t\right)},\ \ x\ge t \qquad \textbf{(6.9)}$$

- Se define la función de tasa de fallo como la expresión anterior evaluada en *x*=*t*:

$$r\left(t\right)=f_T\left(t\middle|T>t\right)=\frac{f_T\left(t\right)}{1-F_T\left(t\right)}=\frac{-R'\left(t\right)}{R\left(t\right)} \qquad \textbf{(6.10)}$$

y se interpreta como la probabilidad de que un componente que funciona falle en el instante siguiente:

$$P\left\{t<T\le t+dt\middle|T>t\right\}=f_T\left(t\middle|T>t\right)dt=r\left(t\right)dt \qquad \textbf{(6.11)}$$

Tasa de fallos constante

- Si la tasa de fallos tiene un valor constante, λ, de (6.11) se obtiene el siguiente resultado:

$$r(t)=\lambda \Rightarrow \frac{-R'(t)}{R(t)}=\lambda \Rightarrow \int_0^t \frac{-R'(t)}{R(t)}dt=\int_0^t \lambda dt$$

$$\lambda t=-\ln R(t) \Rightarrow R(t)=e^{-\lambda t},\quad t>0 \qquad \textbf{(6.12)}$$

- Por tanto, para el tiempo de vida T:

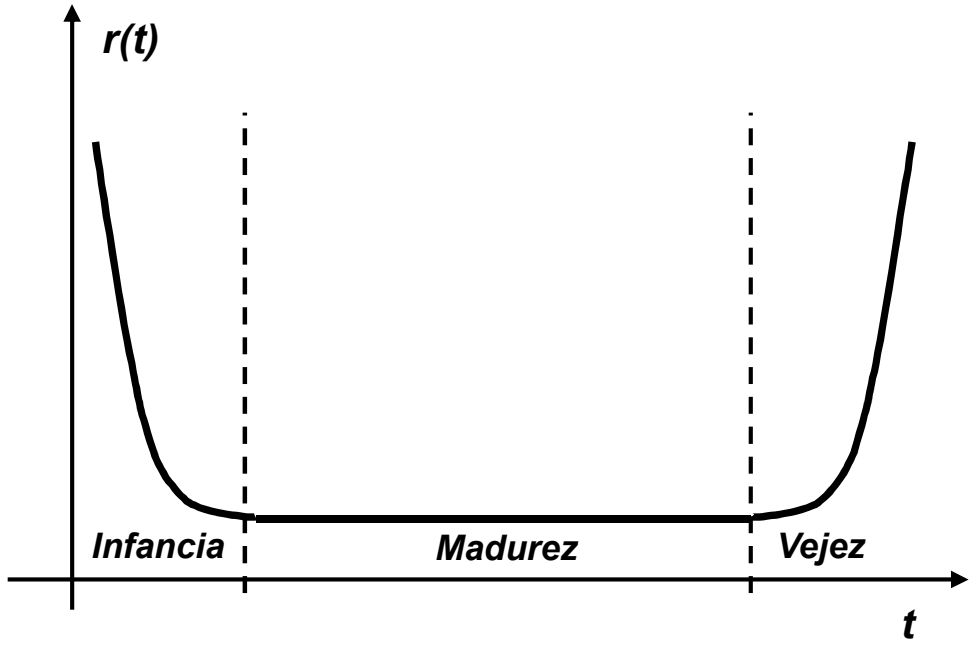
$$f_T(t)=-R'(t)=\lambda e^{-\lambda t} \qquad \textbf{(6.13)} \qquad F_T(t)=1-R(t)=1-e^{-\lambda t} \qquad \textbf{(6.14)}$$

- **El tiempo de vida T es, por tanto, una variable aleatoria con distribución exponencial.** Su valor esperado, el MTTF, será:

$$MTTF=\mathrm{E}[T]=\int_0^\infty R(t)dt=\int_0^\infty e^{-\lambda t}dt=\frac{1}{\lambda} \qquad \textbf{(6.15)}$$

Fallos en equipos físicos

- La función de tasa de fallos en un equipo tiene la forma que se muestra en la figura. Se distinguen tres zonas.
  - Zona de *infancia*: Alta tasa de fallos debido a montaje de componentes defectuosos. Va reduciéndose con el tiempo.
  - Zona de *madurez*: Tasa de fallos constante
  - Zona de *vejez*: El envejecimiento del equipo aumenta la tasa de fallos.
- **Los cálculos se realizan para el periodo de madurez del equipo, y se emplean los resultados obtenidos para *r(t)* constante**.



Fallos en programas (I)

- Se encuentran dos situaciones distintas:
- **Programas que no son reparados cuando se encuentra un defecto.**
  - Es la situación habitual en producción y con programas cerrados.
  - Resolución de problemas se produce en una nueva versión o entrega (*release*) del mismo
  - El modelo de tasa de fallos constante es válido durante la explotación de una versión.
- **Programas cuyos defectos se corrigen conforme se encuentran**.
  - Programas de producción propia o ciclos de pruebas en la producción de programas.
  - Los defectos se van encontrando en el transcurso de la vida del programa, y resolviendo.
  - La corrección de defectos introduce otros nuevos.

Fallos en programas (II)

- El modelo de tasa de fallos constante no es válido.
  - La tasa de fallos será una función decreciente con el tiempo (con funcionalidad constante).
  - Su forma depende del modelo elegido para el **ritmo de descubrimiento de fallos**:
    - **Modelo básico (ritmo lineal):**

$$\lambda(\mu) = \lambda_0\left(1 - \mu/\nu_0\right),\ \ \lambda(t) = \lambda_0 e^{-\frac{\lambda_0}{\nu_0}t} \qquad \textbf{(6.16)}$$

    - **Modelo de Poisson logarítmico (ritmo exponencial):**

$$\lambda(\mu) = \lambda_0 e^{-\theta\mu},\ \ \lambda(t) = \lambda_0/\left(\lambda_0\theta t + 1\right) \qquad \textbf{(6.17)}$$

λ_0: Tasa de fallos inicial

ν_0: Total de defectos

θ: Ritmo de relajación intensidad fallos

μ: Número de fallos experimentados

Mantenibilidad

- Es la probabilidad de realizar una reparación satisfactoria en un tiempo determinado.
  - Mide la rapidez y facilidad con la que un sistema se vuelve a poner operativo tras ocurrir un fallo.
  - Si denominamos T' al tiempo de reparación de un equipo:

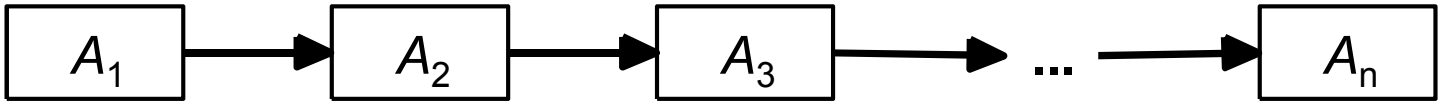
$$M(t) = P\{T' \leq t\} \qquad \textbf{(6.18)}$$

- El tiempo de reparación incluye:
  - El tiempo necesario para descubrir que ha habido un fallo.
  - El tiempo de diagnóstico de la causa del fallo.
  - El tiempo de conseguir las piezas necesarias para realizar la reparación.
  - El tiempo para lograr acceder a las piezas a sustituir.
  - El tiempo de reemplazo de dichas piezas.
  - El tiempo en poner el sistema operativo de nuevo.
  - El tiempo de verificación de que el sistema funciona conforme a sus especificaciones.
  - El tiempo de "cierre" del sistema, y vuelta a operación normal.
- El **tiempo medio de reparación (Mean Time To Repair, MTTR)** viene dado por la expresión:

$$MTTR = \mathrm{E}[T'] \qquad \textbf{(6.19)}$$

Componentes en serie

- El sistema está compuesto por una serie de componentes.
- Si los componentes están en *serie*, un fallo en cualquiera de ellos hace que falle el sistema global.
  - Puede no representar una conexión física en serie, sino una dependencia entre diversos equipos asociados (ordenador cliente, tarjeta de red, programa cliente).



- Suponiendo que el fallo en cada uno de los sistemas es **independiente** del fallo en cualquier otro, la **disponibilidad total** del sistema será:

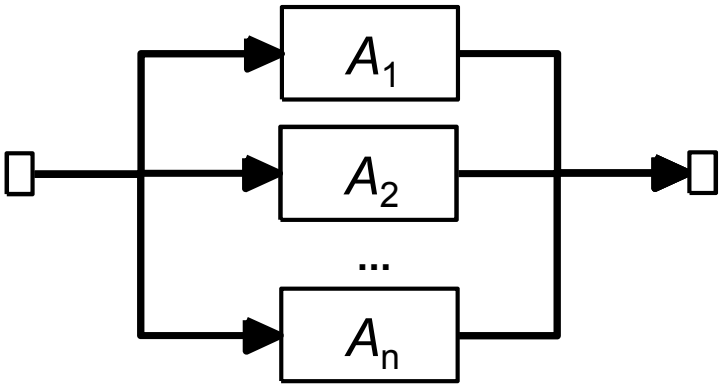
$$A = \prod_{i=1}^{n} A_i \qquad \textbf{(6.20)}$$

- Del mismo modo, para la **fiabilidad del conjunto** se verifica que:

$$R(t) = \prod_{i=1}^{n} R_i(t) \qquad \textbf{(6.21)} \qquad r(t) = \sum_{i=1}^{n} r_i(t) \qquad \textbf{(6.22)}$$

Componentes en paralelo o redundantes

- Un sistema con n componentes redundantes depende para su funcionamiento correcto del funcionamiento de uno sólo de dichos componentes.
  - Representa una conexión en paralelo de sistemas para realizar dicha función.



- La **disponibilidad** de un sistema redundante, suponiendo la **independencia** en las disponibilidades de cada uno de los componentes, vendrá dada por la expresión:

$$A = 1 - \prod_{i=1}^{n} \left(1 - A_i\right) \qquad \textbf{(6.23)}$$

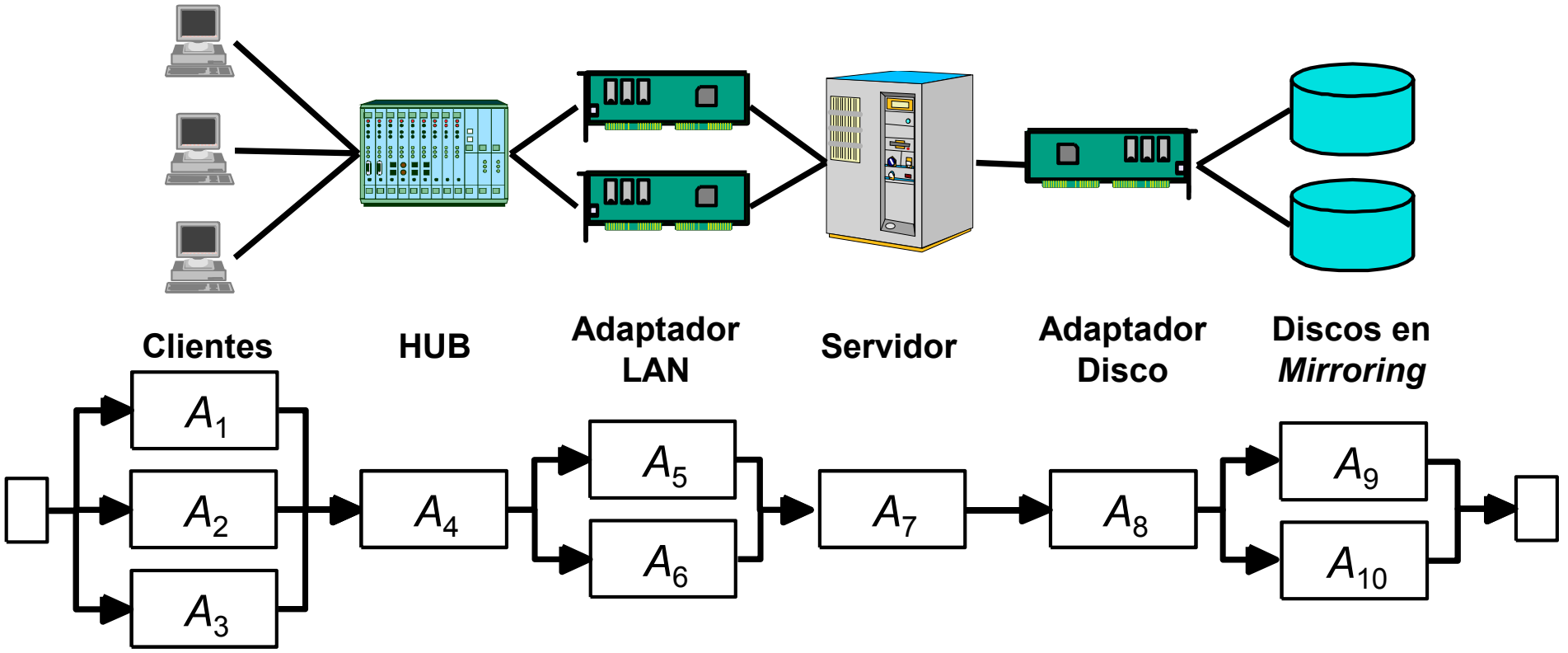
- Del mismo modo, para la **fiabilidad del conjunto** se verifica que:

$$F(t) = \prod_{i=1}^{n} F_{Ti}(t) \qquad \textbf{(6.24)}$$

$$R(t) = 1 - \prod_{i=1}^{n} \left(1 - R_i(t)\right) \qquad \textbf{(6.25)}$$

Diagramas de bloques de disponibilidad

- Representación gráfica de las dependencias de disponibilidad del sistema en forma de una cadena de procesamiento. Ejemplo:



$$A = \left[1-(1-A_1)(1-A_2)(1-A_3)\right]A_4\left[1-(1-A_5)(1-A_6)\right]A_7A_8\left[1-(1-A_9)(1-A_{10})\right]$$

Mejoras de la disponibilidad

- Dividiendo numerador y denominador de (6.2) por MTTF se obtiene la expresión equivalente:

$$A = \frac{1}{1 + MTTR/MTTF} \qquad \textbf{(6.26)}$$

- De esa expresión se puede concluir la forma de mejorar la disponibilidad de un sistema:
  - **Aumentando el MTTF:**
    - Mejorando la calidad de los equipos.
    - Introduciendo elementos **redundantes** en todos los puntos de la cadena de procesamiento.
      - Eliminación de **Puntos Simples de Fallo *(Single Points Of Failure, SPOF)*.**
  - **Disminuyendo el MTTR**: En cualquiera de las siguientes fases:
    - Tiempo de latencia ante el fallo (desde que se produce hasta que se descubre).
    - Tiempo para aislarla (desde que se descubre hasta que se conoce el motivo).
      En ambos casos, estos tiempos se pueden reducir mediante:
      - Empleo de sistemas de gestión proactiva.
      - Servicio adecuado de vigilancia y mantenimiento *(HelpDesk)*.
    - Tiempo para corregirla.
      - Contratos de mantenimiento.
      - Redundancia de equipos críticos.
    - Tiempo para verificar de nuevo el correcto funcionamiento del sistema.
      - Herramientas de prueba automáticas.

Arquitecturas que incrementan la disponibilidad

- **La disponibilidad de la cadena de procesamiento es siempre menor que la disponibilidad del menor de sus elementos.**
- Los puntos más críticos de la cadena de procesamiento son aquellos en los que en caso de producirse un fallo se produce la caída del servicio.
  - Ese punto se denomina ***Single Point Of Failure, SPOF***.
- Las arquitecturas orientadas a proporcionar la mayor disponibilidad posible del sistema se basan en la utilización de elementos redundantes (paralelo) en todas las partes de la cadena de procesamiento, denominados ***clusters***.
  - La utilización de *clusters* es la solución que más incrementa el MTTF de cada componente de la cadena de procesamiento.
  - Se eliminan los SPOF.
- Conseguir que varios sistemas realicen en paralelo la misma función no es sencillo, siendo necesarias **soluciones especiales** para conseguirlo.
- La naturaleza de las soluciones varía en función de diversos factores:
  - Las necesidades de disponibilidad objetivo que se persigan.
  - El tipo de elemento al que se necesite proporcionar la redundancia.
- En todos los sistemas es necesario considerar dos procesos:
  - ***Fail-over***: cómo actuar cuando se produce un fallo para mantener el servicio.
  - ***Fail-back*****:** cómo actuar para recuperar la situación normal cuando el fallo se resuelve.

Tipos de redundancia

- Según el **estado** de cada elemento del *cluster*:
  - **Activo-Activo (AA):**
    - Todos los componentes del *cluster* se encuentran activos, es decir, realizando tareas de proceso significativas para el proceso final del sistema.
  - **Activo-*Stand by* (AS):**
    - Un componente del cluster se encuentra activo.
    - El resto se encuentran disponibles para su activación casi instantánea.
  - **Activo-Pasivo (AP):**
    - Un componente del cluster se encuentra activo.
    - El resto se pueden activar bajo demanda después de un intervalo determinado de tiempo.
- Según el **reparto de carga** entre los elementos del cluster:
  - **Reparto dinámico de la carga (D)**, según las necesidades de proceso de cada momento.
    - El fallo de uno de los elementos del sistema no requiere reconfiguración.
  - **Reparto estático de la carga (E).**
    - El fallo de uno de los elementos requiere reconfiguración del sistema, manual o automática.
- Según el **tratamiento de las conexiones activas**:
  - **Continuidad** de las sesiones activas tras un fallo (**C**).
  - **Interrupción** de las sesiones activas tras un fallo (**I**).

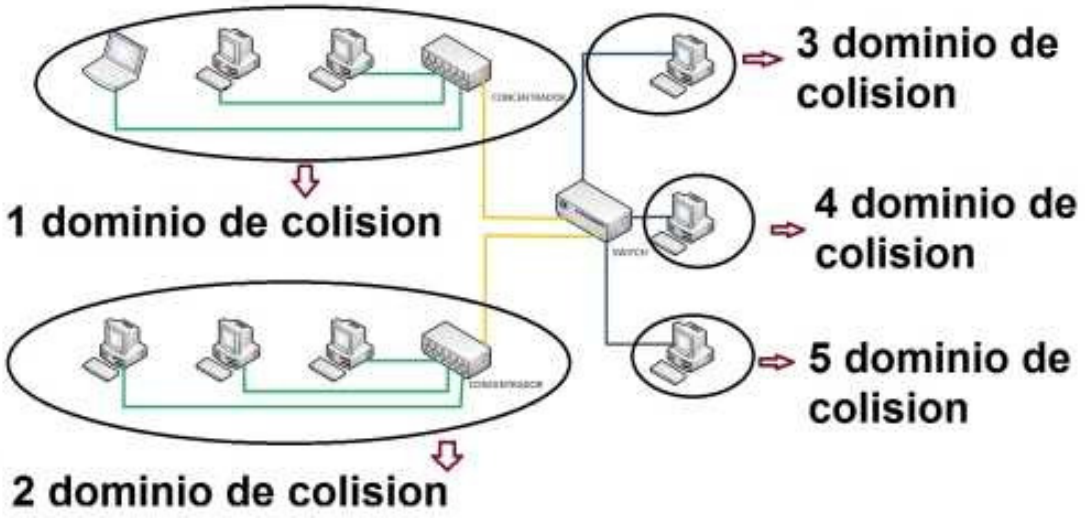
Redundancia en los sistemas de comunicaciones

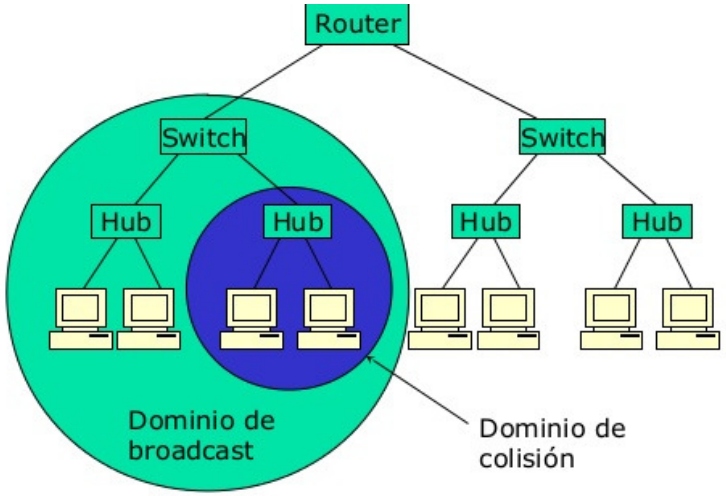
- **Los sistemas de comunicaciones son uno de los puntos críticos de todo sistema distribuido.**

- Dentro de las comunicaciones es necesario estudiar
  - **Redundancia en las redes de área local (*Local Area Networks, LAN*).**
    - Enlaces entre elementos.
    - Equipos que forman el núcleo de las LAN: *hubs*, *switches*...
  - **Redundancia en las redes de área extendida *(Wide Area Networks, WAN)*.**
    - Redundancia de las líneas de comunicaciones.
    - Redundancia de los equipos de comunicaciones: *routers.*

- En los casos que se presentan se considerarán únicamente los **estándares *de facto*** en sistemas de comunicaciones actuales:
  - **Redes LAN basadas en Ethernet**, en sus diversas variaciones.
  - **Protocolo TCP/IP** como medio de transporte.

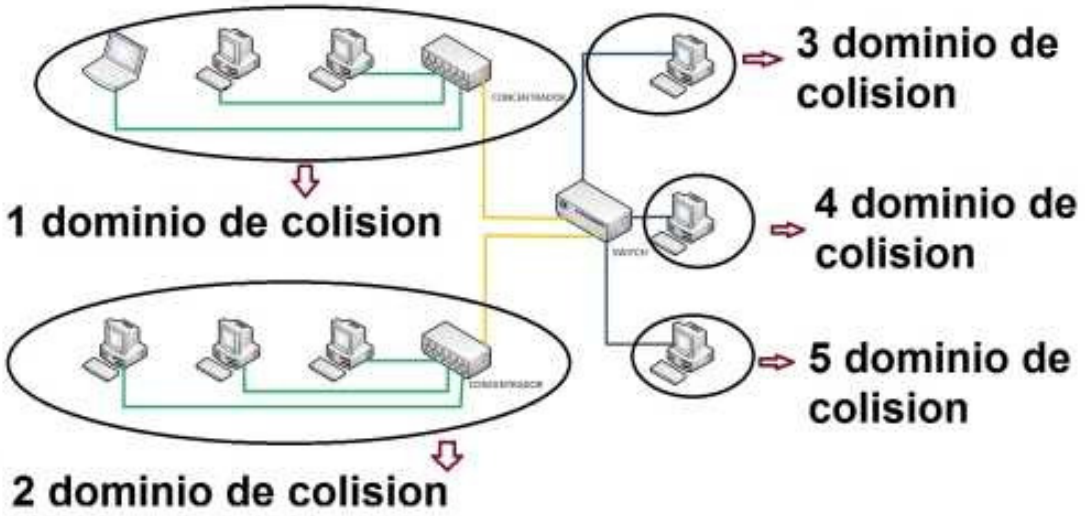
Dominios de tráfico

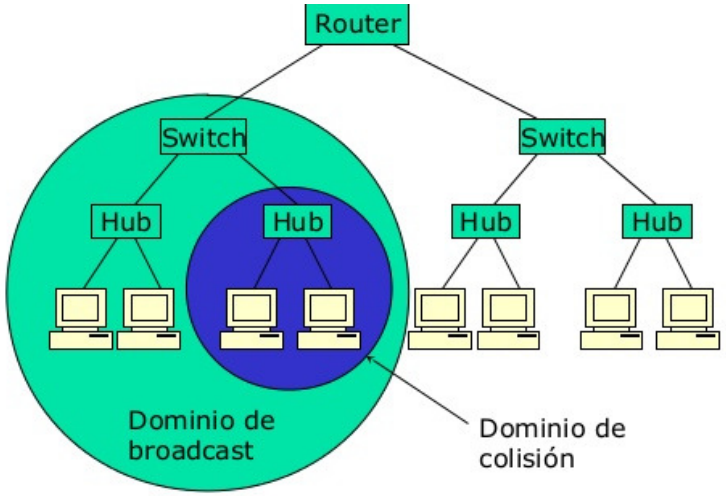




- Un **dominio de colisión** es un segmento físico de una red de computadores donde es posible que las tramas puedan "colisionar" (interferir) con otros. Estas colisiones se dan particularmente en el protocolo de red Ethernet.
- A medida que aumenta el número de nodos que pueden transmitir en un segmento de red, aumentan las posibilidades de que dos de ellos transmitan a la vez → ***colisión*.**
- Conforme aumenta el número de colisiones **disminuye el rendimiento y disponibilidad** de la red.

