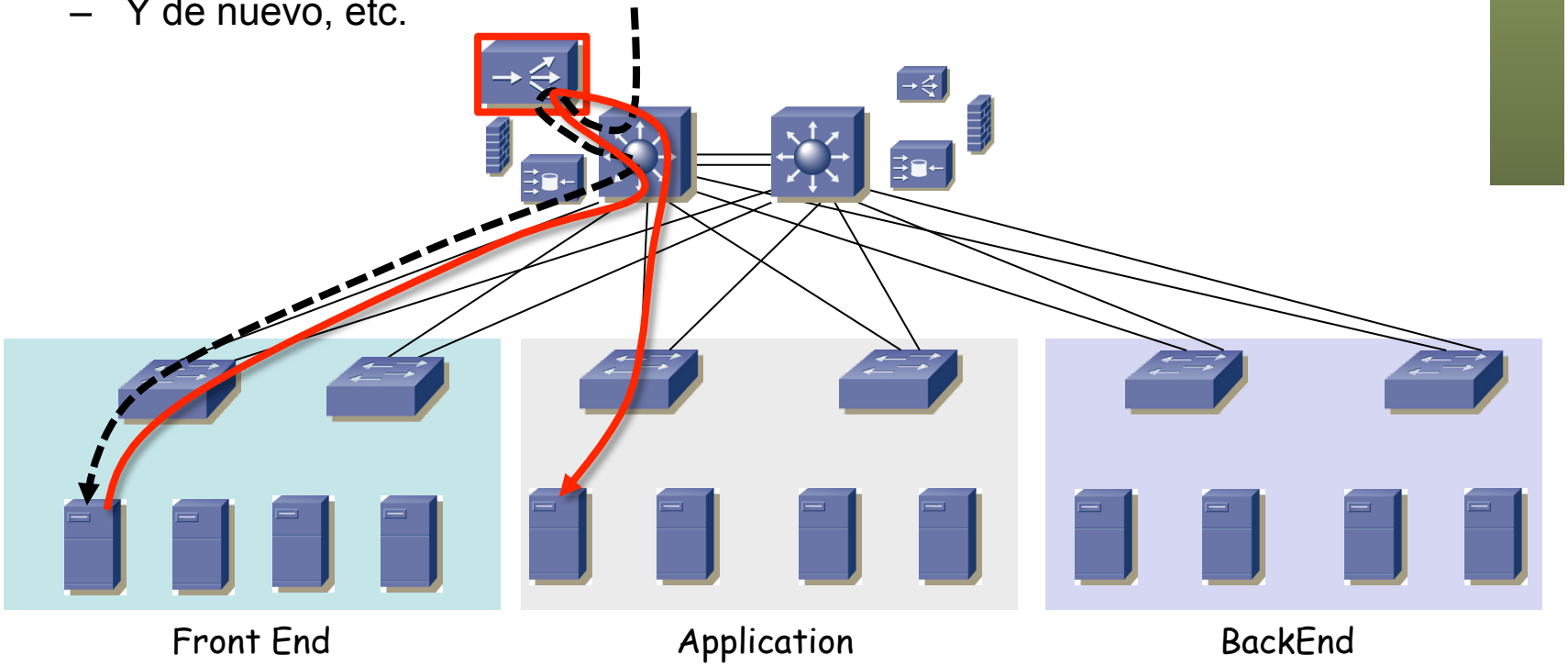
Ubicación de los servicios