Dominios de tráfico

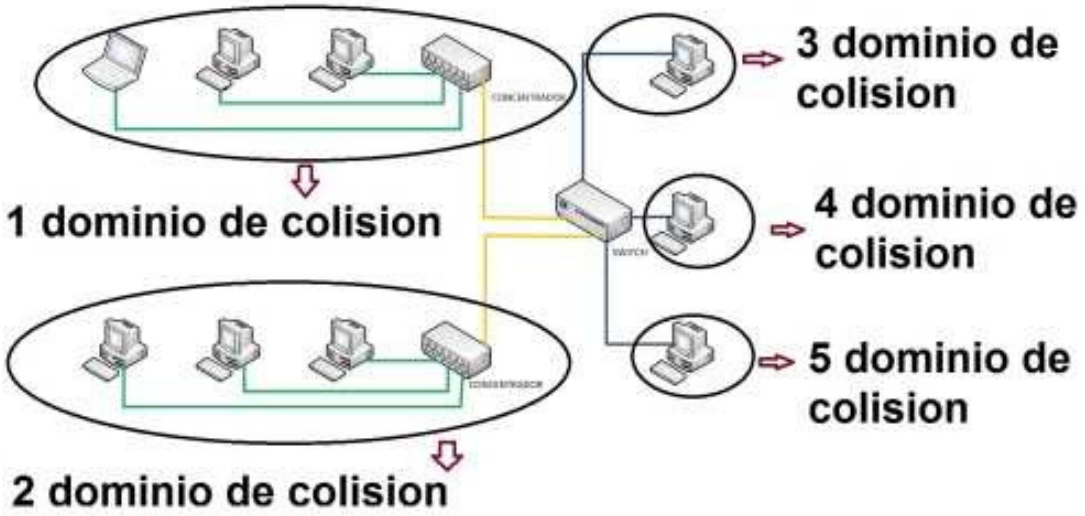


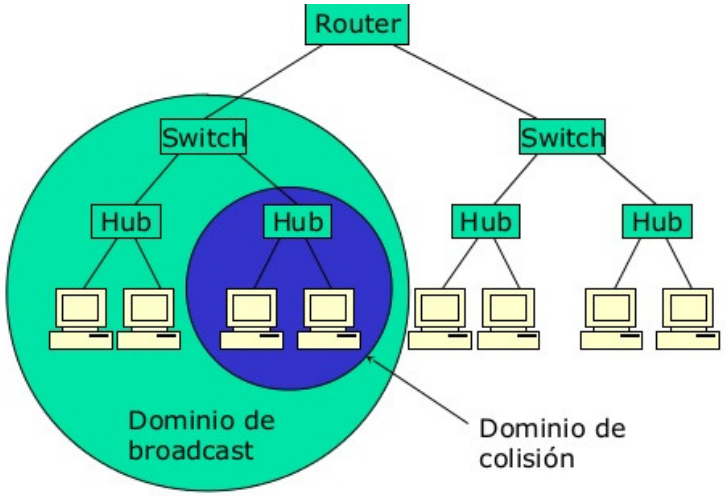


- Un **dominio de difusión** (***broadcast domain***) es el área lógica en una red de computadoras en la que cualquier computadora conectada a la red puede transmitir directamente a cualquier otra computadora en el dominio sin precisar ningún dispositivo de encaminamiento, dado que comparten la misma subred, dirección de puerta de enlace y están en la misma red de área local (**LAN**) virtual o **VLAN** (predeterminada o instalada).

- Es un área de una red de computadoras, formada por todos las computadoras y dispositivos de red que se pueden alcanzar enviando una trama a la **dirección de difusión** de la capa de enlace de datos.

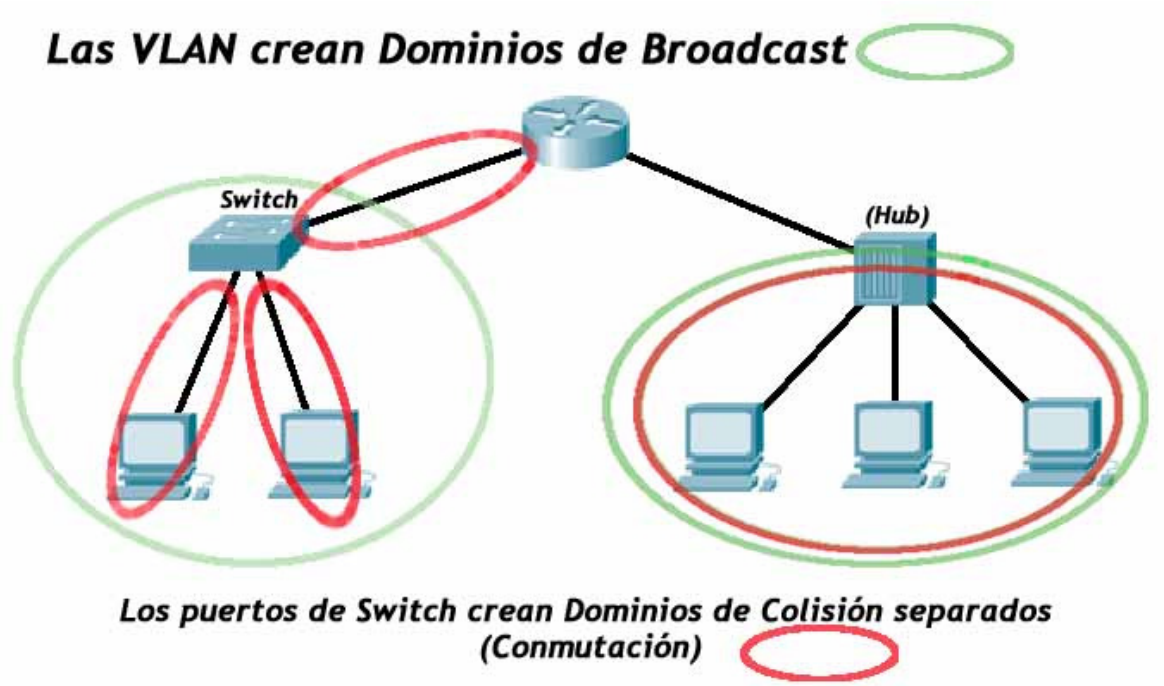
Dominios de tráfico





- Los dispositivos de la capa 1 OSI (como los **hubs**) **extienden** los dominios de colisión (1 y 2 en la figura).
- Los dispositivos de la capa 2 y 3 OSI (como los **switch/conmutadores**) **segmentan** los dominios de colisión (3-5 en la figura).
- Los dispositivos de la capa 3 OSI (como los **routers**) segmentan los dominios de colisión y difusión (broadcast).

Dominios de colisión y de broadcast



- Una serie de *switches* interconectados forman un **dominio de *broadcast* simple** → Cuando se conectan dos *switches*, aumenta el dominio de *broadcast*.

- Esto provoca una disminución en la eficacia (ineficiencia) de la red dado que el ancho de banda se utiliza para propagar el tráfico de *broadcast* y aumenta la probabilidad de colisión.

- Para reducir dominio broadcast → **VLAN**.

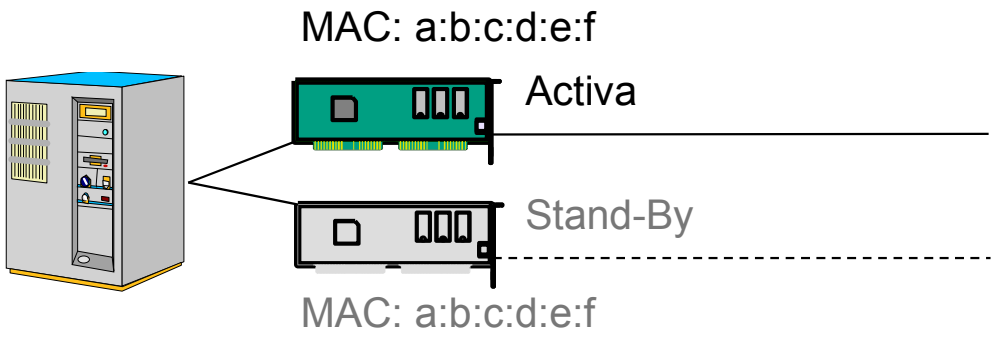
Arquitectura de las LAN en un Centro de Proceso de Datos (CPD)

- Las características de los equipos que actualmente se emplean para implementar una LAN hacen que estas redes se puedan construir atendiendo a modelos distintos:
  - **Nivel 2 compartido**: Múltiples servidores se conectan al mismo segmento de LAN implementado en un ***Hub***.
  - **Nivel 2 conmutado**: Cada enlace es un segmento de LAN. Todos los segmentos se interconectan entre sí mediante conmutadores *(**switches**)*, que actúan como puentes multipuerta.
  - **Nivel 3**: Cada enlace es un segmento de red que une un elemento con un ***router***. El *router* se implementa también en los propios *switches* mediante módulos especiales.
- **Estos modelos suelen coexistir** dentro de un Centro de Proceso de Datos (CPD), empleándose cada uno en distintos ámbitos:
  - Nivel 2 compartido: menos eficiente y flexible, y por ello menos utilizado para la conexión de servidores con el abaratamiento de las otras soluciones.
  - Nivel 2 conmutado: **nivel de acceso**. Conexiones de los ordenadores de las granjas de servidores.
    - Organizados en múltiples **LAN físicas** (redes de acceso con *switches* independientes) o **lógicas** *(Virtual Local Area Networks, VLAN)*.
  - Nivel 3: **nivel de agregación**. Interconexión de los diversos niveles de acceso existentes, y servidores especiales de alto tráfico, y de ellos con las redes externas.

Redundancia en la conexión de servidores a nivel 2 (dual-homed servers)

AS | E | C

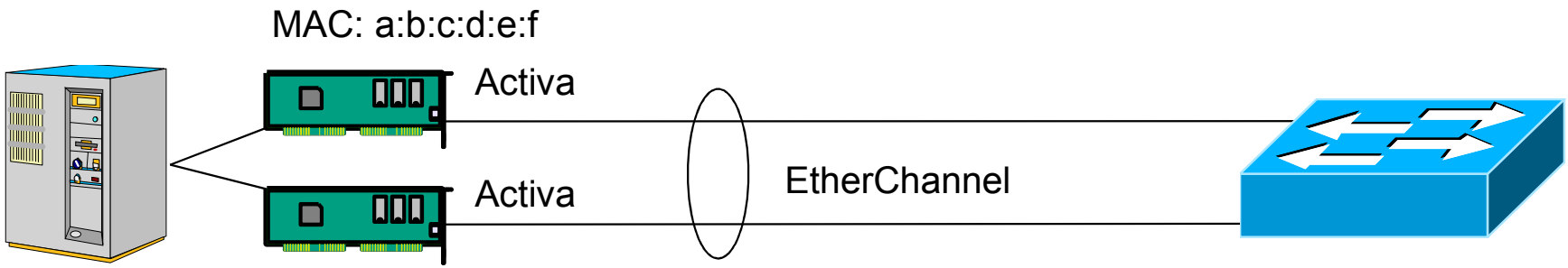
- La redundancia básica en la conexión de un ordenador a una LAN se consigue mediante el uso de **dos o más adaptadores de red.**
  - Aunque cada adaptador puede tener más de un puerto, es recomendable utilizar un puerto en cada adaptador para lograr un nivel mayor de redundancia.
- La gestión del adaptador en el sistema (a través de su driver) hace que únicamente se encuentre activo uno de ellos → **Activo-Standby.**
  - Ambos puertos comparten la misma dirección MAC.
  - En caso de fallo de uno de ellos, el sistema conmuta al adaptador de reserva.
    - El tiempo habitual de conmutación es menor de 1 s.
  - No hay cambios para los protocolos de niveles superiores.
- Cada una de las conexiones puede ir a un *switch* de acceso distinto.
  - Requiere que la misma VLAN esté disponible en ambos *switches*.



Agrupamiento de múltiples enlaces en EtherChannels

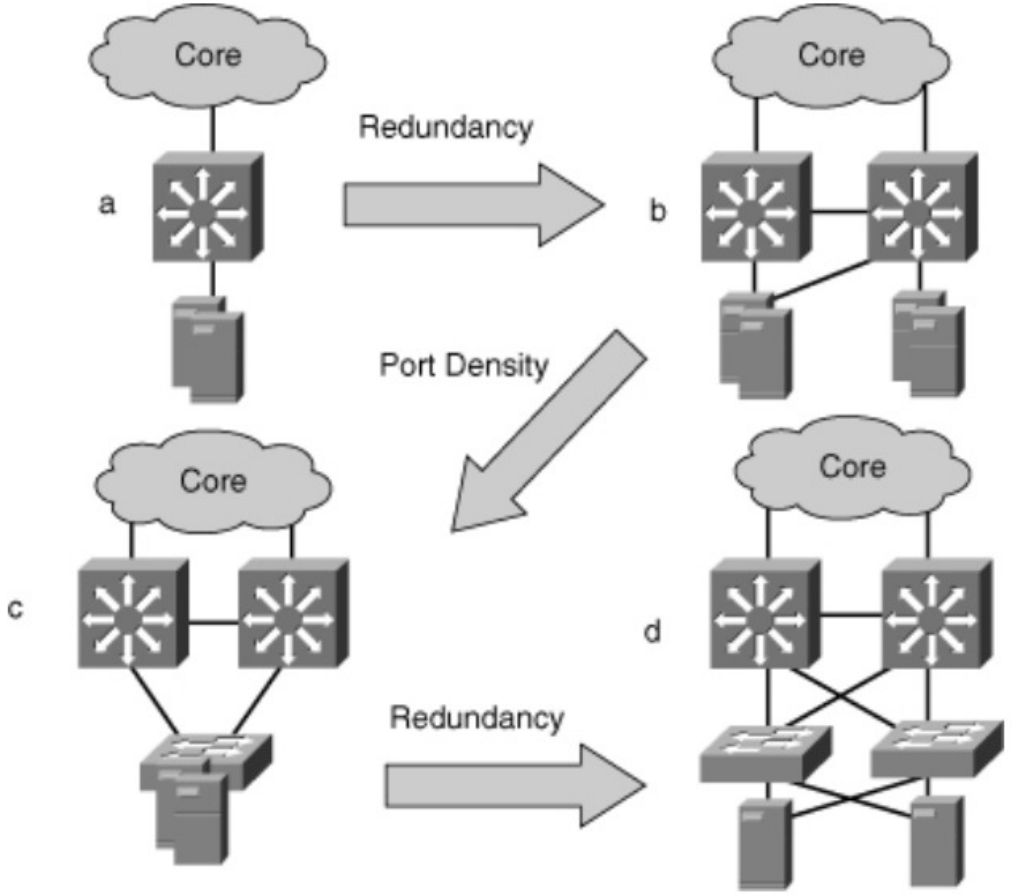
AA | D | C

- EtherChannels son grupos de enlaces Ethernet que se comportan como un único enlace.
  - Todos los puertos de un EtherChannel **comparten la misma dirección MAC**.
  - **Todos los puertos se encuentran activos** simultáneamente.
  - Los protocolos de niveles superiores lo ven como un único enlace de nivel 2.
- **Ventajas**: mayor rapidez en la resolución de fallos **Y** aumento el ancho de banda de las conexiones. **Inconveniente**: mayor complejidad en la configuración y mayor coste.
- Para su implantación debe existir soporte de EtherChannel en ambos extremos del enlace de nivel 2:
  - *Driver* de red, si el extremo de la conexión es un servidor.
  - Puertos del elemento de red que recibe la conexión (típicamente un *switch*).
- **El EtherChannel se comporta como una conexión única entre dos elementos**. Sus componentes no se pueden disgregar para conectar a más de un elemento destino.

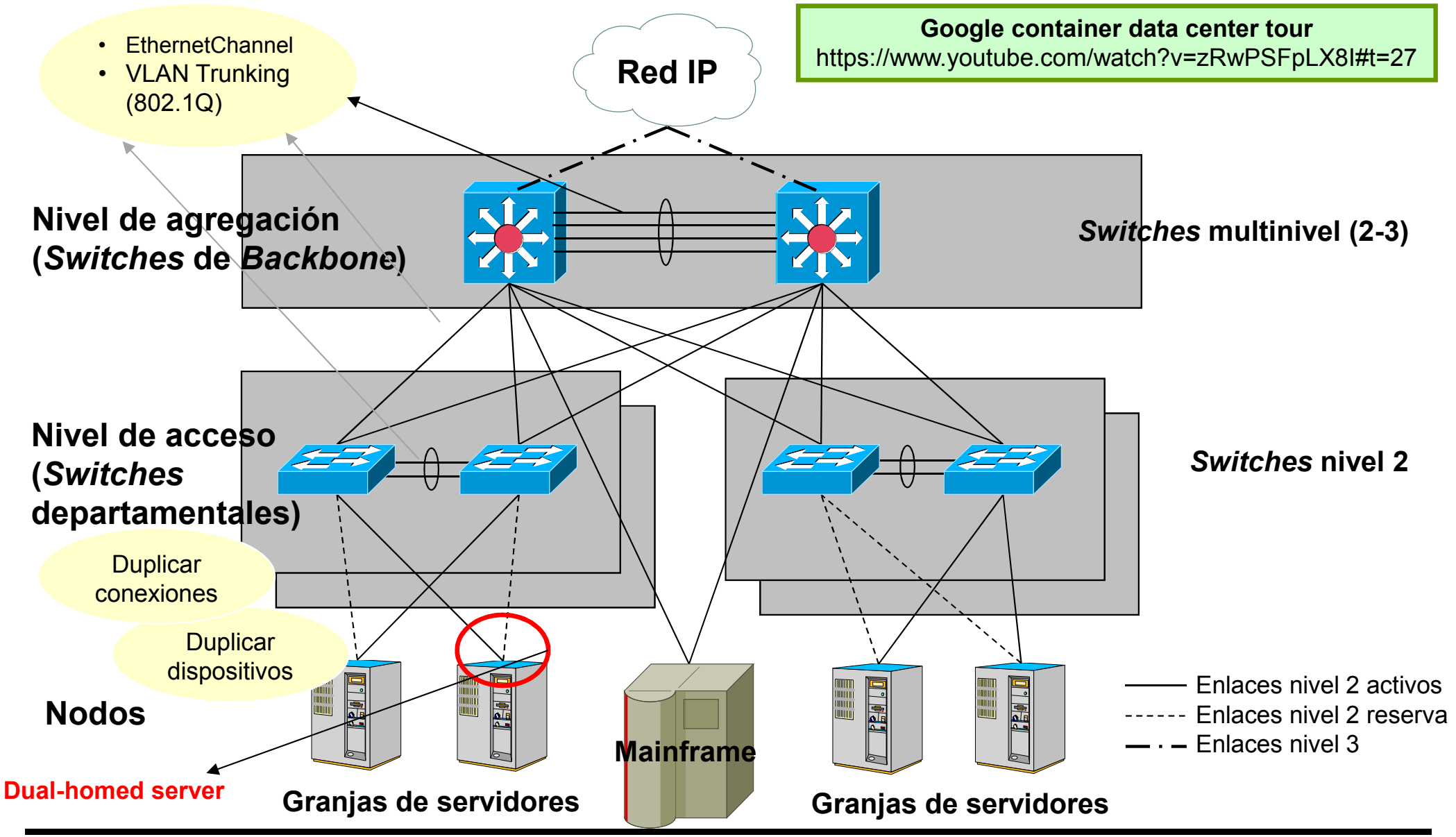


Ejemplo: topología totalmente redundante (sin SPOF)

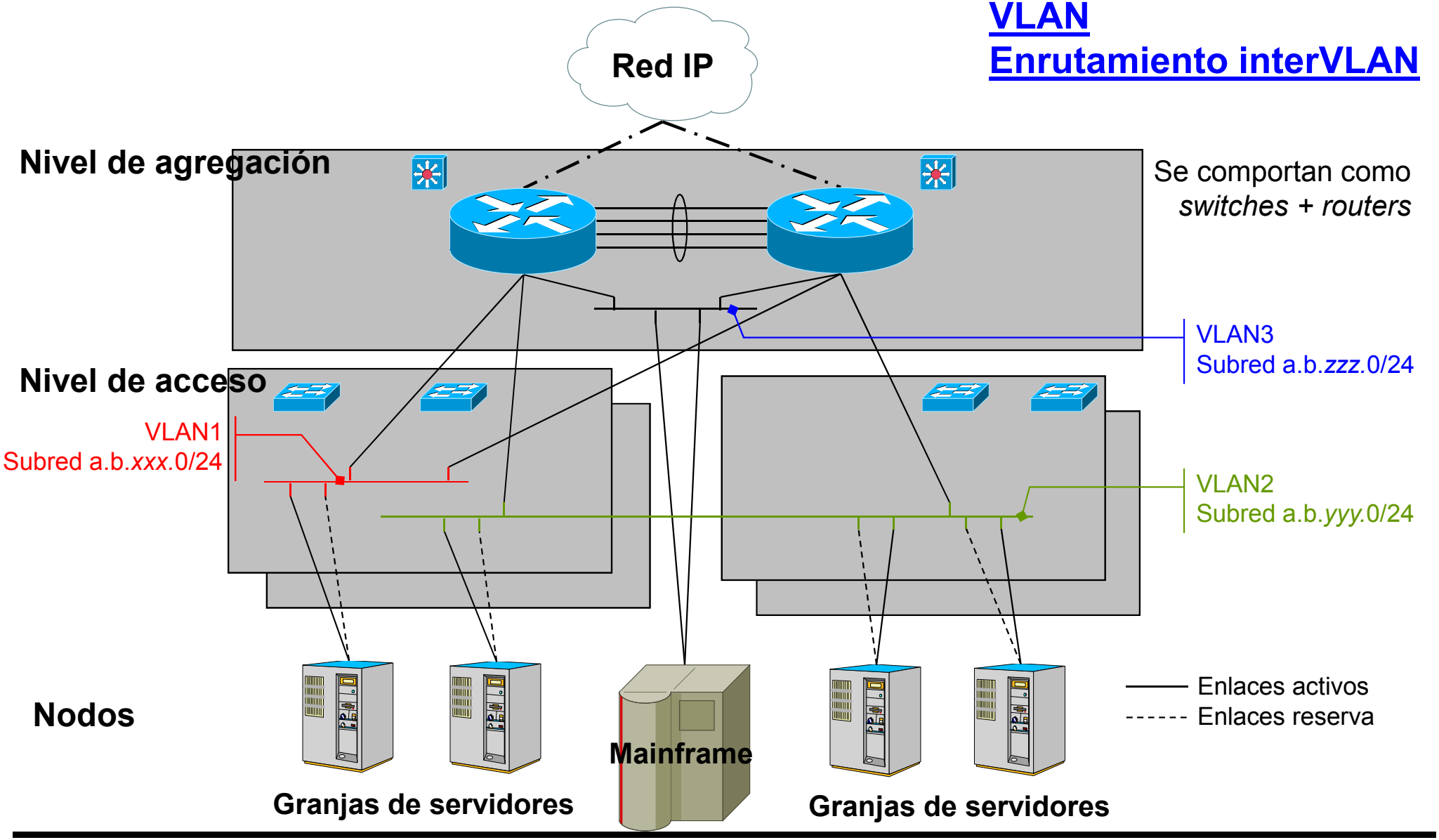
Figure 4-12. Multilayer Redundant Design



Topología de LAN redundante para un CPD. Ejemplo



Topología de LAN redundante para un CPD
Ejemplo de estructura lógica



Implicaciones de la redundancia en las LAN

- La instalación de elementos redundantes a nivel 3 implica, para su correcto funcionamiento, el uso de un protocolo de encaminamiento dinámico (OSPF, BGP), igual que en el caso de redes de área extendida *(WAN)*.
- La instalación de elementos redundantes a nivel 2 implica la aparición de bucles en los posibles caminos entre dos nodos de la red.
  - Incompatible con el protocolo de ***Transparent Bridging*** empleado en redes Ethernet.
  - Se resuelve con caminos únicos mediante el ***Spanning Tree Protocol, STP*** (802.1D).
    - En caso de fallo de un enlace, al reconfigurarse el *spanning tree* se utilizará un enlace alternativo para comunicar con los nodos que dependan de él.
  - El proceso de reconfiguración del STP es lento, por lo que se ha diseñado el ***Rapid Spanning Tree Protocol, RSTP*** (802.1w).
  - Adicionalmente, la optimización del uso de VLANs requiere la creación de múltiples *spanning trees*, cada uno adecuado a la topología de su VLAN. Esta solución la proporciona el ***Multiple Spanning Tree, MST*** (802.1s)
- La redundancia en los enlaces de conexión se puede proporcionar mediante:
    - Para servidores, mediante múltiples enlaces gestionados a nivel 2.
    - Para servidores y switches, con el agrupamiento de enlaces en *Etherchannels*.

Rapid Spanning Tree Protocol, RSTP

AS | E | C

- Cada *switch* tiene asignado un **identificador**, que contiene su **prioridad**.
  - El *switch* con mayor prioridad se denomina *switch* raíz *(root)*.
  - El siguiente en prioridad, *switch* raíz alternativo *(alternate root)*.

- Los puertos de cada *switch* pueden desempeñar tres papeles:
  - **Puerto raíz *(root port, RP)*:** el que da el mejor camino (menor coste) al *switch* raíz. Habrá uno por switch (excepto root switch).
  - **Puerto designado *(designated port, DP)*:** el que proporciona a cada segmento de red conectado al *switch* el mejor camino al *switch* raíz.
  - **Puerto alternativo *(alternate port, AP)*:** Puerto que no es ni RP ni DP. No transmite tramas → se rompen bucles.

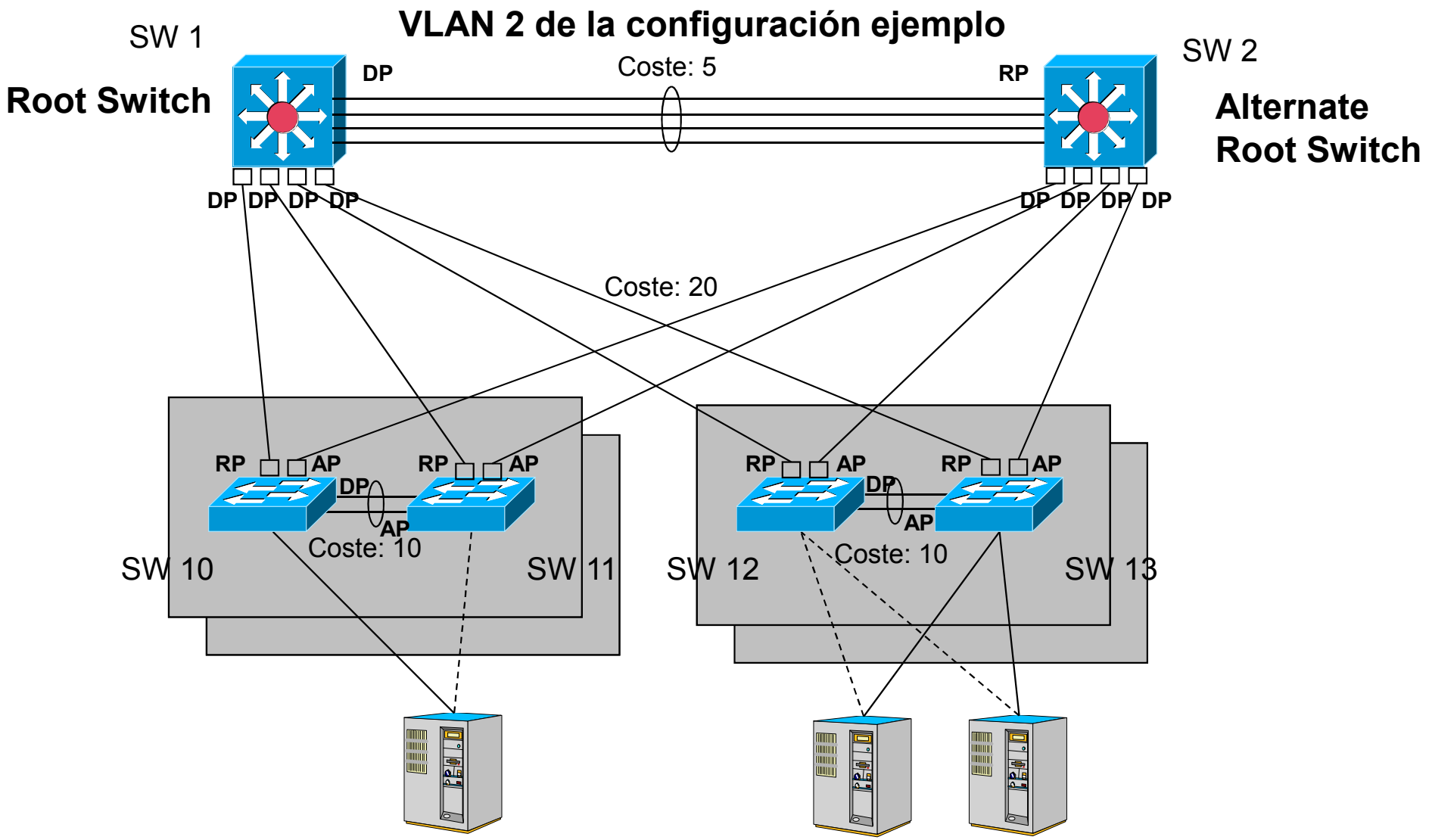
- Periódicamente (cada 2 s. por defecto) cada *switch* genera una ***Bridge Protocol Data Unit, BDPU***, y la envía a todos los *switches* de la red.
- Vídeo muy detallado: https://www.youtube.com/watch?v=-5xeFFmXeXY

Rapid Spanning Tree Protocol, RSTP

AS | E | C

- La **BDPU** contiene:
  - El identificador el *switch* origen que envía la BDPU.
    - Bridge Identifier (BID) : Bridge priority "+" MAC.
  - El puerto del *switch* origen por el que se ha transmitido la BDPU.
  - El identificador del *switch* elegido como raíz.
  - El coste total de alcanzar el *switch* raíz por ese puerto.
- **Cada vez que se recibe una BDPU**, cada *switch* evalúa:
  1. El ***switch* raíz**, como el de mayor prioridad de los identificadores elegidos.
  2. El **RP**, como el puerto que proporciona el camino de menor coste para conectar al *switch* raíz (desempate por port priority).
  3. Los **DP** para los segmentos de red que gestiona, como los que proporcionan el camino de menor coste hasta el *switch* raíz (desempate por BID).
  4. El resto de los puertos quedan como **AP**, y no se utilizan para el envío de tramas.
- **En pocos intercambios, el RSTP converge hasta proporcionar a todos los *switches* el mismo *spanning tree* de configuración de la LAN.**

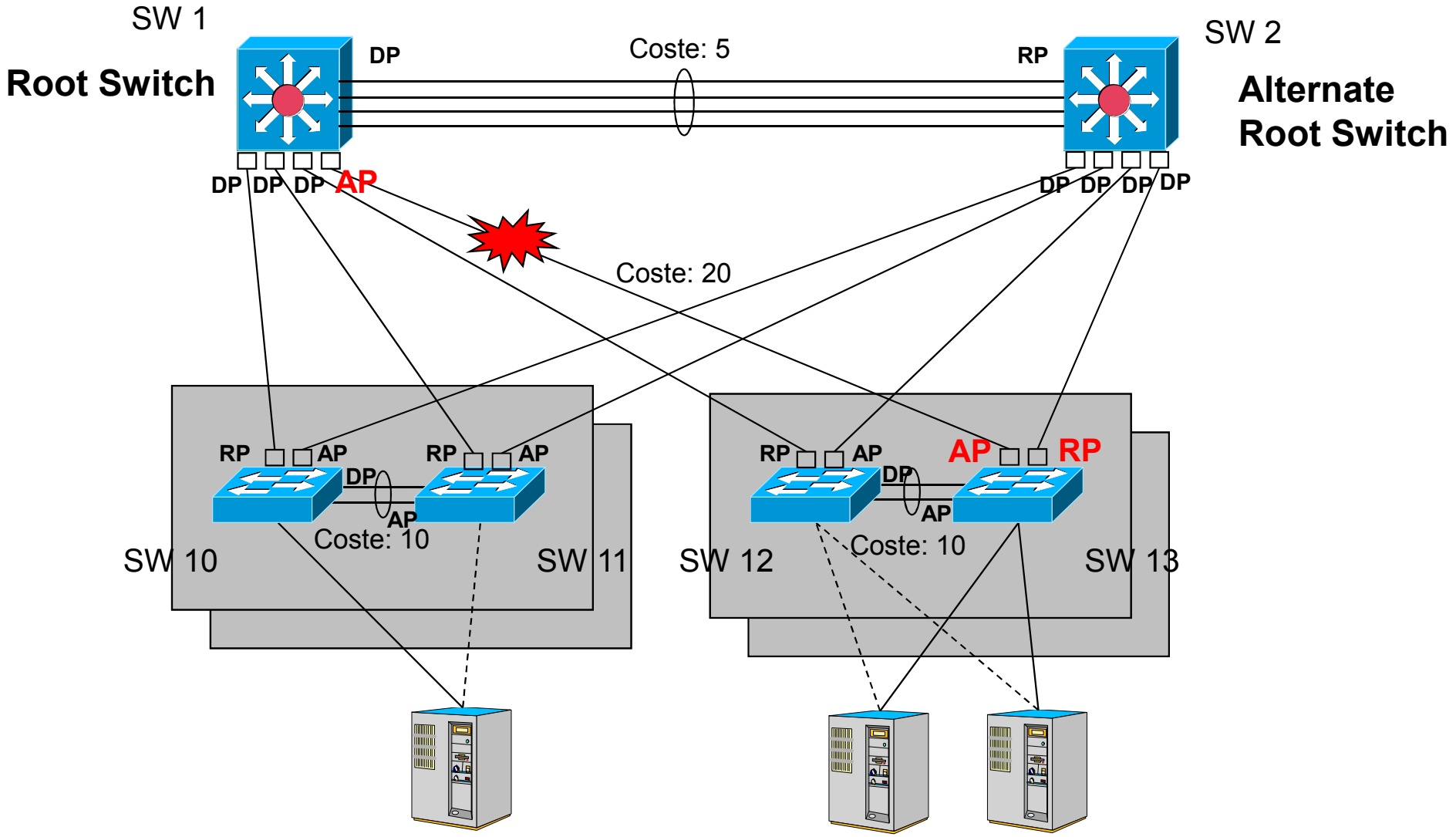
Ejemplo de *spanning tree* formado con RSTP



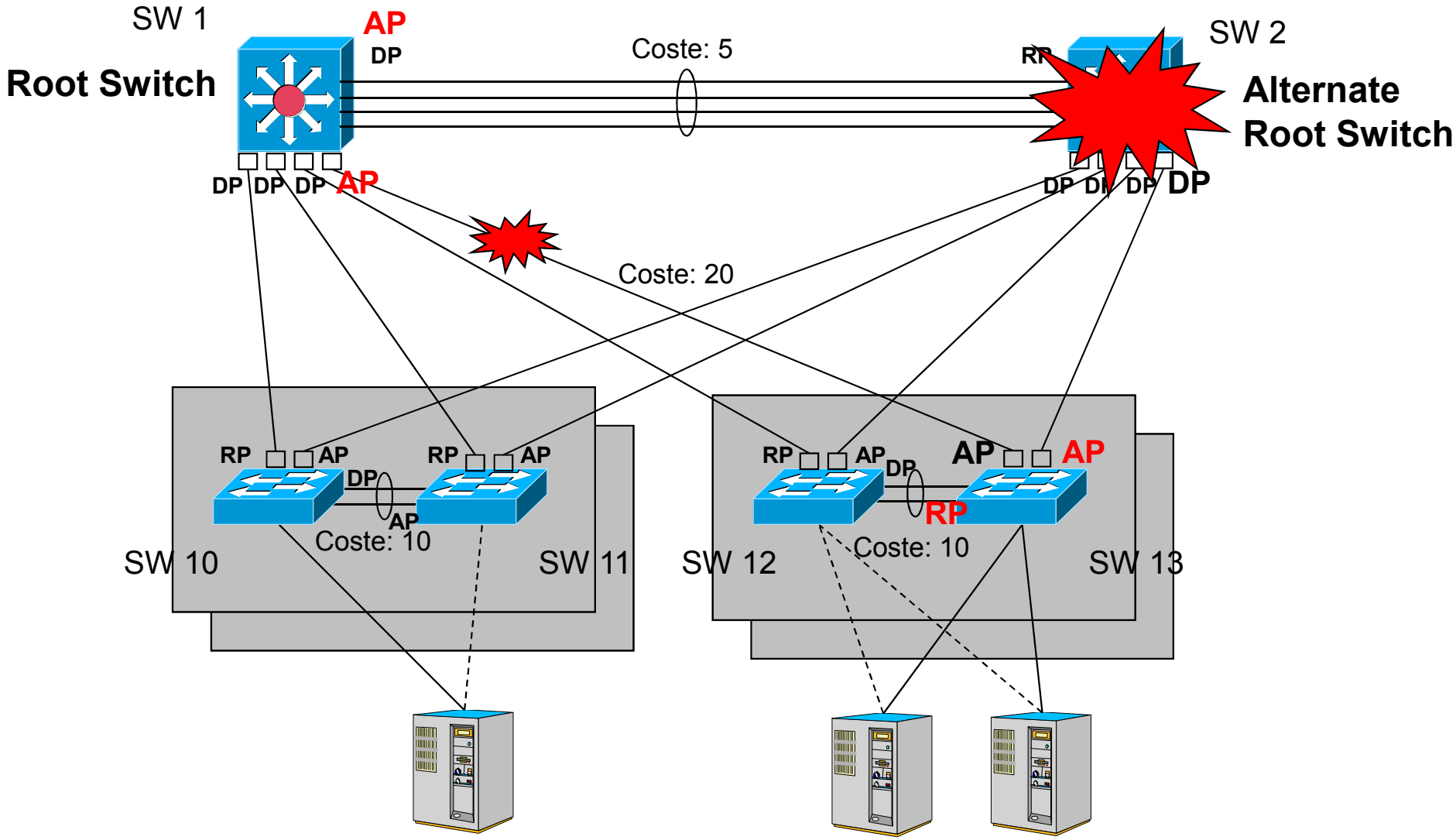
Detección de fallos en *RSTP*

- En RSTP se detecta un fallo por dos posibles causas:
  - Fallo en un enlace conectado a un RP o DP. Detección directa por los *switches* a los que está conectado.
  - Fallo en un *switch*:
    - Las BPDUs actúan como los “latidos del corazón” *(hearthbeats)* o mensajes de *keep-alive* del switch para sus vecinos.
    - Si no se reciben 3 BPDUs consecutivas de un *switch*, se considera que ha fallado.
- Tras el fallo, los *switches* reconfiguran el *spanning tree*, buscando un camino para la conexión de cada uno de sus segmentos al *root switch.*
  - La nueva convergencia es rápida, y se produce mediante intercambios locales de los *switches* afectados por el cambio, que se propagan al resto de la red.
- El *fail-back* sigue el mismo mecanismo.
- En el *spanning tree* normal:
  - Únicamente el *root bridge* transmite BPDUs.
  - Sólo se detecta un problema tras la expiración del temporizador *Max Age*, que es de 20s.
  - Tras esto, se sabe que hay un fallo, pero se ignora dónde se encuentra. Se comienza a recalcular el *spanning tree* desde cero.

Ejemplo de reconfiguración del RSTP tras un fallo en un enlace



Ejemplo de reconfiguración del RSTP tras la caída de un *switch*

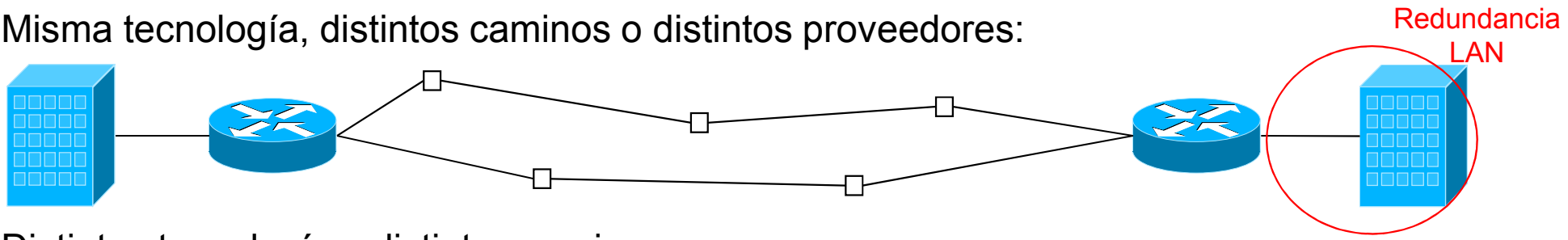


Redundancia en las WAN

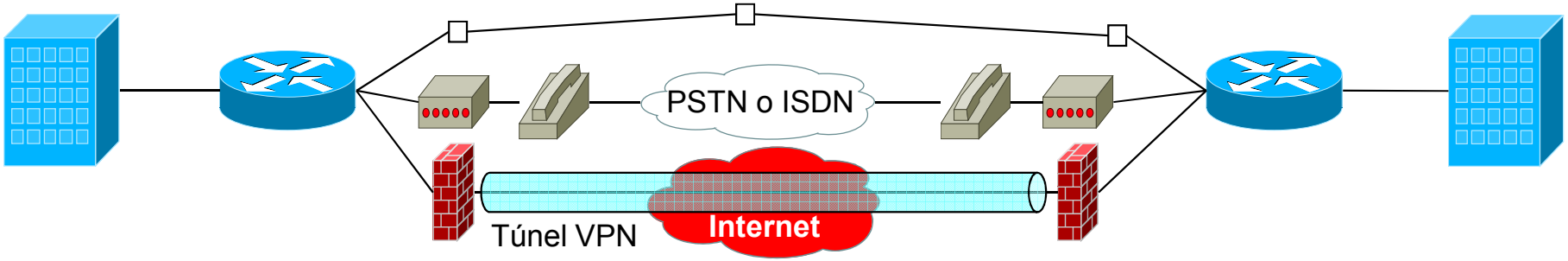
- La interconexión entre elementos distantes de sistemas distribuidos requiere del uso de una Red de Área Extendida *(Wide Area Network, WAN)*.
- Actualmente **TCP/IP** es el protocolo estándar de facto en las comunicaciones en las WAN.
- Dentro de una WAN hay dos tipos fundamentales de componentes: **enlaces** y unidades de encaminamiento ***(routers)***.
- La alta disponibilidad de las WAN se consigue mediante la implantación de **múltiples enlaces y *routers***, de modo que se garantice la existencia de rutas alternativas a través de ella para unir los distintos nodos que forman el sistema distribuido.
- La gestión del tráfico a través de múltiples rutas se realiza a nivel 3 mediante el uso de un **protocolo de encaminamiento dinámico (OSPF, BGP).**
  - Implementado en todos los *routers* y en muchos servidores finales.
  - Permiten el **cálculo dinámico** de rutas de acuerdo con estado actual de la red.
- En el caso de elementos de la red que requieran la configuración de **rutas estáticas** (por ejemplo, estaciones de trabajo, servidores, etc.), el *router* por defecto que se asigna a dichos dispositivos se convierte en un **SPOF**.
  - Para eliminarlo es necesario que dicho *router* utilice una configuración de alta disponibilidad.
  - Se consigue mediante un ***cluster* de *routers***, gestionado por un protocolo como, por ejemplo, el ***Virtual Router Redundancy Protocol, VRRP***.

Redundancia en los enlaces de WAN

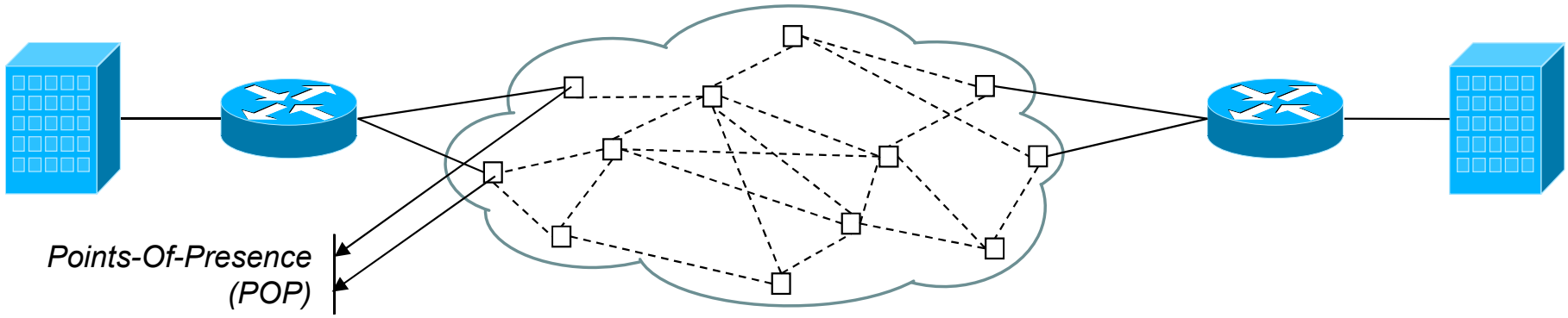
- Misma tecnología, distintos caminos o distintos proveedores:



- Distintas tecnologías, distintos caminos:



- Múltiples accesos a redes de conmutación de paquetes:



Protocolo *Open Shortest Path First, OSPF*

https://www.youtube.com/watch?v=u4tuLXEF_4M

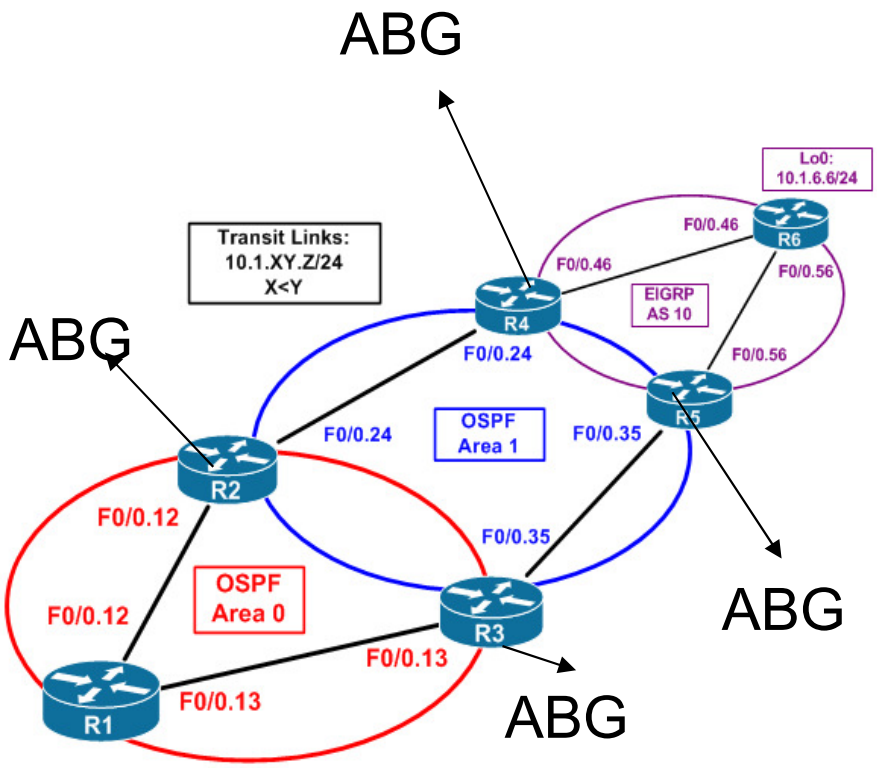
AA | D | C

- Protocolo de encaminamiento dinámico eficaz y ampliamente **utilizado en redes IP autónomas.**
- Basado en el conocimiento de la información de **estado de los enlaces** de toda la red. Con esa información se construye el árbol de caminos más cortos por el algoritmo de Dijkstra.
- Permite realizar una **organización jerárquica** de la red en áreas.
  - Los *routers* encaminan los paquetes dentro de un área.
  - Para realizar el encaminamiento entre áreas se emplean los *Area Border Gateways*.
- El intercambio de información entre *routers* se realiza mediente el protocolo *OSPF*.
  - ***OSPF Hello***: Empleado para conocer la topología de la red.
    - Cada *router* conoce sus vecinos (*routers* adyacentes).
    - En redes multiacceso (LAN) se elige el *Designated Router (DR)*, que actúa como representante de todos los conectados a dicha red.
  - ***Link State***: Intercambio de información del estado de los enlaces *(Link State Advertisements, LSA)*.
- Permite **encaminamiento en paralelo** a través de rutas de igual peso *(multipath routing)*.
  - Permite tener redundancia activo-activo entre líneas de igual peso, y activo-pasivo entre líneas de distintos pesos.
- Permite el empleo de **varias métricas** simultáneamente. Utilizado para hacer encaminamiento basado en distintas calidades de servicio *(Quality of Service, QoS)*.

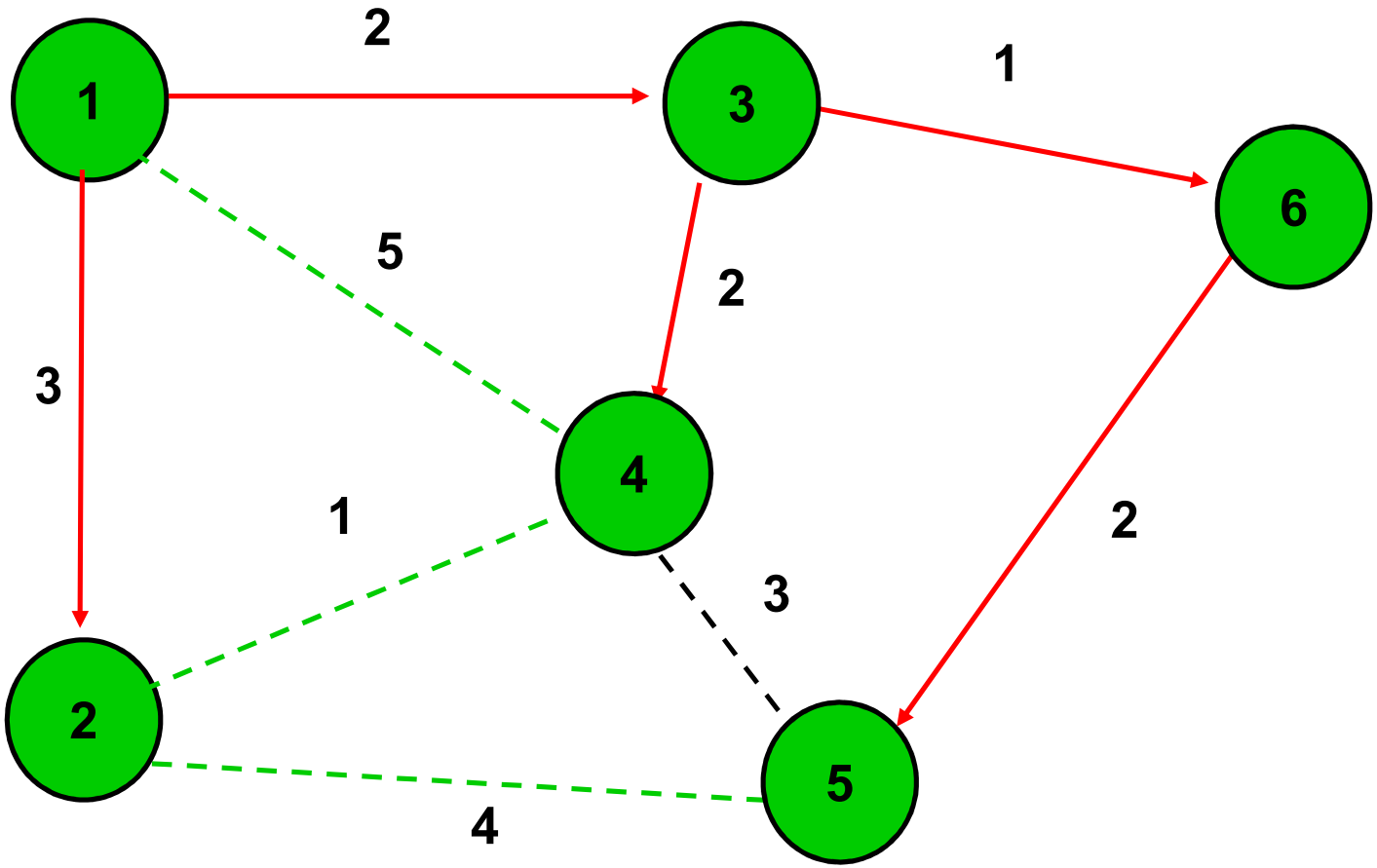
Ejemplo BGP, OSPF

AA | D | C

Red IP autónoma



Ejemplo de red y árbol de camino más corto



Detección de fallos en OSPF

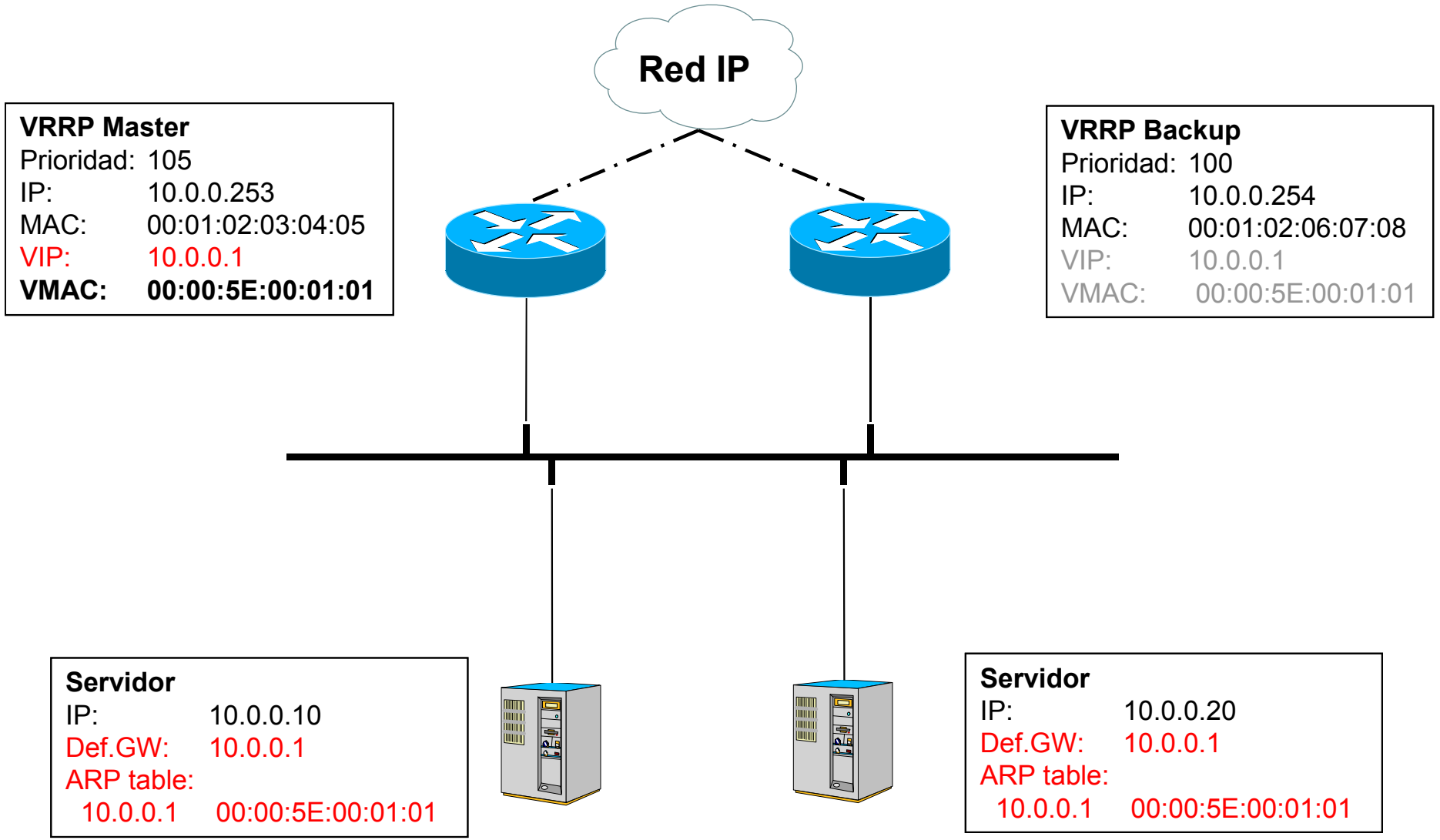
- Por **detección de la caída de un enlace**.
- Por **detección de que no hay dispositivos conectados a una VLAN** *(Autostate)*.
  - En *switches* con conmutación mixta a nivel 2 y 3, esta situación hace que se desactive la interfaz virtual del *router* a dicha VLAN.
  - Este cambio en la situación de los enlaces se notifica al resto de los *routers*.
- A través del **protocolo *Hello***. Sirve de *keep-alive* para indicar que el *router* sigue activo.
  - Los paquetes *Hello* se intercambian periódicamente entre *routers*.
    - Por defecto, cada 10 s.
  - Si se pierden 4 *Hello* consecutivos de un *router*, se considera que está inactivo.
    - El *router* que lo detecta envía un LSA a todos los *routers* de su área.
  - Se deben elegir los *time-outs* del protocolo *Hello* de modo que sean compatibles con los de protocolos inferiores (por ejemplo, RSTP en LAN).
- El *fail-back* sigue el mismo mecanismo.

Virtual Router Redundancy Protocol, VRRP

AS | E | C

- Protocolo de ***clustering* para *routers***. Empleado para proporcionar redundancia en *routers* en los casos en que representan un punto único de fallo en rutas de la red.
- El *router* se sustituye por un ***virtual router***, formado por un *router* activo y otro en *stand-by.*
- A cada *router* se le asigna una prioridad.
- Al *virtual router* se le asigna una ***Virtual IP, VIP***, y una dirección **MAC virtual (VMAC)**.
- El *router* de mayor prioridad (o el primero en conectarse, según se configure) es el activo.
  - Asume las direcciones VIP y VMAC como propias.
- Otros protocolos similares propiedad de Cisco son:
  - *Hot Standby Router Protocol, HSRP*
  - *Gateway Load Balancing Protocol, GLBP.*
    - GLBP permite configuración del *cluster* de *routers* en modo activo-activo.
- Estos protocolos se emplean también para lograr la alta disponibilidad de otros dispositivos que en su funcionalidad también incluyen encaminamiento de paquetes:
  - Balanceadores de carga.
  - Cortafuegos de seguridad.

Ejemplo de configuración de red con VRRP



Detección de fallos en VRRP

- El *router* activo envía paquetes *Hello* por *multicast* con una periodicidad definida en configuración *(advertisement interval)*. Por defecto es 1s.

- Si transcurre el tiempo configurado en el *master-down interval* sin recibir paquetes *Hello* del master se considera que ha caído, y el siguiente *router* en prioridad pasa a activo.
  - Por defecto es el triple del *advertisment interval* mas un tiempo de *skew* calculado a partir del nivel de prioridad de cada *router*.
    - A mayor prioridad del *router*, menor es el tiempo de *skew*.
    - De este modo, en caso de fallo, el *router* de mayor prioridad es el que pasa a ser el activo en caso de fallo.
  - Definir de modo que no interfiera con otros *time-outs* de protocolos de enlace.
    - Tratar que el enlace empleado para transmitir el *Hello* sea lo más directo posible.
- En el *fail-back* hay dos opciones:
  - Si la configuración activa *preempt*, el router de mayor prioridad asume el tráfico al detectar un paquete *Hello* de un *router* de prioridad inferior.
    - Supone una breve interrupción del servicio mientras se produce el cambio.
  - Si la configuración desactiva *preempt*, se deja el *router* actual como activo hasta un posible fallo, quedando el reincorporado como *router* de *backup*.

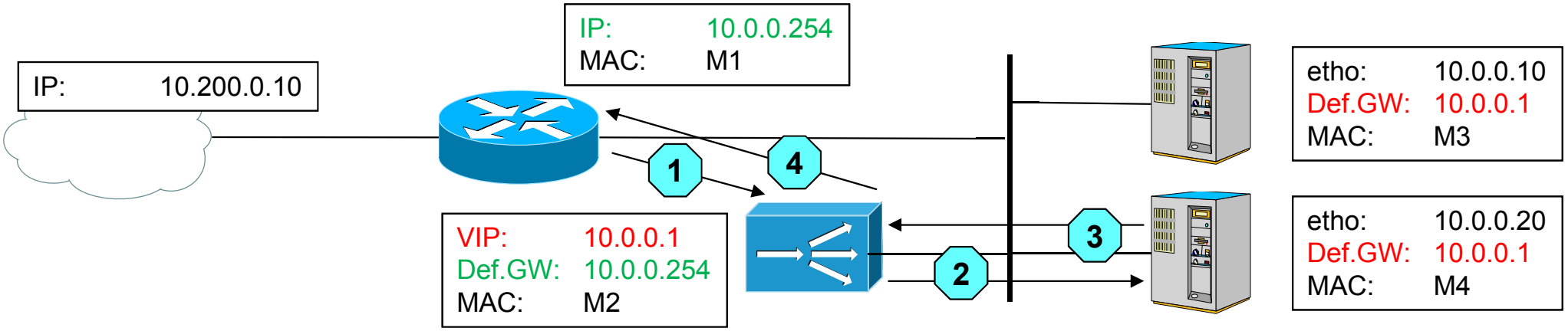
Balanceadores de carga

- Un balanceador de carga *(Load Balancer, LB)* es un dispositivo capaz de **distribuir peticiones entre un grupo de servidores para su proceso.**
- Sus **objetivos** son dos:
  - **Aumentar la capacidad de proceso** para atender al servicio, aumentando el número de servidores que lo prestan.
  - **Aumentar la disponibilidad del servicio**, al proporcionar elementos redundantes en paralelo para prestarlo.
- Previamente, esta función se hacía en redes IP a través del DNS, resolviendo el mismo nombre por distintas direcciones IP. Solución muy limitada:
    - No garantiza que el servidor esté activo.
    - Cambios en el DNS se propagan muy lentamente por la red.
- **El LB se interpone entre la red y los servidores para interceptar el tráfico y dirigirlo al destino más adecuado.**
  - Frente a un *switch*, que sólo maneja información de nivel 2, y a un *router*, que únicamente maneja información de nivel 3, el LB también utiliza información de los niveles de transporte y aplicación para realizar el encaminamiento.
    - Como mínimo, el LB debe garantizar que en protocolos orientados a conexión (TCP) todos los paquetes de la misma conexión se entregan al mismo servidor.
  - Por el motivo anterior también se suelen denominar ***Application Switches (AS).***

Funcionamiento básico de los balanceadores de carga

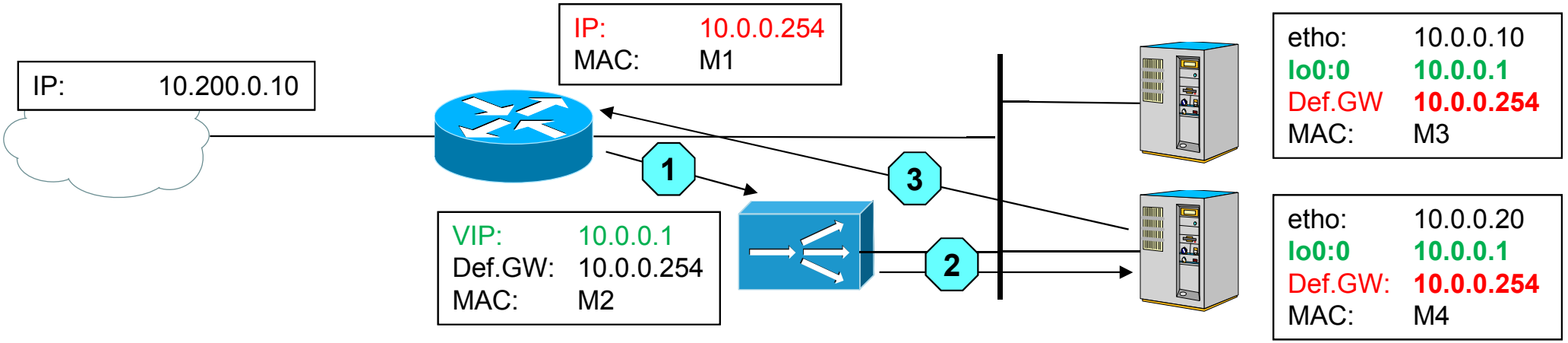
- **Configuración:**
  - Se define una **dirección IP virtual (VIP), puerto y protocolo** para el servicio a prestar.
    - Esta VIP se asigna al LB.
  - Se asocian a estos datos los servidores que prestan el servicio (IP reales y puerto).
    - Con esta información queda construida la tabla de encaminamiento del LB.
  - Definir el tipo de mecanismo de verificación de actividad *(**health check**)* que utilizará el LB para comprobar el estado de los servidores.
  - Configurar el **mecanismo de distribución** de carga que se empleará.
  - Configurar el **mecanismo de entrega** de paquetes a utilizar.
- **Encaminamiento:**
  - El cliente envía una solicitud de servicio a la VIP.
  - El LB recibe la solicitud, y **determina el servidor al que tiene que enviar la petición**.
    - Según estado de actividad de los servidores y algoritmo de distribución de la carga.
    - En todos los paquetes UDP y en el TCP SYN en caso de conexiones TCP.
  - El LB envía la petición al servidor asignado empleando el mecanismo de entrega elegido.
    - En TCP, guarda la asignación en una tabla para mantenerla en toda la conexión.
  - El servidor devuelve la contestación, conforme al mecanismo de entrega elegido.

Entrega de paquetes mediante *Destination NAT*



	SRC MAC	DST MAC	SRC IP	DST IP
1	M1	M2	10.200.0.10	10.0.0.1
2	M2	M4	10.200.0.10	**10.0.0.20**
3	M4	M2	10.0.0.20	10.200.0.10
4	M2	M1	**10.0.0.1**	10.200.0.10

Retorno directo del servidor *(Direct Server Return)*



	SRC MAC	DST MAC	SRC IP	DST IP
1	M1	M2	10.200.0.10	10.0.0.1
2	M2	M4	10.200.0.10	10.0.0.1
3	M4	M1	10.0.0.1	10.200.0.10

Mecanismos de distribución de la carga

- Los **mecanismos de distribución de la carga** son similares a los empleados por otros repartidores de trabajo:
  - *Round Robin*.
  - Menor número de conexiones.
  - Cualquiera de los anteriores con pesos distintos por servidor.
  - Menor tiempo de respuesta.
  - Por origen de la petición.
  - Por datos contenidos en el paquete de aplicación (nivel 7).
    - Requiere el empleo de ***delayed binding***.

- En todos los casos, es necesario considerar **condiciones límite para realizar la entrega**:
  - Máximo número de conexiones permitido.
  - Umbral de tiempo de respuesta.
  - *Graceful Shutdown*: Suspensión de nuevas asignaciones a un servidor para que pueda desconectarse al terminar las conexiones pendientes.
  - Tiempo *de gracia*: Intervalo de tiempo que se concede a un servidor desde que se detecta que está activo hasta que se le comienza a enviar tráfico, para permitir su estabilización adecuada.

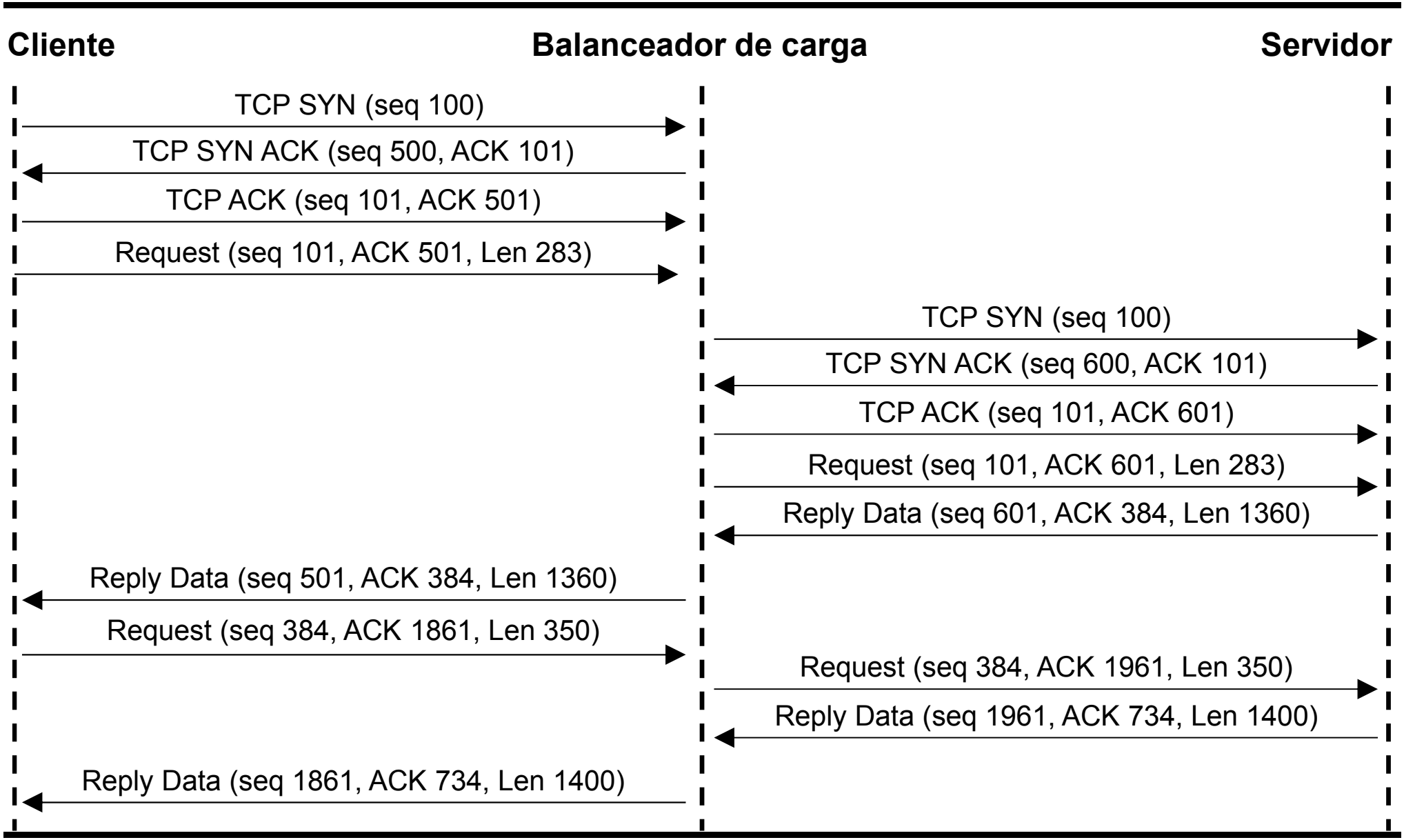
Detección de fallos en servidores conectados a LB

- Mediante los mecanismos de verificación de actividad *(health check)*:
  - Mediante monitorización “en banda” *(in-band)*, basada en el tráfico de clientes.
    - Requiere que las respuestas del servidor pasen por el LB, pero no tráfico adicional.
    - Monitorización del *handshake* TCP.
    - Monitorización de los códigos de retorno de aplicación. Por ejemplo, códigos de respuesta en protocolo HTTP (200 OK / 500 Server Error) .
  - Mediante monitorización “fuera de banda” *(out-of-band)*.
    - Utilizable aunque las respuestas no pasen por el LB, pero requiere tráfico adicional.
    - Pruebas nivel 3: ICMP echo / reply.
    - Pruebas nivel 4:
      - Apertura / cierre de conexiones TCP.
      - Envío de paquetes UDP / espera ICMP *destination unreachable*.
    - Pruebas nivel 7 (aplicación):
      - Envío de una solicitud de conexión al servidor o una consulta simple e inocua (HTTP HEAD, SSL *handshake,* conexión FTP, resolución DNS...).
    - Agentes específicos: SNMP, *Dynamic Feedback Protocol (DFP)* de Cisco, etc.
- Si el servidor falla en la prueba tras un número de reintentos, se marca como inactivo.
- *Fail-back*: Mediante monitorización fuera de banda se comprueba si vuelve a la actividad.

Afinidad de la sesión

- **Sesión**: secuencia de trabajo a nivel aplicación. No soportada de modo estándar en TCP/IP.
- **Los LB deben ser capaces de mantener sesiones entre clientes y servidores**. Los mecanismos más empleados para realizarlo son:
  - **Por origen**: Distribución de la carga por origen de la petición. Finalización de la sesión por *time-out* entre paquetes consecutivos.
    - Problema: si existe un *proxy* o *Source NAT* cliente entre clientes y servidores, todas las peticiones se reciben de la misma dirección IP.
  - **Por concurrencia de conexiones**: una nueva conexión de un cliente se envía al mismo servidor que mantiene la actual. Válido, por ejemplo, para FTP en modo pasivo.
  - **Por análisis de contenido del paquete de aplicación**.
    - Buscar la información en el paquete que identifica la sesión:
      - *Cookies* de sesión, campos en la URL *(URL rewrite)...*
      - Identificador SSL.
    - Requiere analizar el paquete de aplicación antes de elegir el servidor destino.
      - El balanceo no se puede hacer por paquete. Es necesario partir la conexión TCP en el LB *(**delayed binding**)*.
    - Finalización de la sesión por *time-out* entre paquetes consecutivos.

Delayed Binding



Redundancia en los sistemas de almacenamiento de información

- **La información es el punto más crítico de todo sistema informático.**
  - Perder datos o no tenerlos disponibles adecuadamente puede ser fatal para cualquier tipo de negocio.
- La redundancia en los elementos de información es más compleja que en los elementos de procesamiento.
  - Es necesario **mantener consistente la información** en todas las copias.
  - Los elementos de procesamiento redundantes deben acceder a la misma copia de la información.
    - Es necesario utilizar **elementos de almacenamiento externos compartidos**.
- La utilización de elementos de almacenamiento externos permite cambiar el modo de diseño de los sistemas distribuidos, apareciendo arquitecturas con nuevas características:
  - Consolidación del almacenamiento en disco en **servidores especializados**.
    - Incorporan funciones autónomas de gestión de la información y las copias.
  - **Virtualización**: independencia del almacenamiento físico (discos y cintas) y las unidades asignadas a los servidores *(Logical Units, LUNs* en discos y cintas virtuales*)*.
  - **Centralización** de la monitorización y gestión del almacenamiento.
  - Necesidad de crear de **redes de almacenamiento** para acceder a la información.
    - En ellas se incluyen los servidores de discos y los dispositivos de cinta.

Arquitecturas de almacenamiento

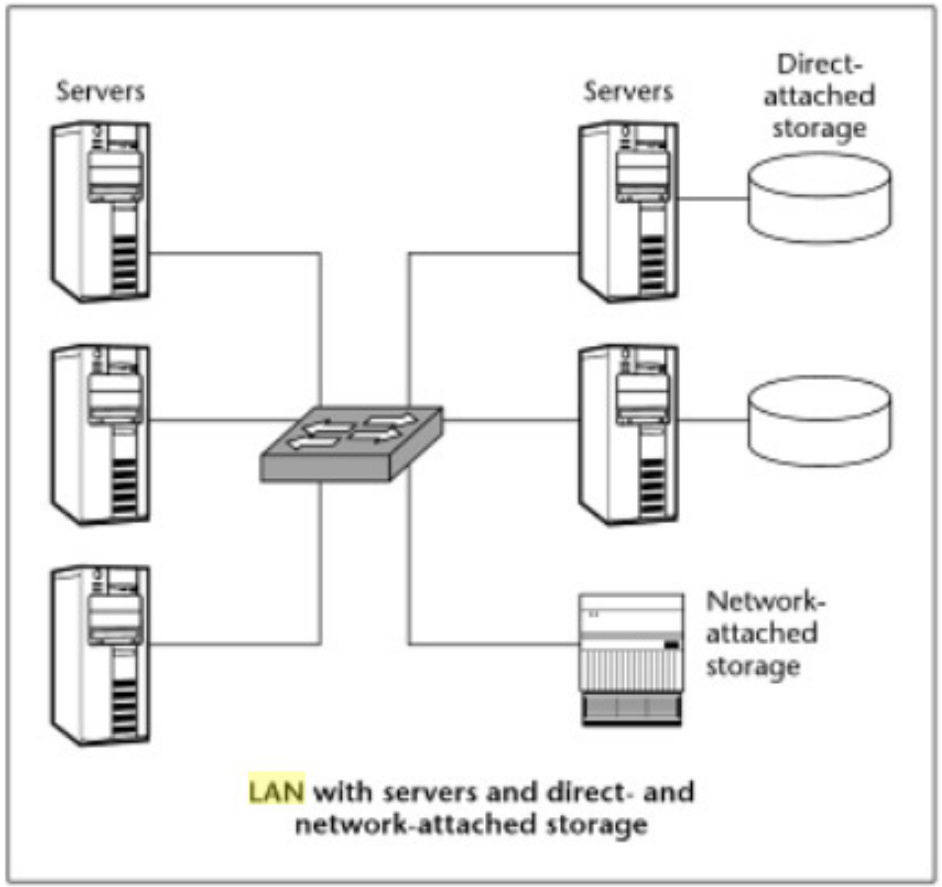
- **Conexión directa *(Direct Attached Storage, DAS)*.**
  - Normalmente se usa directamente el protocolo SCSI en alguna de sus variantes.
- **Conexión a red *(Network Attached Storage, NAS)*.**
  - Servidores de almacenamiento conectados a **LAN**.
  - Usa protocolos de compartición de datos en red (CIFS, NFS).
- **Red de área de almacenamiento *(Storage Area Network, SAN)***
  - Red específica para conectar dispositivos de almacenamiento.
  - Utiliza fibra óptica como medio de transporte.
  - Usa protocolo *FibreChannel (FC)*.
  - Es el mecanismo más utilizado en la actualidad en entornos de alto rendimiento y alta disponibilidad.

- **Internet SCSI (iSCSI)**. Encapsulamento de SCSI-3 sobre IP.
- ***FibreChannel Over IP (FCIP)*.** Encapsulamiento de FC sobre IP. Establece túneles FC sobre conexiones TCP.
  - Mantiene las sesiones FC extremo a extremo entre servidor y dispositivos.
- ***Internet Fibre Channel Protocol (iFCP)*.** También implementa FC sobre IP, pero no en modo túnel, como el FCIP, sino interpretando el protocolo FC.
  - Las sesiones no son extremo a extremo, sino que se interrumpen en el *router* iFCP.

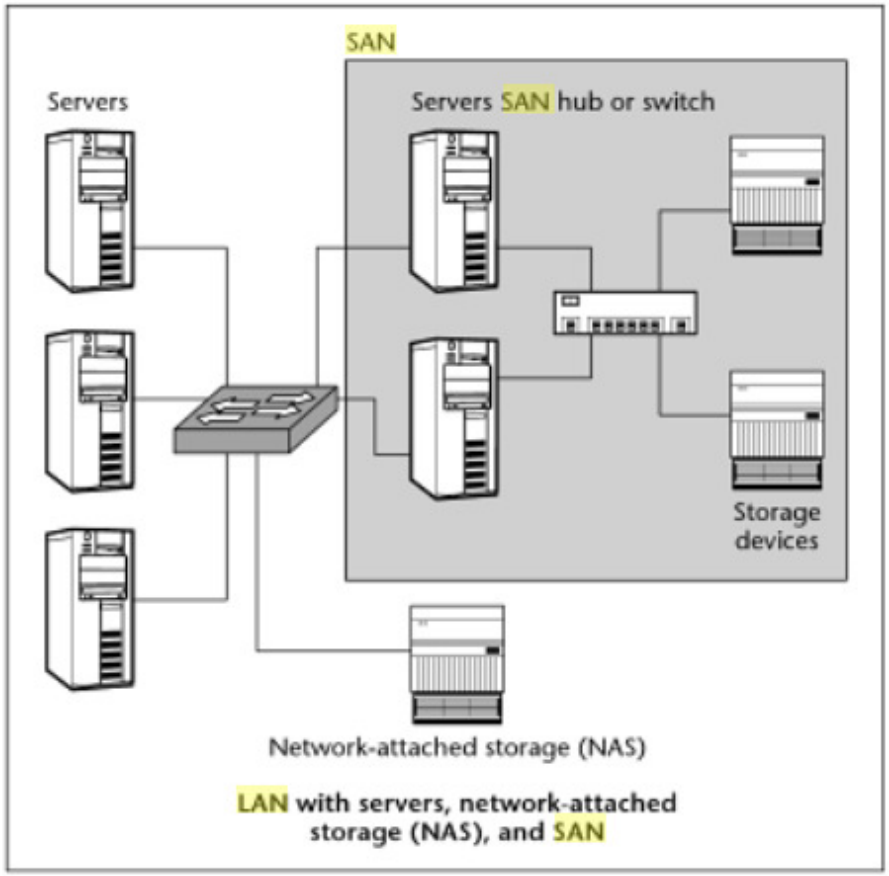
Almacenamiento sobre IP

Arquitecturas de almacenamiento

LAN típica: servidores, DAS y NAS



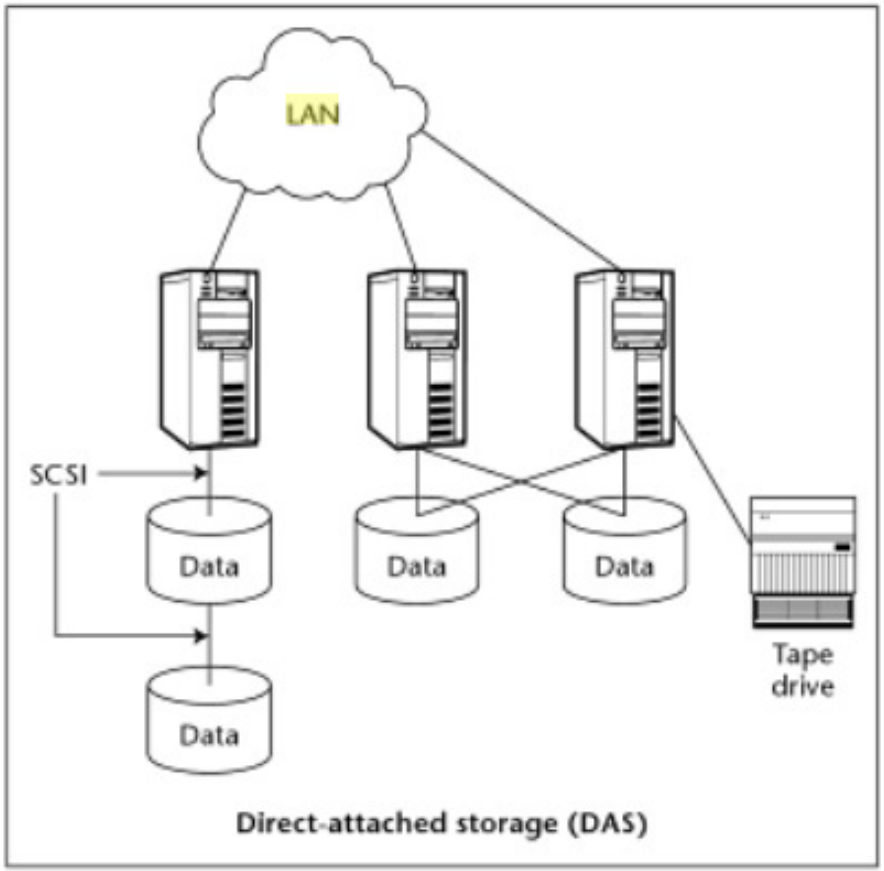
SAN típica: LAN con servidores, NAS y red SAN

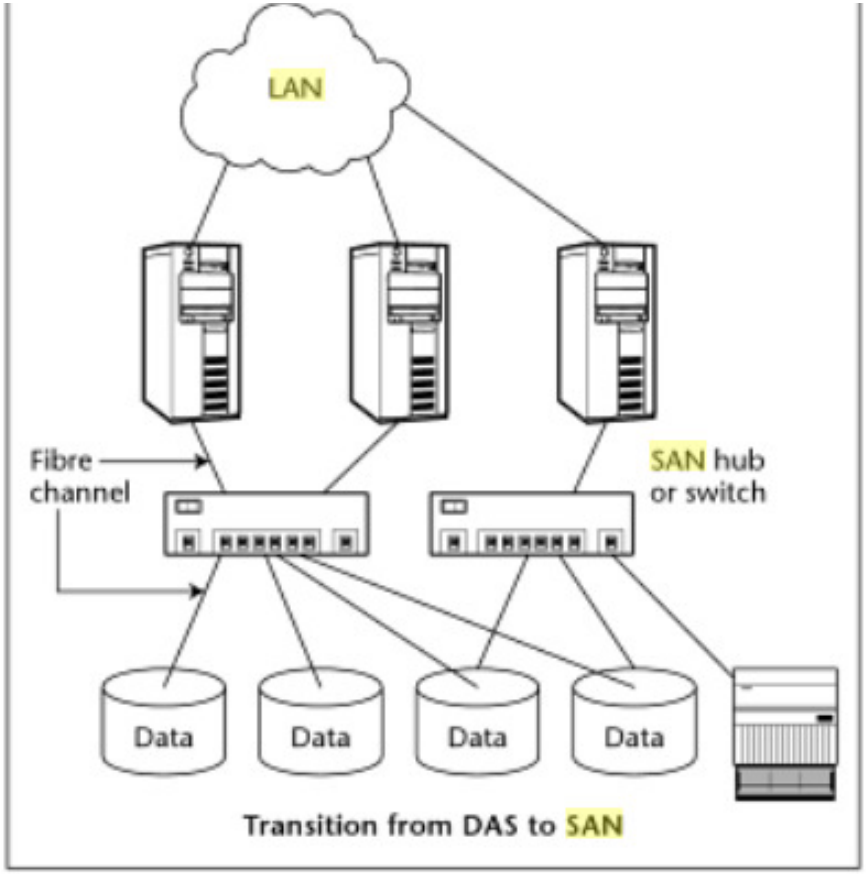


Fuente: JAYASWAL, K., *Administering Data Centers*, Indianapolis (USA), Wiley, 2006.

Arquitecturas de almacenamiento

Small Computer System Interface (**SCSI)** is a **set of standards** for physically connecting and transferring data between computers and peripherical devices. The SCSI standards define commands, protocols and electrical and optical interfaces. SCSI is most commonly used for hard disks and tape drives, but it can connect a wide range of other devices, including scanners and CD drives, although not all controllers can handle all devices.

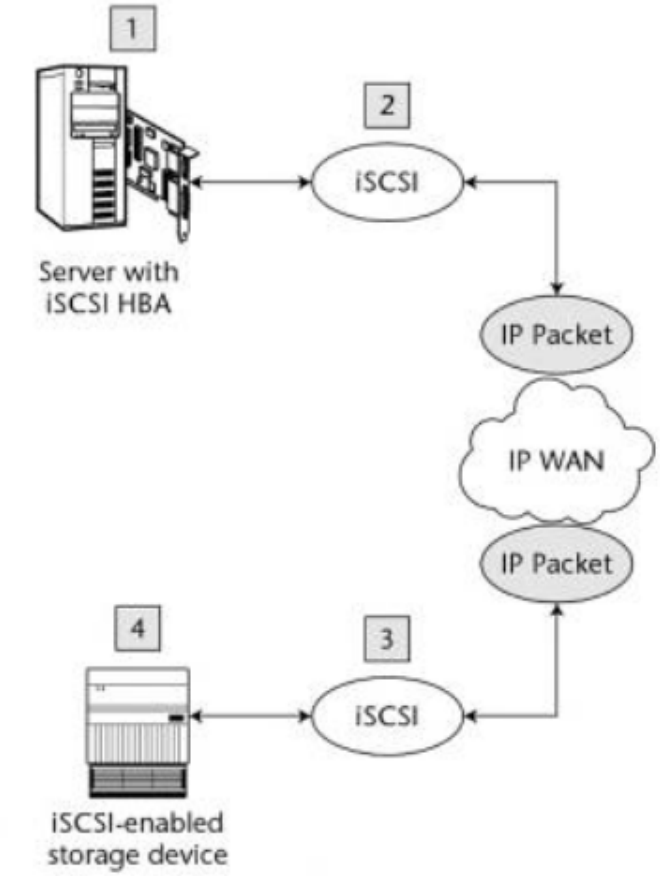




Fuente: JAYASWAL, K., *Administering Data Centers*, Indianapolis (USA), Wiley, 2006.

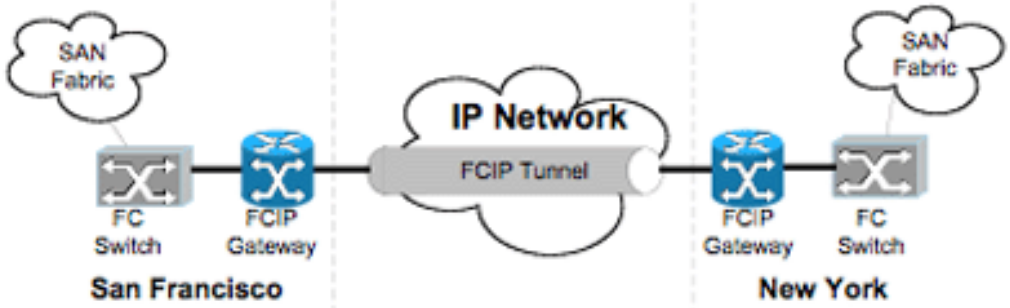
Arquitecturas de almacenamiento

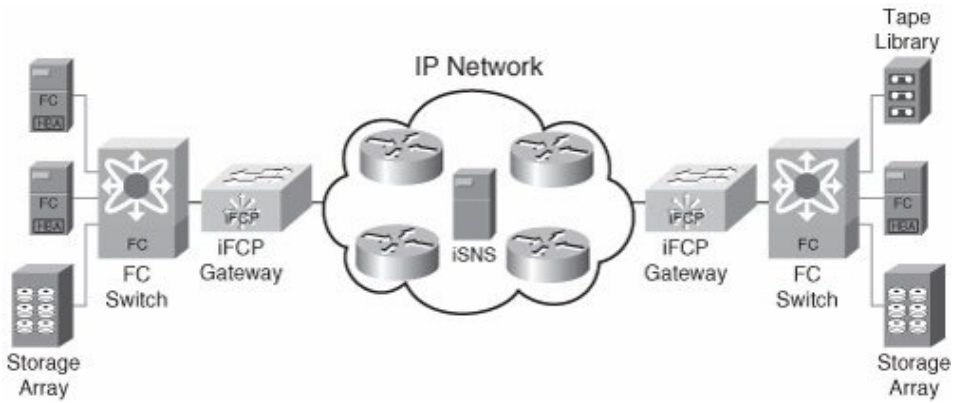
Flujo de datos con iSCSI



Fuente: JAYASWAL, K., *Administering Data Centers*, Indianapolis (USA), Wiley, 2006.

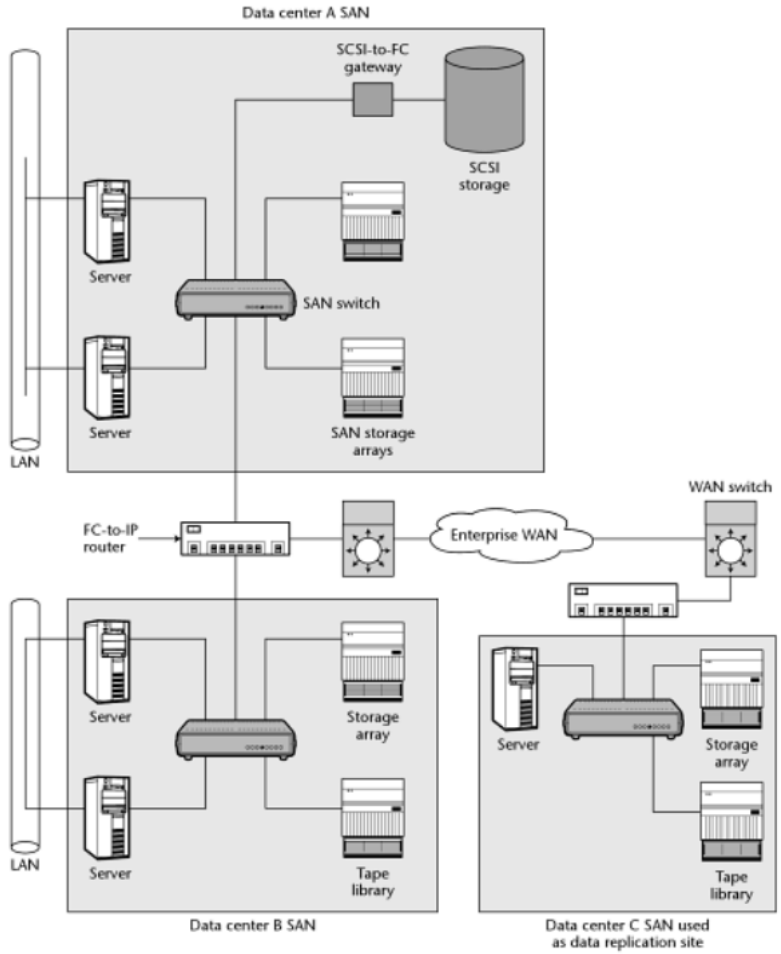
FCIP versus iFCP





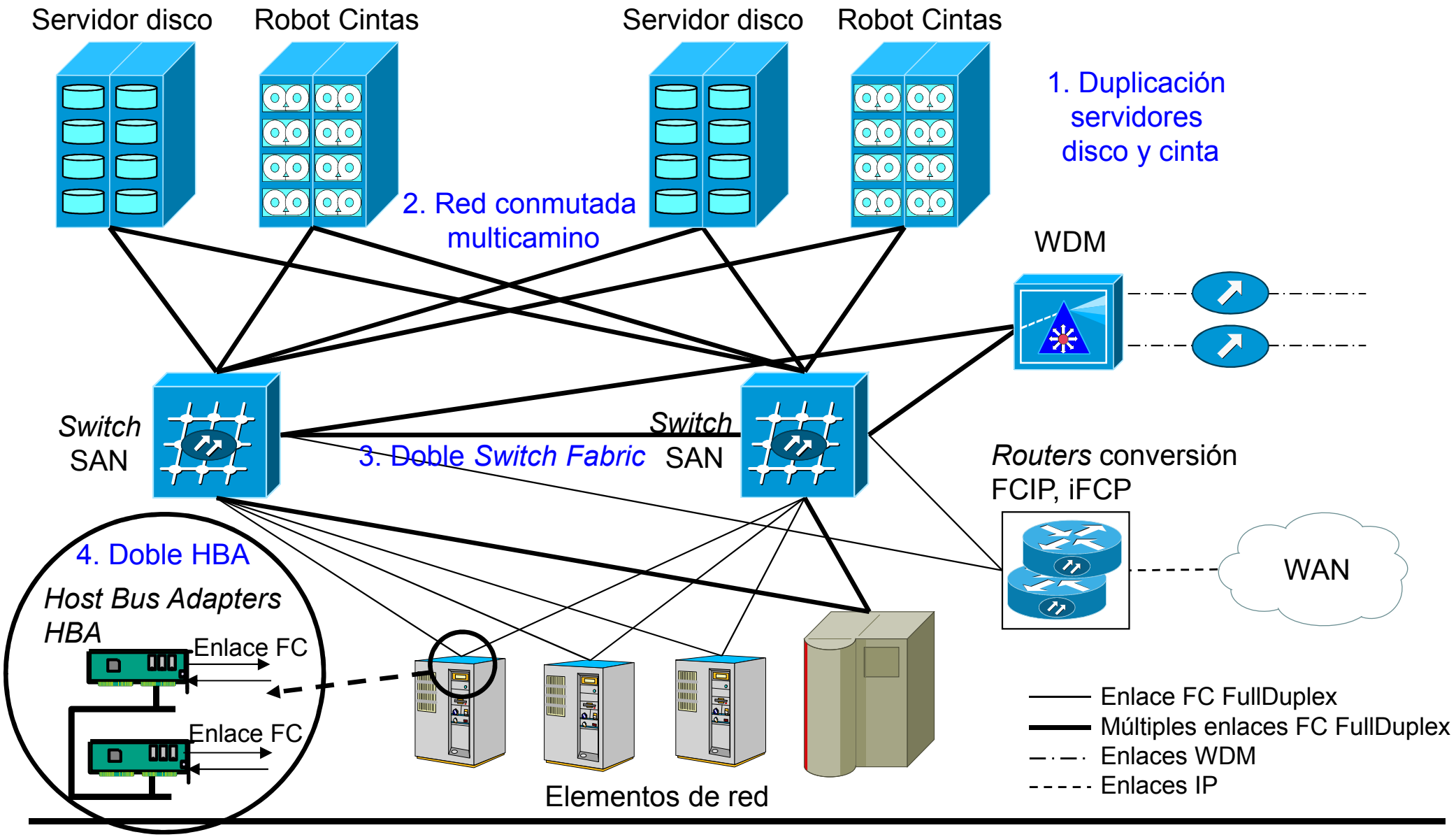
Arquitecturas de almacenamiento

SAN de alta disponibilidad con copia remota de datos



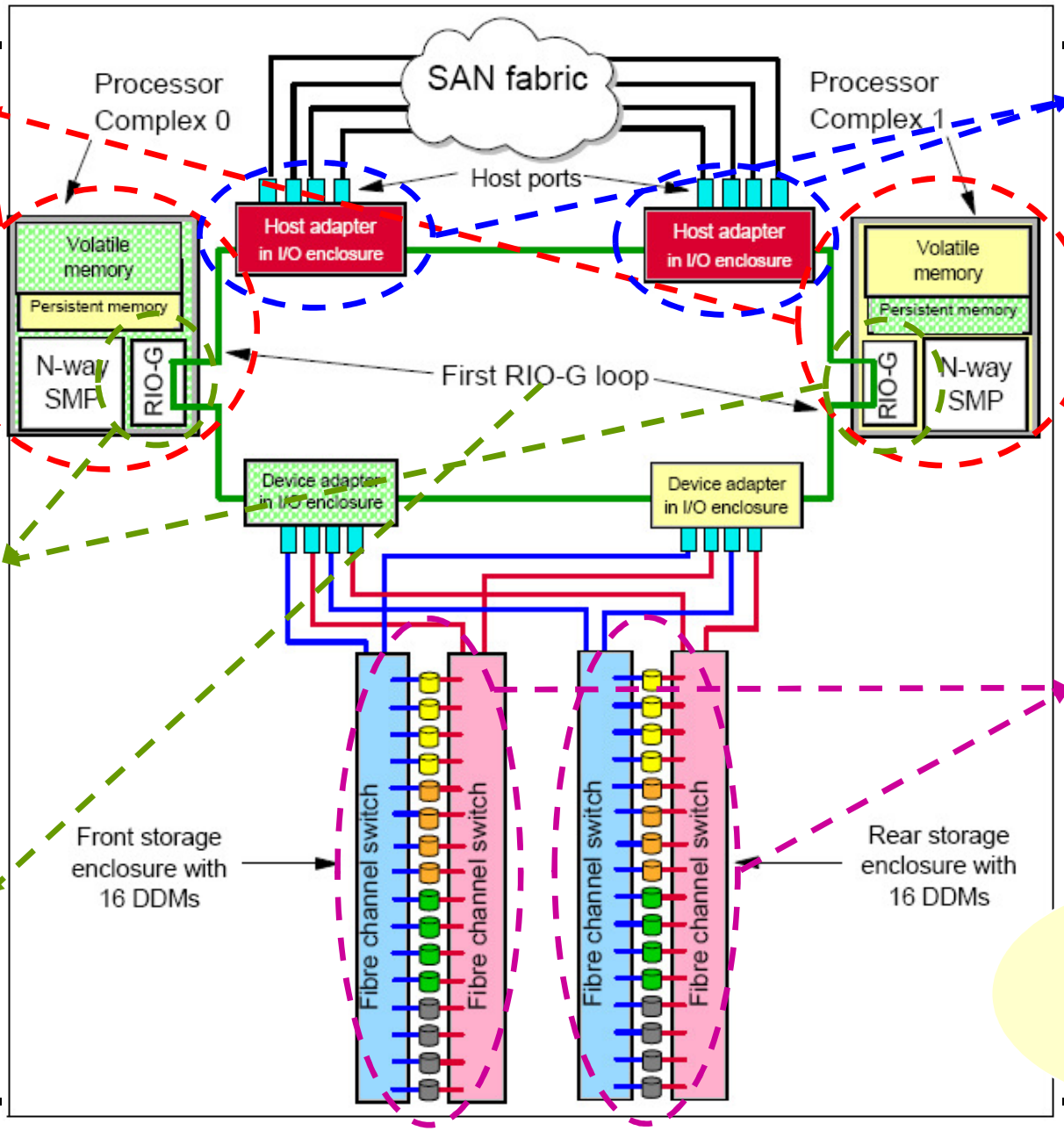
Fuente: JAYASWAL, K., *Administering Data Centers*, Indianapolis (USA), Wiley, 2006.

Modelo de arquitectura de alta disponibilidad para SAN



Estructura interna de un servidor de disco para SAN IBM DS8000

Estructura interna de un servidor de disco para SAN



Unidades de proceso

HBAs compartidos para conexión externa

Remote Input Output (RIO-G)
Puertos conexión
Unidad de proceso
A los FC-AL internos

Disk Drivers Modules
Contienen los discos

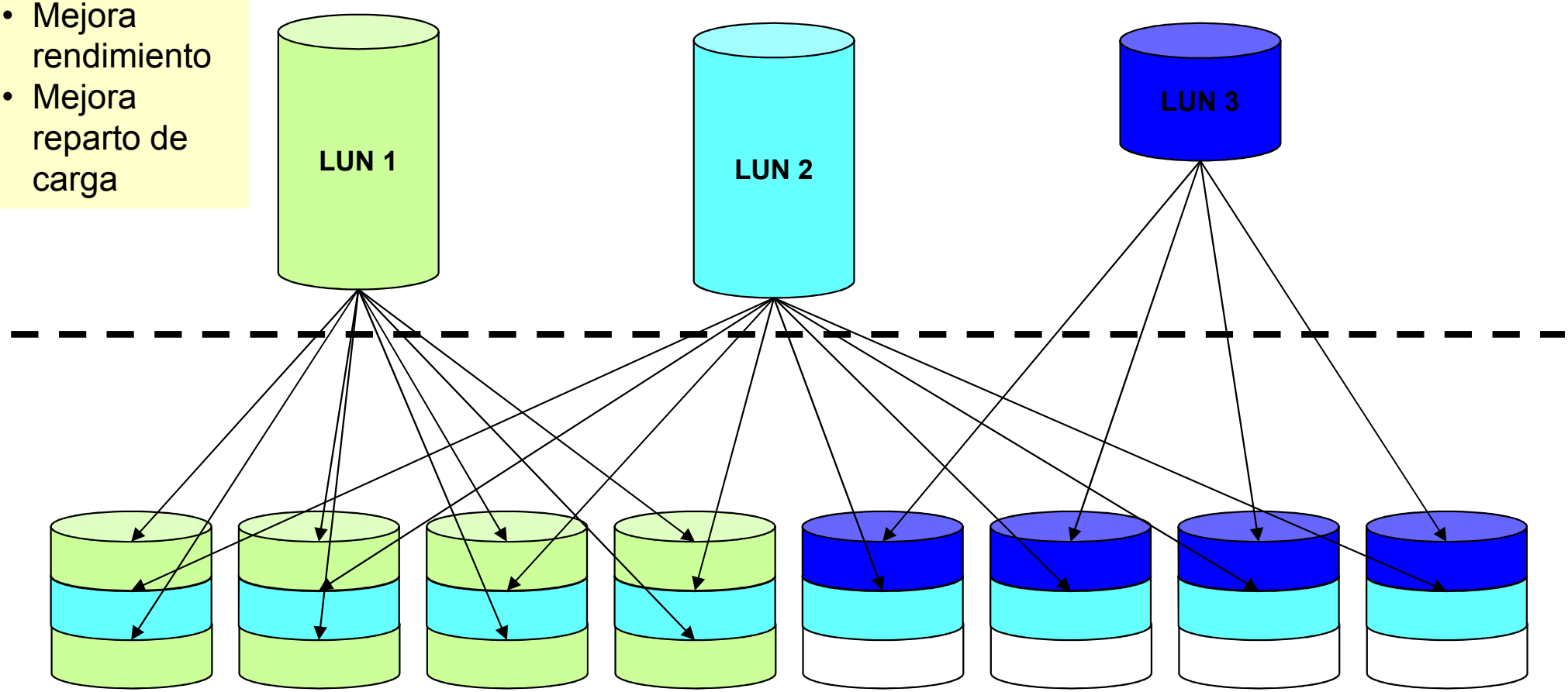
FibreChannel Arbitrated Loops (FC-AL)

- Elementos redundantes
- Múltiples caminos

Asignación de espacio a una Unidad Lógica

Unidades lógicas asignables (LUNs): el servidor de disco asigna espacio a los servidores a través de LUNs

- Mejora rendimiento
- Mejora reparto de carga

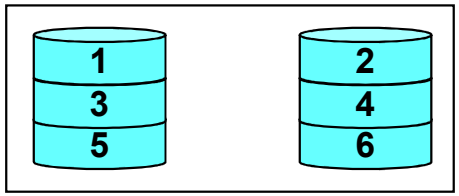


Redundant Array of Independent Disks, RAID

Sistema de tolerancia a fallos para discos.

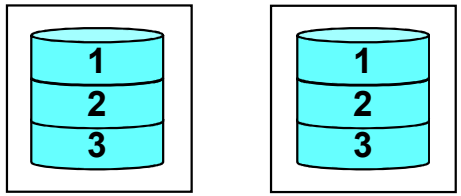
Mejoras en disponibilidad y/o rendimiento.

- **RAID-0: *Stripping***.



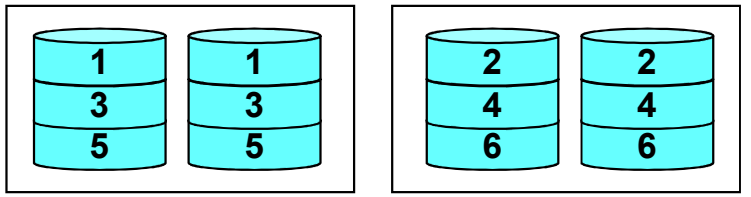
  - Información segmentada y repartida.
  - Mejora rendimiento (R&W).
  - No hay protección contra fallos.
  - No necesita espacio adicional.

- **RAID-1: *Mirroring***.



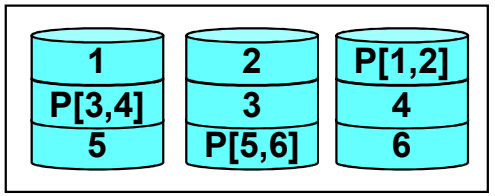
  - Información duplicada.
  - Mejora rendimiento en lectura.
  - Protección contra fallos por copia.
  - Duplicación espacio necesario.

- **RAID-10: *Stripped Mirrors***



  - Información repartida y duplicada.
  - Mejora rendimiento (R&W).
  - Protección contra fallos por copia.
  - Duplicación espacio necesario

- **RAID-5: *Stripping Distributed Parity Based***.

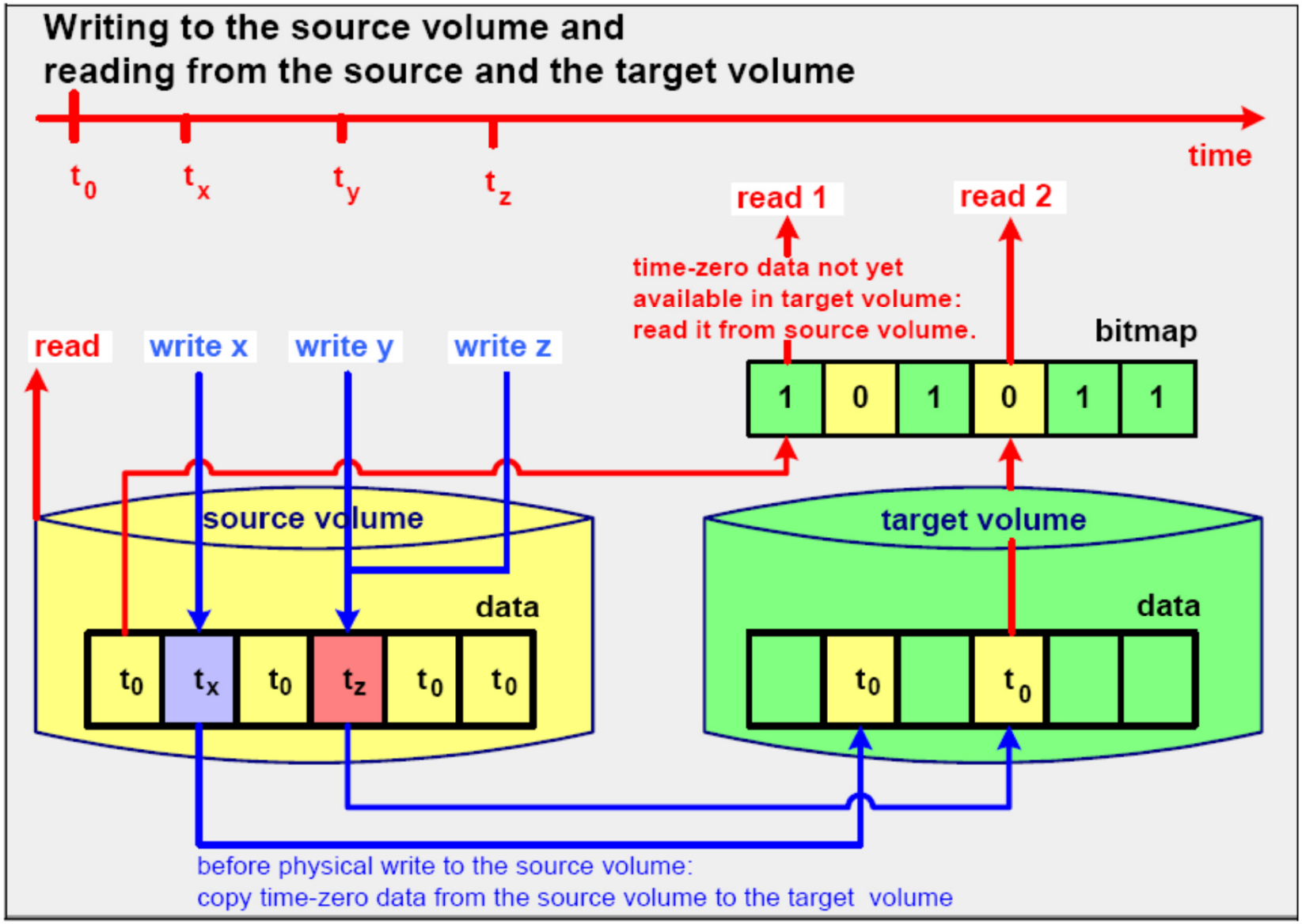


  - Menor rendimiento en escritura.
  - Protección contra fallos grabando paridad.
  - Recuperación automática ante fallo de un disco (degrada temporalmente el rendimiento).
  - Mejor aprovechamiento de disco.

Sistemas de copia en los servidores de disco

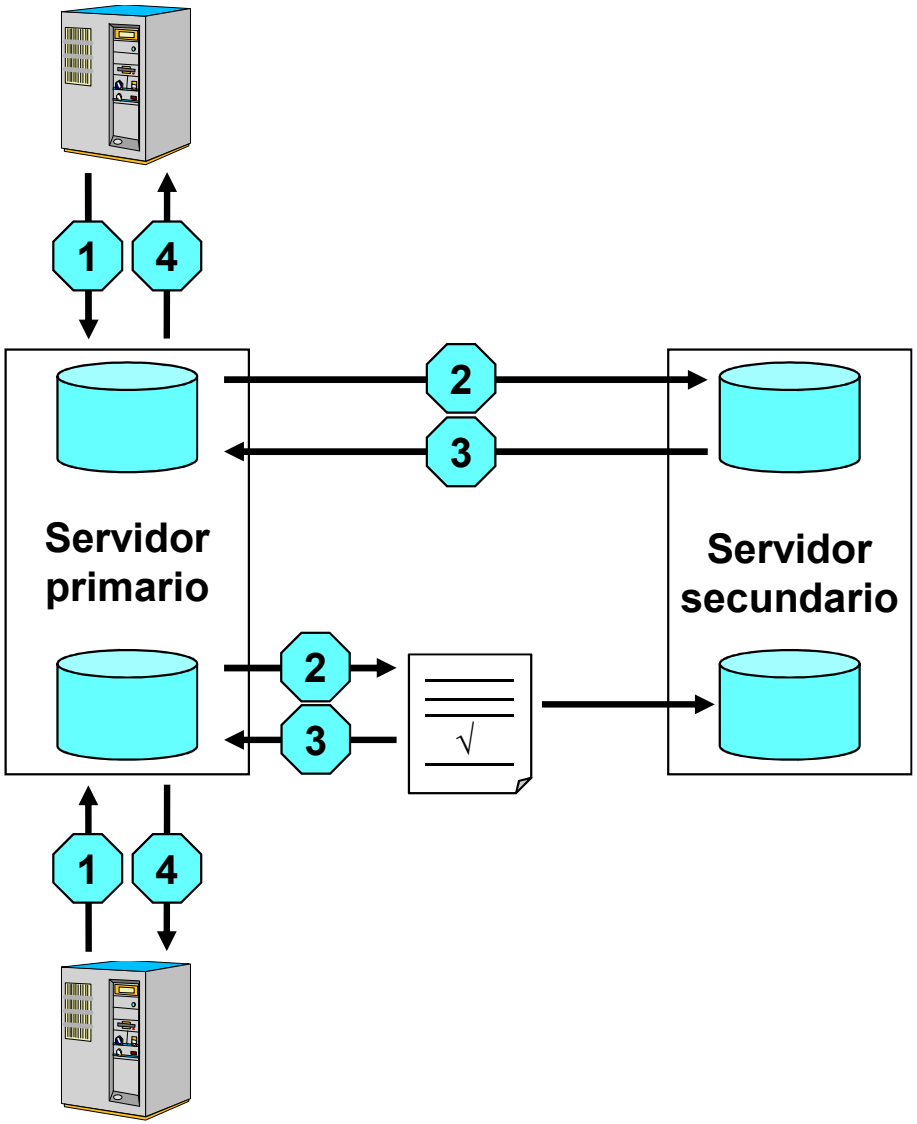
- Los servidores de disco proporcionan **herramientas autónomas** para garantizar la copia de la información, y garantizar así su disponibilidad para los servidores que la utilizan.
  - Se libera a los servidores de aplicaciones de la tarea de realización de las copias.
- Los principales tipos de copia son:
  - **Copias locales**:
    - **Clonación de discos**: copia de la información de una LUN a otra.
    - **Copias instantáneas *(Snapshots o Flash Copy)***:
      - Se realiza una copia de una LUN en el instante que se solicita.
        - » Se copian los punteros a los sectores que contienen la información.
      - En caso de actualización de un sector, se copia la información antigua y se actualiza el puntero de la copia instantánea.
      - Empleadas para generar una copia consistente de la información en un momento preciso para su uso posterior (por ejemplo, salvado a cinta).
  - **Copias remotas**:
    - **Copia síncrona.**
    - **Copia asíncrona.**

Copias instantáneas *(Snapshots o Flashcopy)*



Copias remotas

- **Copias síncronas:**
  - Actualizaciones de la información en el servidor primario se copian a discos del secundario inmediatamente.
  - La escritura en disco no termina hasta que no se ha registrado en los dos servidores.
  - La latencia de escritura en disco aumenta con la distancia entre los servidores.
  - Empleado para realizar **copias locales** que garanticen la alta disponibilidad.
- **Copias asíncronas**:
  - La escritura en disco termina al grabarse la información en el servidor primario.
  - Las actualizaciones se guardan en un *buffer*.
  - Se transmiten al disco secundario mediante un proceso en paralelo.
  - Necesario completar con procesos de sincronización *(check-point* o *go-to-sync).*
  - Empleado para realizar **copias remotas** que permitan la recuperación ante desastres.



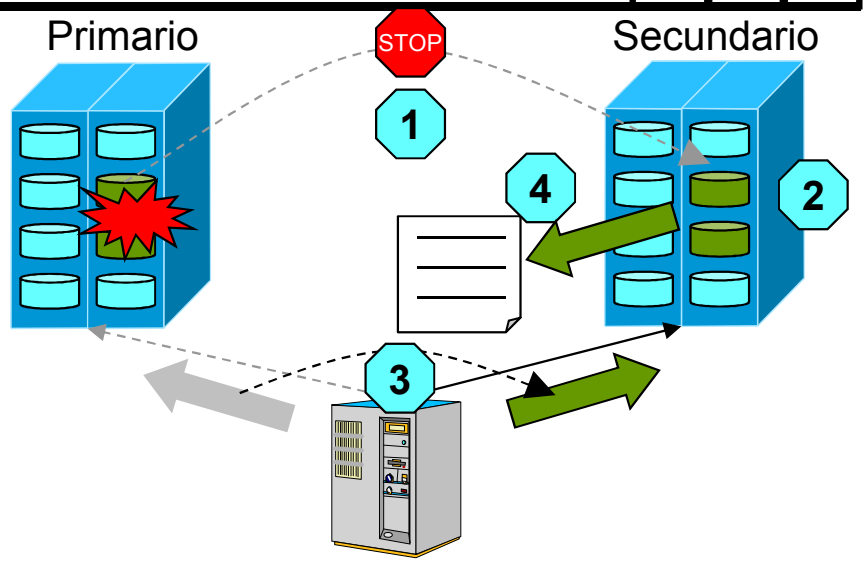
Detección de fallos en servidores de disco con copia remota

AS | E | I

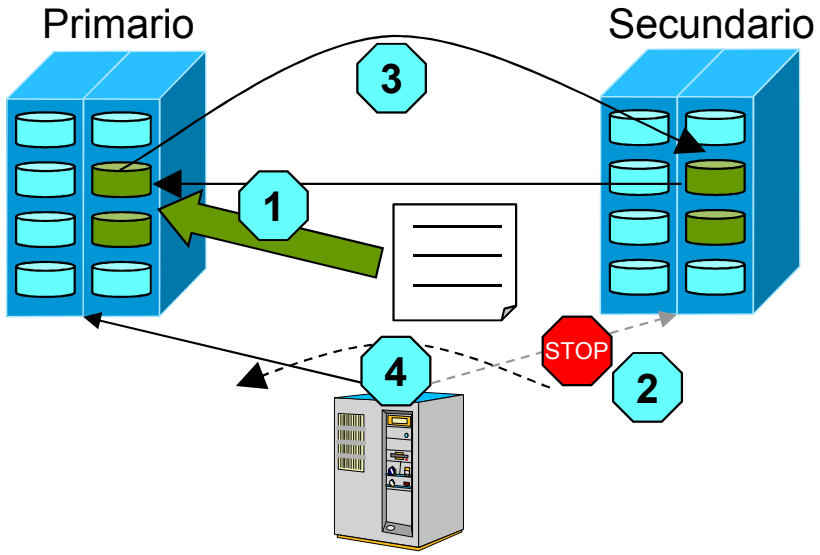
Averías en discos físicos individuales son resueltas por la arquitectura RAID *(fault tolerant)*.

Averías en LUN completa o un servidor de disco:

- ***Fail-Over***:
  1. Interrupción de la copia remota.
  2. LUN secundaria pasa a primaria.
  3. Se notifica al servidor de proceso que conmute a utilizar la LUN secundaria.
  4. Se comienzan a registrar los cambios.



- ***Fail-Back***: Tras recuperarse la LUN primaria:
  1. Se establece la copia remota del secundario activo al primario recuperado hasta su sincronización total.
  2. Se detiene el trabajo del servidor de proceso.
  3. Se reestablece la copia directa.
  4. Se notifica al servidor de proceso que conmute a utilizar la LUN primaria.



Redundancia en los sistemas de proceso (*failover clusters*)

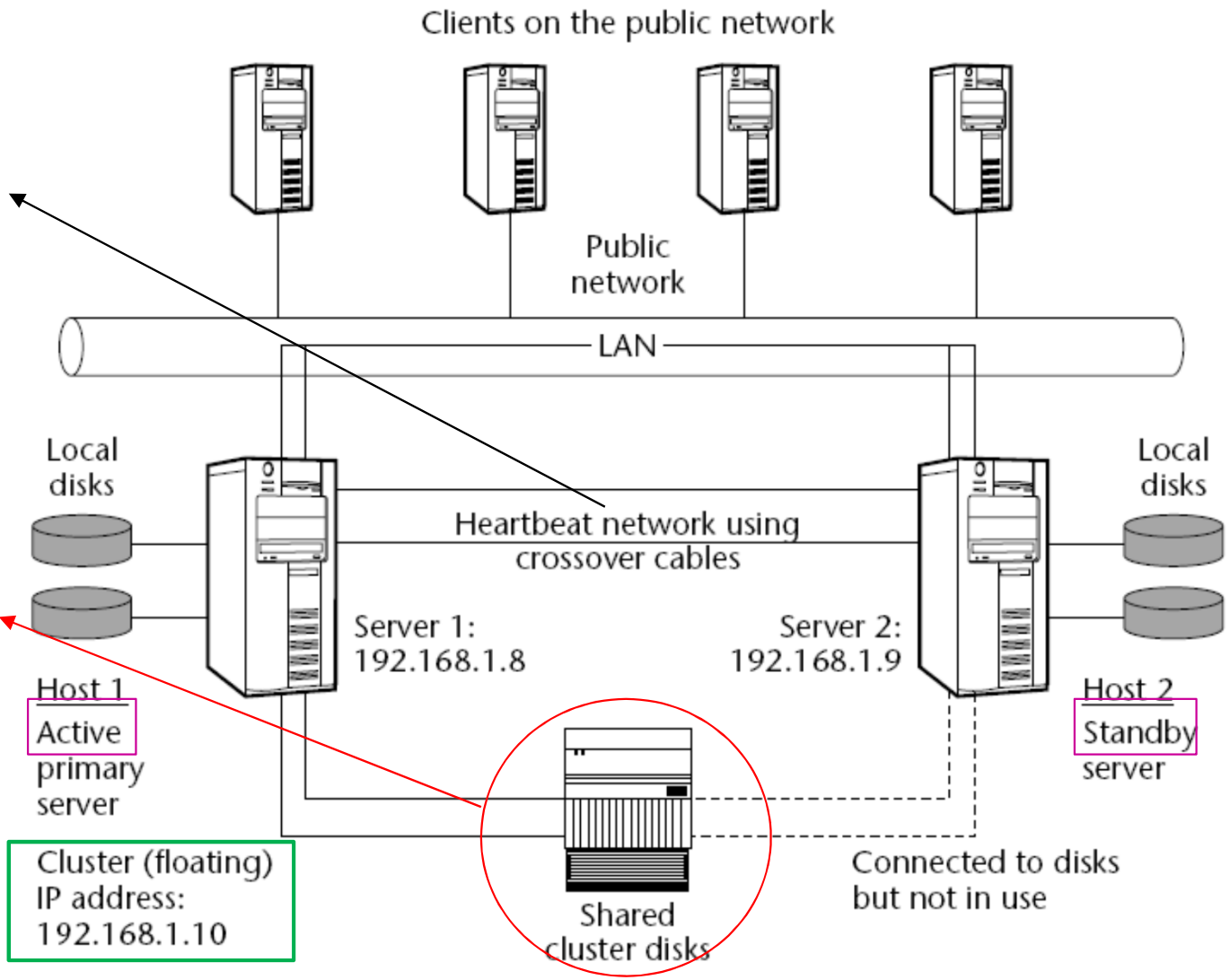
- La redundancia de los elementos de proceso (servidores) se consigue mediante su asociación en grupos que realizan una tarea común. Estos grupos se denominan *clusters*.
  - Conjunto **homogéneo** de ordenadores independientes que realizan un mismo proceso.
- Cada servidor ejecuta un determinado número de grupos de servicio *(**service groups**)*, compuestos por:
  - Una identidad de red: direcciones a las que el servicio responde.
  - Un conjunto de información almacenada en discos, físicos o LUNs.
  - Un conjunto de procesos.
- Cuando uno de los servidores de un *cluster* falla, los grupos de servicio que soporta deben ser trasladados a otro elemento del *cluster*.
  - Migración transparente, rápida, con intervención manual mínima y datos accesibles
- Según su modo de funcionamiento y gestión, hay diversos tipos de *clusters*:
  - ***Clusters* Activo-Pasivo o asimétricos**: Cada grupo de servicio se ejecuta en un nodo. Existen nodos de reserva en *stand-by* para ejecutarlos en caso de fallo.
  - ***Clusters* Activo-Pasivo cruzados o simétricos**: Cada grupo de servicio se ejecuta en un nodo. Todos los nodos ejecutan grupos de servicio. En caso de fallo de un nodo, el resto se hacen cargo de la ejecución de sus grupos de servicio.
  - ***Clusters* Activo-Activo**: Cada grupo de servicio se ejecuta en múltiples nodos. El fallo de uno de los nodos no supone la caída del grupo de servicio.

Cluster Activo-Pasivo de dos nodos

AS | E | I

- **Heartbeat network**: red que proporciona una comunicación fiable entre nodos de un clúster y no es accesible desde fuera del clúster.
- Los nodos del clúster usan esta red para monitorizar el estado de cada nodo y comunicarse entre ellos.
- Cada nodo manda un mensaje cada cierto tiempo para confirmar que está "vivo".

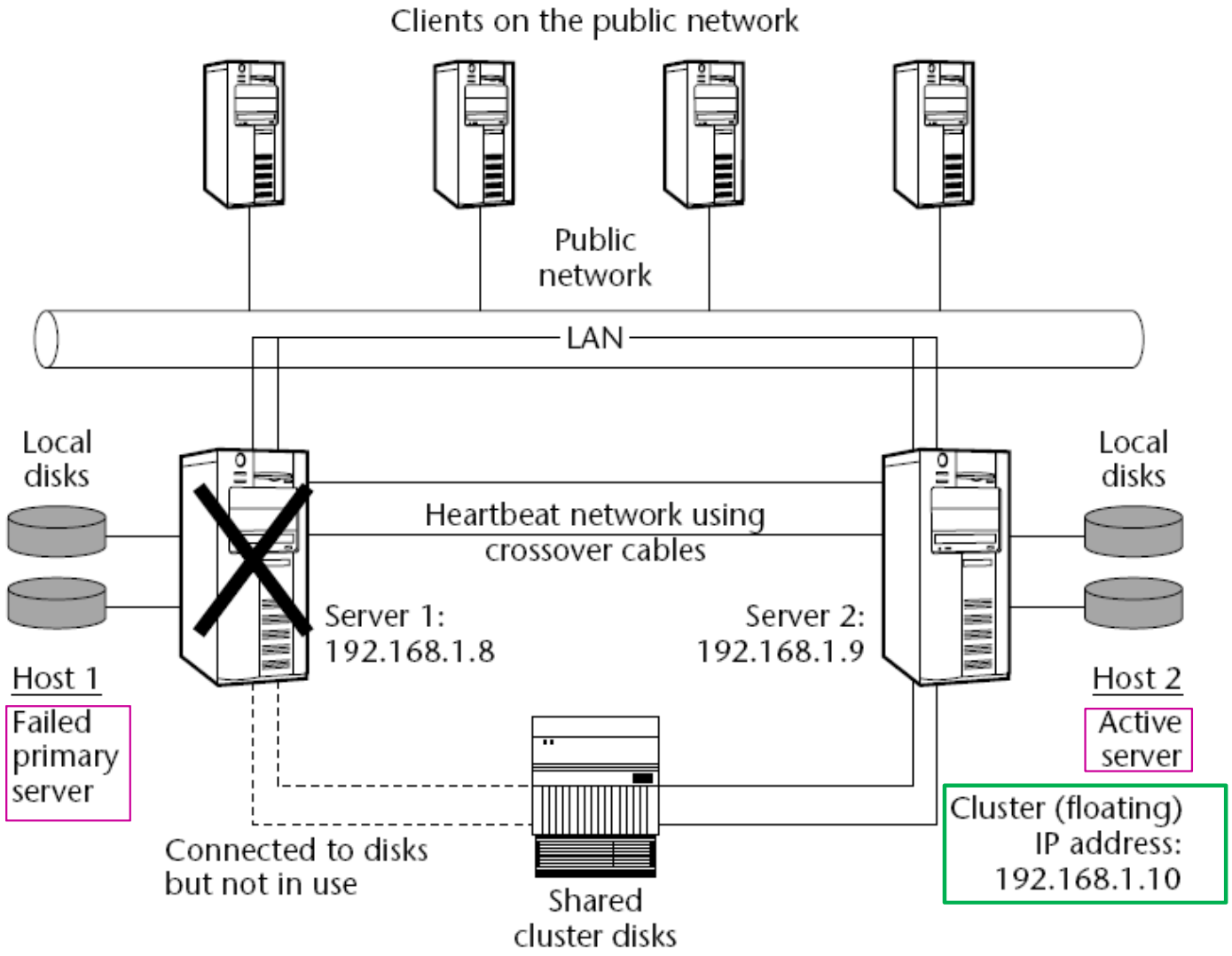
- **Dual-hosted disk**
- **Dos controladores separados**
- **Dos arrays de discos** separados con la misma información (mirroring)
- Un disco en particular solo puede ser accedido por un servidor simultáneamente (controlado por el software de administración del clúster).



Heartbeat

- **Es uno de los puntos críticos de un *cluster* de alta disponibilidad**. Si falla, se producirá la activación simultánea de ambos nodos, generando conflictos y caída del servicio.
- Mecanismos empleados para transportar los *heartbeat*:
  - Red de área local.
  - Escritura en una sección reservada de disco compartido.
- Modos de reforzar la fiabilidad de los *heartbeat*:
  - Eliminar SPOF en los mecanismos de transporte del *heartbeat*.
  - Emplear dos *heartbeat* (dos redes o red+disco). Actuar sólo en caso de fallo de ambos.
  - Emplear redes independientes lo más simples posibles: cable cruzado para dos nodos, o segmentos de LAN con *switches* independientes.
  - Vigilar el correcto funcionamiento de los procesos que generan y atienden los *hearbeats* en cada nodo.
    - Instalar un *watchdog* local que los rearranque en caso de caída.

Cluster Activo-Pasivo de dos nodos. Fallo del nodo activo



Problemas para el rearranque en el servidor secundario

- Si el cambio de la dirección IP se realiza sobre un adaptador con otra dirección MAC, la ***cache* ARP** de los clientes mantendrá la antigua hasta que detecten la nueva situación.
  - El cambio es más rápido si se asume también la antigua dirección MAC.
- Si la parada del servidor activo es **no planificada (fallo)**, habrá pérdida de información:
  - Datos en memoria en las aplicaciones.
  - Datos en memoria en el software de sistemas (*caches* de disco, por ejemplo).
  - Transacciones que se estaban realizando.
- Por estos motivos, el arranque de servicios en el servidor de reserva debe seguir los **procedimientos de arranque tras un cierre anormal**.
  - Revisión de la consistencia de los sistemas de archivos a nivel de sistema operativo.
  - Revisión de la consistencia de los datos en disco a nivel de aplicación (por ejemplo, bases de datos).
  - ***Rollback*** de las transacciones activas en el momento del fallo.
- El rearranque en este caso puede ser lento, pudiendo incluso requerir la reiniciación completa del servidor de reserva.
- En el caso del ***fail-back* este proceso siempre podrá ser planificado**, realizándose un cierre ordenado.
  - Pero también habrá una interrupción del servicio para realizarlo.

Cluster Activo-Pasivo cruzado o simétrico de dos nodos

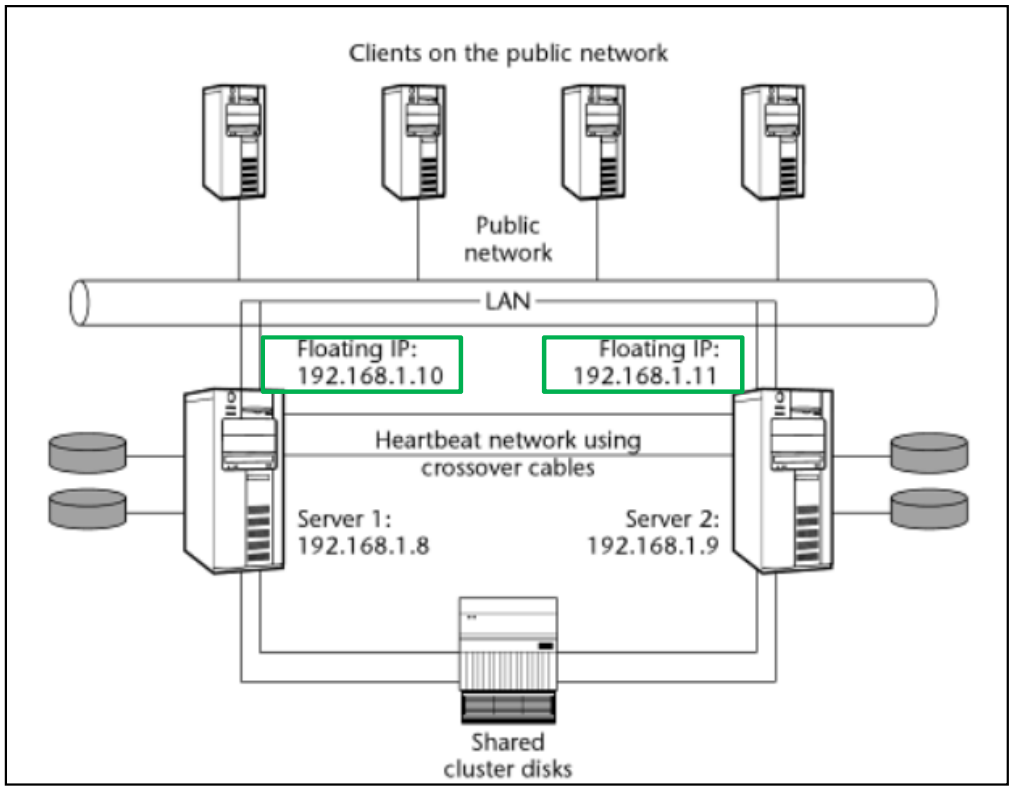
AS | E | I

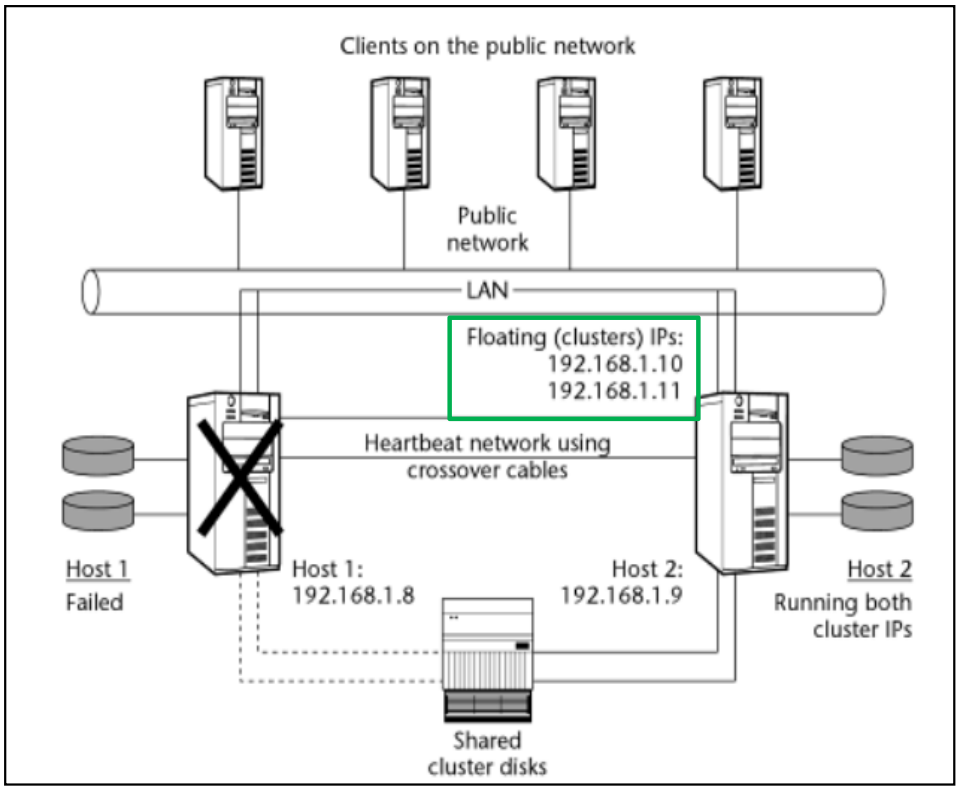
- El modelo es similar al anterior, pero con **grupos de servicios activos en ambos nodos**.
- **Cada nodo tiene direcciones de servicio fijas**. En caso de fallo de uno de los nodos, el superviviente ha de hacerse cargo de todas las direcciones.

- **Ventajas** frente al *cluster* Activo-Pasivo:
  - No hay ningún sistema parado. Mejor aprovechamiento *aparente* del coste.
    - Si no hay fallo, hay menos capacidad de proceso desaprovechada.
- **Inconvenientes**:
  - Tendencia a sobrecargar los nodos. En caso de fallo, puede ser que el nodo superviviente no tenga capacidad suficiente para absorber toda la carga.
    - Por tanto, la capacidad libre necesaria en el cluster para garantizar la operación sin degradación del servicio resulta ser igual en los dos tipos de *clusters*.
  - Las aplicaciones en ejecución pueden afectar a la capacidad del nodo superviviente para recuperar rápidamente los grupos de servicio del nodo que falló.
  - Imposibilidad de asignar dos direcciones MAC en el adaptador del nodo que asume la carga total en caso de caída.
    - Los clientes deben realizar un nuevo ARP para asociar la nueva dirección MAC a la IP conocida.

Cluster Activo-Pasivo cruzado o simétrico de dos nodos

AS | E | I





Otras arquitecturas de *clusters* Activo-Pasivo

AS | E | I

La aparición de las SAN, que permiten la compartición sencilla de discos por más de dos ordenadores, ha posibilitado la creación de arquitecturas de *clusters* más complejas:

- ***Clusters* N a 1**: Extensión del cluster Activo-Pasivo asimétrico a N servidores.
  - Ejecución de los grupos de servicio en N servidores.
  - Existe un servidor libre dedicado exclusivamente al respaldo de cualquiera de ellos.
    - Acoge los grupos de servicio de cualquier servidor del *cluster*.
  - Requiere una red redundante dedicada para *heartbeat* común a todos los servidores.
  - Mejor relación de coste por servicio, por usar un único servidor inactivo para N activos.
  - Menor capacidad para respaldar simultáneamente todos los grupos de servicio del *cluster*, pero igual capacidad de respuesta para un fallo simple de un nodo.
  - Mayor complejidad del software de gestión y control del *cluster*.
  - Variante: ***Clusters* N+1**. No hay asignación fija del servidor libre, sino que puede ser cualquiera. No se realiza *fail-back* tras la recuperación del servidor que falla.
- ***Clusters* N+M**: Similar al anterior pero con M ordenadores libres para recuperación.
- ***Clusters* 1 a N**: Los grupos de servicio que se ejecutan en un único nodo del cluster se reparten entre N nodos de menor capacidad que el original en caso de fallo.
- ***Clusters* N a N**: Extensión del cluster Activo-Pasivo cruzado o simétrico a N servidores.

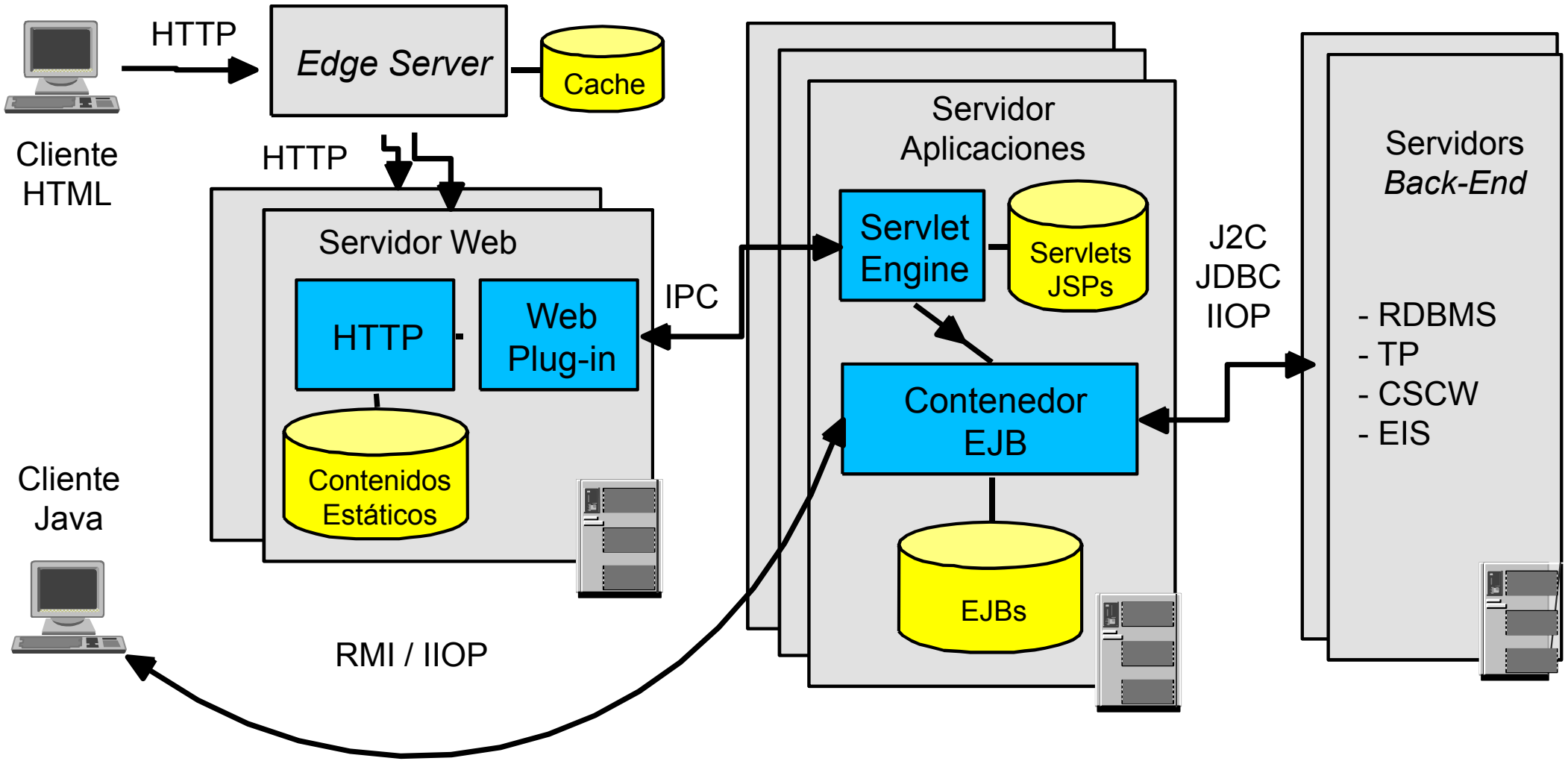
Cluster Activo-Activo

AA | D | C

- **Cada grupo de servicio se ejecuta en múltiples nodos**.
- Existe un mecanismo de **balanceo de la carga** para asignar las peticiones de los clientes a alguno de los nodos activos, que realiza el reparto de la carga y la verificación de la actividad.
- **El fallo de uno de los nodos no supone la caída del grupo de servicio.**
  - Dependiendo del tipo de aplicación, las conexiones activas en el momento del fallo pueden mantenerse también.
- Las **necesidades del *cluster*** dependen de las características de las aplicaciones, pero suelen incluir soluciones para resolver los siguientes problemas:
  - **Acceso compartido** a los discos en lectura y escritura.
  - Creación de áreas de memoria compartida entre los nodos del *cluster*. **Control del acceso concurrente** a las mismas.
  - **Consistencia de la información** distribuida entre los distintos nodos.
  - **Sincronización** temporal entre procesos en distintos nodos.
- Algunos **ejemplos**:
  - Aplicaciones desarrolladas con servidores de aplicaciones J2EE.
  - *IBM System Z Parallel Sysplex*.

Arquitectura de un servidor de aplicaciones J2EE (Presentada en el tema 2)

AA | D | C



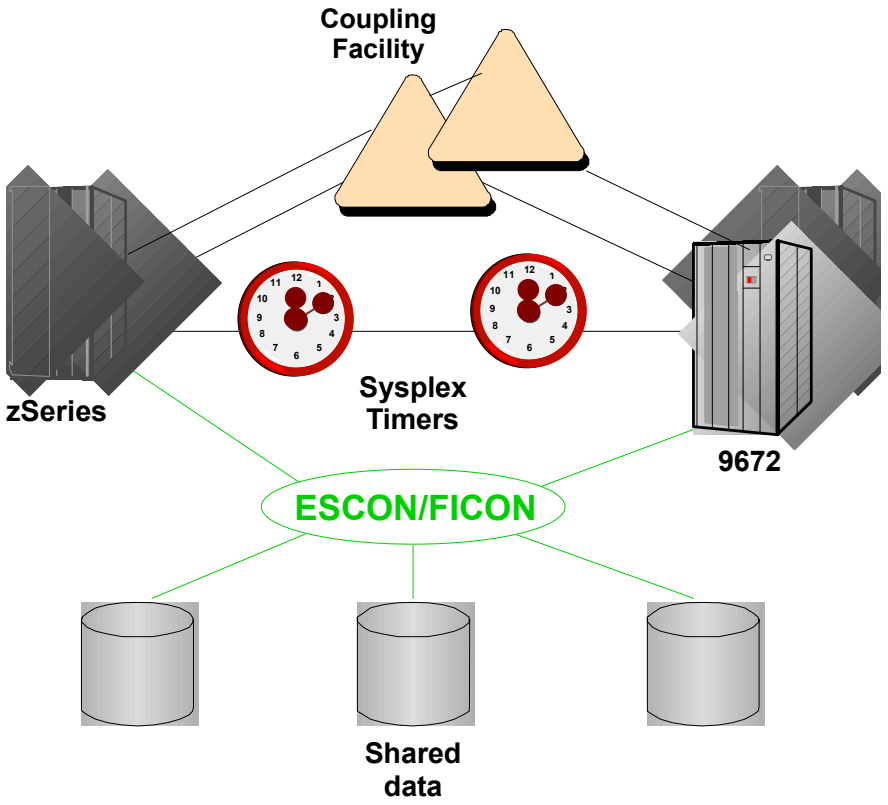
Cluster Activo-Activo en servidores de aplicaciones J2EE

AA | D | C

- Balanceo de carga entre servidores Web realizado por el *Edge Server*.
  - Si únicamente manejan contenidos estáticos, no necesita afinidad de sesión.
- Balanceo de carga entre *Servlet Engines* realizado por el Web Plug In.
  - Requiere afinidad de sesión proporcionada a través del identificador de sesión.
  - El identificador de sesión contiene el identificador de la *Servlet Engine* que inicia la sesión para facilitar el encaminamiento.
- Alta disponibilidad en el acceso a los contenedores EJB proporcionada por IIOP (Corba).
- Alta disponibilidad en el acceso a los servicios de *Back End* proporcionada por estos mediante mecanismos de *clustering*.
- El conjunto *Servlet Engine* + Contenedor EJB proporciona alta disponibilidad para las funciones que realiza mediante el empleo de un gestor de alta disponibilidad distribuido:
- Continuidad de la sesión entre distintas *Servlet Engines* mediante el mantenimiento común de los datos de sesión. El soporte común puede ser:
  - Tabla de sesiones en una base de datos común.
  - Gestor de réplica centralizado: Comunicación con las *Servlet Engines* para almacenar / recupearar los datos de sesión mediante IPC. Redundado para evitar SPOF.
  - Gestor de réplica distribuido: Los datos de sesión se transmiten mediante IPC a las *Servlet Engines* del *cluster*.

System Z Parallel Sysplex

- **Cluster Activo-Activo** en Mainframe de IBM
- Combinación de **HW y SW**.
- **Elementos**:
  - Nodos de proceso: Ordenadores Sistema Z.
  - *Sysplex Timers*: Proporcionan la sincronización temporal.
  - *Coupling Facility*: Unidad de memoria compartida conectada a los nodos de proceso mediante enlaces de alta velocidad *(Integrated Cluster Bus, ISC).*
  - Unidades de disco compartido en una SAN.
- **Proporciona**:
  - **Compartición** de datos con bloqueo.
  - **Distribución dinámica de la carga** entre los componentes del *cluster*.
  - **Sincronización temporal**.
  - **Conexión a alta velocidad** entre los componentes.
- Disponibilidad de 0.99999 (Paradas menores de 5.5 minutos / año)

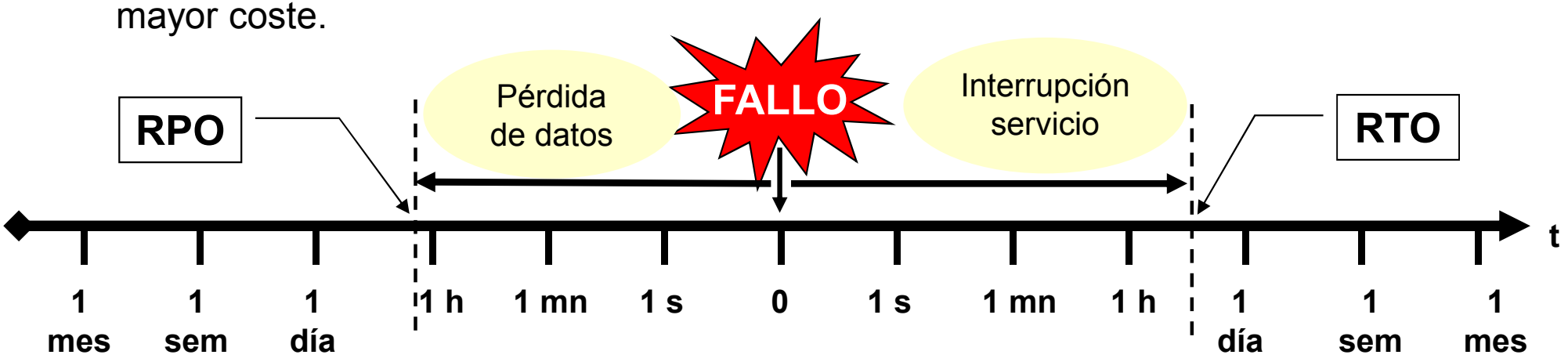


Recuperación ante desastres

- Un desastre es cualquier circunstancia que puede producir un fallo en la operativa del negocio de una empresa. Pueden tener múltiples causas:
  - Naturales: Huracanes, terremotos, inundaciones, incendios...
  - Humanos: Guerras, vandalismo, terrorismo...
  - Accidentes: Transportes, radiaciones, explosiones, derrumbamientos...
  - Infraestructuras: Fallos en comunicaciones, energía, suministros...
- **Aunque su probabilidad es muy baja, el impacto en el negocio es muy alto**.
  - En caso de ocurrir, pueden acabar con la actividad económica de la empresa.
  - Deben contemplarse como un riesgo que es necesario asumir o reducir.
- Contemplados en el **Plan de Continuidad de Negocio *(Business Continuity Plan, BCP)***, que incluye todas las áreas de actividad de la empresa.
  - Incluyen el **Plan de Continuidad de las Tecnologías de la Información y Comunicaciones (TIC).**
- Para garantizar la recuperación ante un desastre no basta con las técnicas de alta disponibilidad analizadas. Son necesarias nuevas técnicas que se presentan en este capítulo.

Parámetros de medida para la recuperación de desastres

- Para caracterizar el comportamiento de los sistemas TIC de cara a su recuperación ante desastres se emplean dos parámetros:
  - ***Recovery Point Objective (RPO)***: Instante del tiempo al que se es capaz de recuperar la información existente en el sistema tras un fallo.
    - Un RPO <> 0 implica que, en caso de fallo, habrá **pérdida de datos**.
  - ***Recovery Time Objective (RTO)***: Tiempo que se tarda en recuperar los sistemas tras producirse un fallo.
    - Un RTO <> 0 implica que, en caso de fallo, habrá **interrupción del servicio**.
- Los objetivos de RTO y RPO se deben fijar de acuerdo con las necesidades de negocio de cada servicio que se presta, y recogerse dentro de los SLAs.
  - Cuanto menor se desee que sean RTO y RPO, más compleja será la solución y de mayor coste.



Reglas básicas de diseño de arquitecturas tolerantes a desastres

- Proteger datos y nodos de proceso mediante **distribución geográfica**.
  - Cuanto más alejados se encuentren, mayor seguridad, pero mayor complejidad y coste.
- Generar **copias consistentes de la información**, garantizando la capacidad de recuperar los datos desde ellas.
  - **Copias *off-line***: *backups* en cinta.
  - **Copias *on-line***: réplicas directas de disco a disco.
- Proporcionar a los CPDs **redundancia en el suministro de energía**. Preferiblemente con acometidas independientes, a través de rutas alternativas de distribución y proporcionadas por distintas entidades suministradoras.
- Proporcionar a los CPDs **redundancia en el acceso de telecomunicaciones**. Con las mismas consideraciones que en el caso anterior.

Sistemas de creación de copias de la información

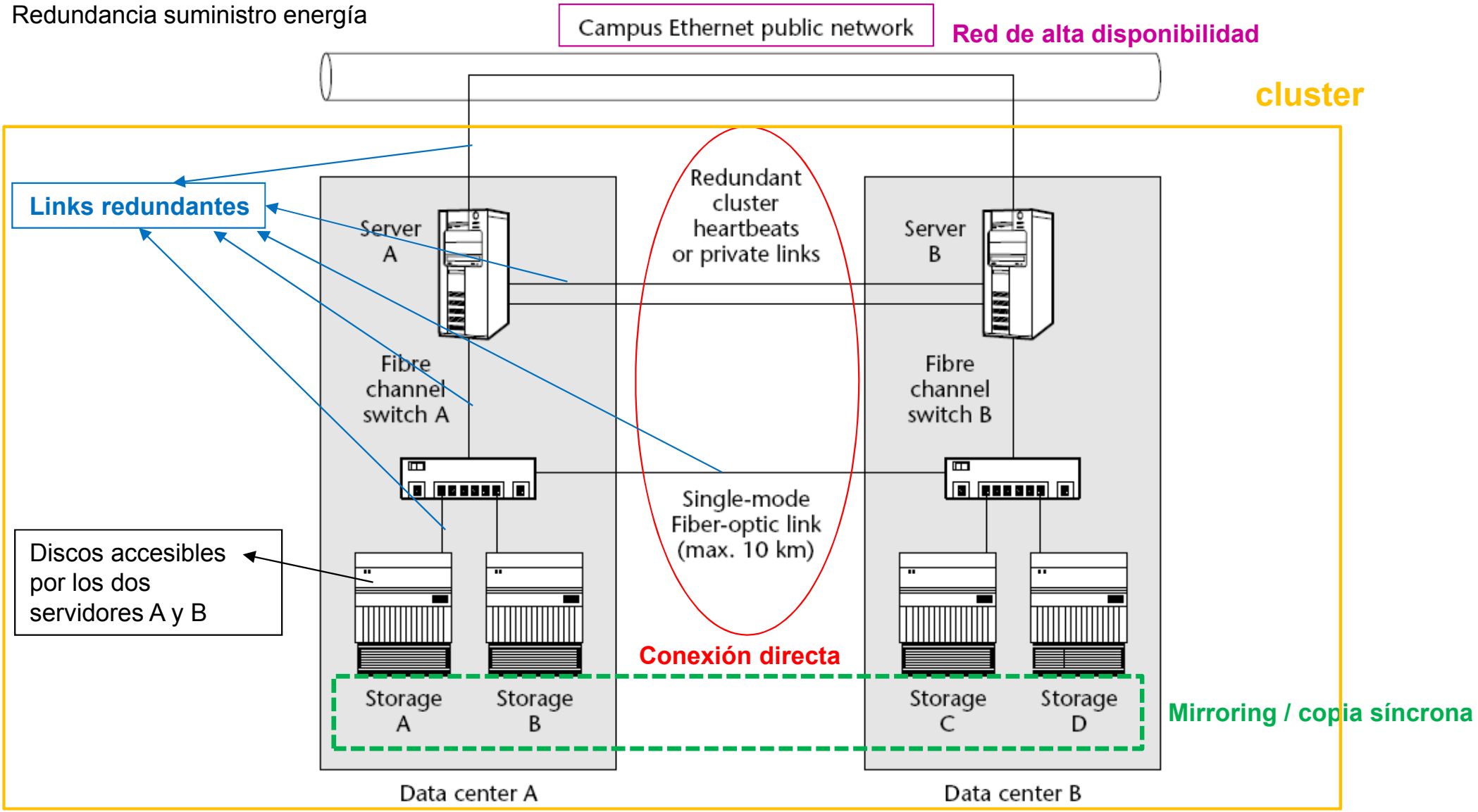
- ***Off-line (backups* en cinta):** Proporcionan un RPO elevado.
  - Deben realizarse sobre información consistente.
    - Requiere detener las aplicaciones mientras se realiza el *backup* o emplear técnicas de copia instantánea de la información.
  - Debe probarse que la información se restaura correctamente.

- ***On-line* (réplicas de disco):** Proporcionan un RPO reducido. Mecanismos más habituales, ordenados de menor a mayor fiabilidad:
  - ***Scripts* de usuario**. Automatizados por planificador.
  - **Réplicas basadas en *software*** en los sistemas de proceso, a distintos niveles:
    - Transacciones múltiples: ejecución en dos sistemas distintos.
    - Réplicas de bases de datos: *staging* de escrituras en servidor primario que se transmite y aplica en el servidor secundario.
    - Copia de escrituras físicas en disco transaccionales, log de escrituras...
  - **Réplicas basadas en nivel de *driver***: *mirroring* remoto.
  - **Réplicas basadas en el hardware de los servidores de disco**: síncronas y asíncronas, estudiadas previamente.

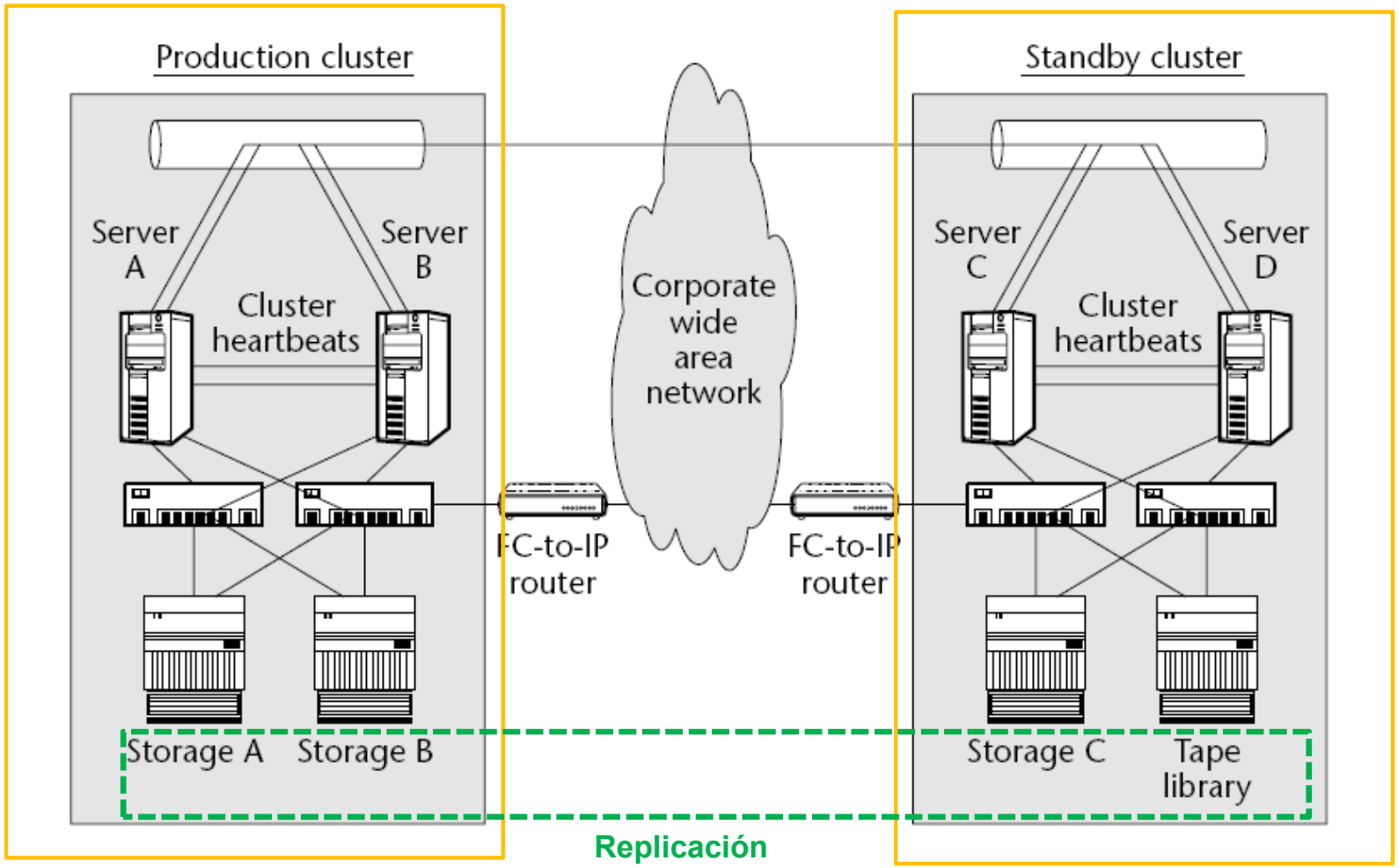
Arquitecturas de CPDs orientadas a la recuperación ante desastres

- ***Clusters* a nivel de campus.**
  - Los elementos del *cluster* se distribuyen entre dos edificios próximos.
  - Extensión de LAN y SAN entre ambos edificios, normalmente con cableado propio.
  - Es posible la copia síncrona de información entre servidores de disco.
  - El funcionamiento lógico es como un *cluster* local: Proporciona alta disponibilidad.
  - Protege frente a desastres que ocurran en un único edificio.
- ***Clusters* metropolitanos.**
  - Similar al anterior, pero con edificios en ubicaciones distantes que no excedan el rango en el que es posible realizar copia síncrona de la información.
  - Extensión de LAN y SAN con enlaces alquilados de alta velocidad (“fibra oscura”, WDM).
  - Protege frente a desastres locales.
- ***Clusters* continentales.**
  - Sin límite de distancia entre centros.
  - Requieren conexión a través de red WAN.
  - Protege frente a desastres en áreas extensas (región o estado), según la distancia entre centros.
- **Soluciones mixtas.**
  - *Cluster* local o metropolitano para garantizar alta disponibilidad, añadiendo un *cluster* continental para garantizar recuperación ante desastres.

Clusters de campus y metropolitanos

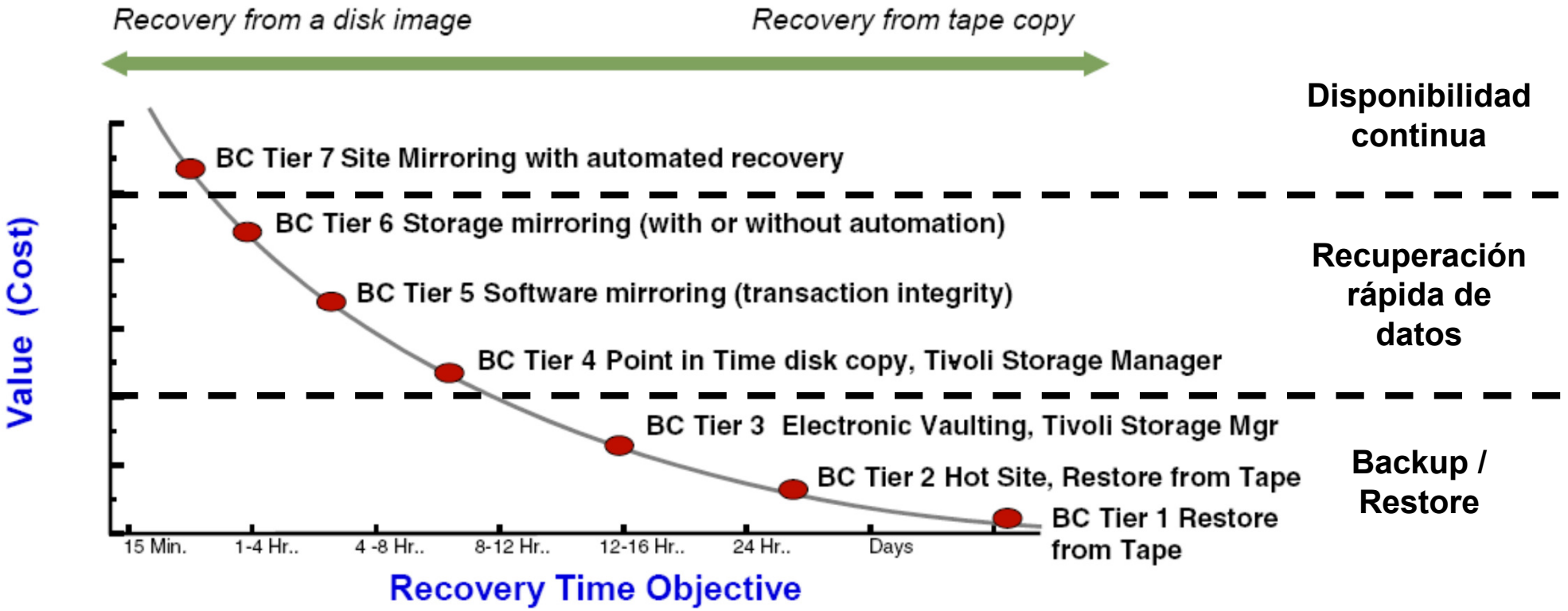


Clusters continentales



Los niveles en la continuidad del negocio

- Según SHARE e IBM, se distinguen **siete niveles distintos** dentro de los sistemas que permiten garantizar la continuidad de negocio IT, que se agrupan en **tres segmentos**:



Bibliografía especial del tema

- ARREGOCES, M. y PORTOLANI, M., *Data Center Fundamentals*, Indianapolis (USA), Cisco Press, 2004.
- Cisco, *Internetworking Technology Handbook*, http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/index.htm.
- JAYASWAL, K., *Administering Data Centers*, Indianapolis (USA), Wiley, 2006.
- KOPPARAPPU, C., *Load Balancing Servers, Firewalls, and Caches*, New York, Wiley, 2002.
- LEON-GARCÍA, A, *Probability and Random Processes for Electrical Engineering*, Reading (MA), Addison-Wesley, 1994.
- MARCUS, E. y STERN, H., *Blueprints for High Availability*, Wiley, 2003. 2ª ed.
- MUSA, J.D., IANNINO, A. y OKUMOTO, K., *Software Reliability. Measurement, Prediction, Application,* New York, McGraw-Hill, 1987.
- PAPOULIS, A. y PILLAI, U., *Probability, Random Variables and Stocastic Processes,* McGraw-Hill, 2002. 4ª ed.
- Reliasoft, *System Reliability Theory & Principles Reference*, http://www.weibull.com/systemrelwebcontents.htm.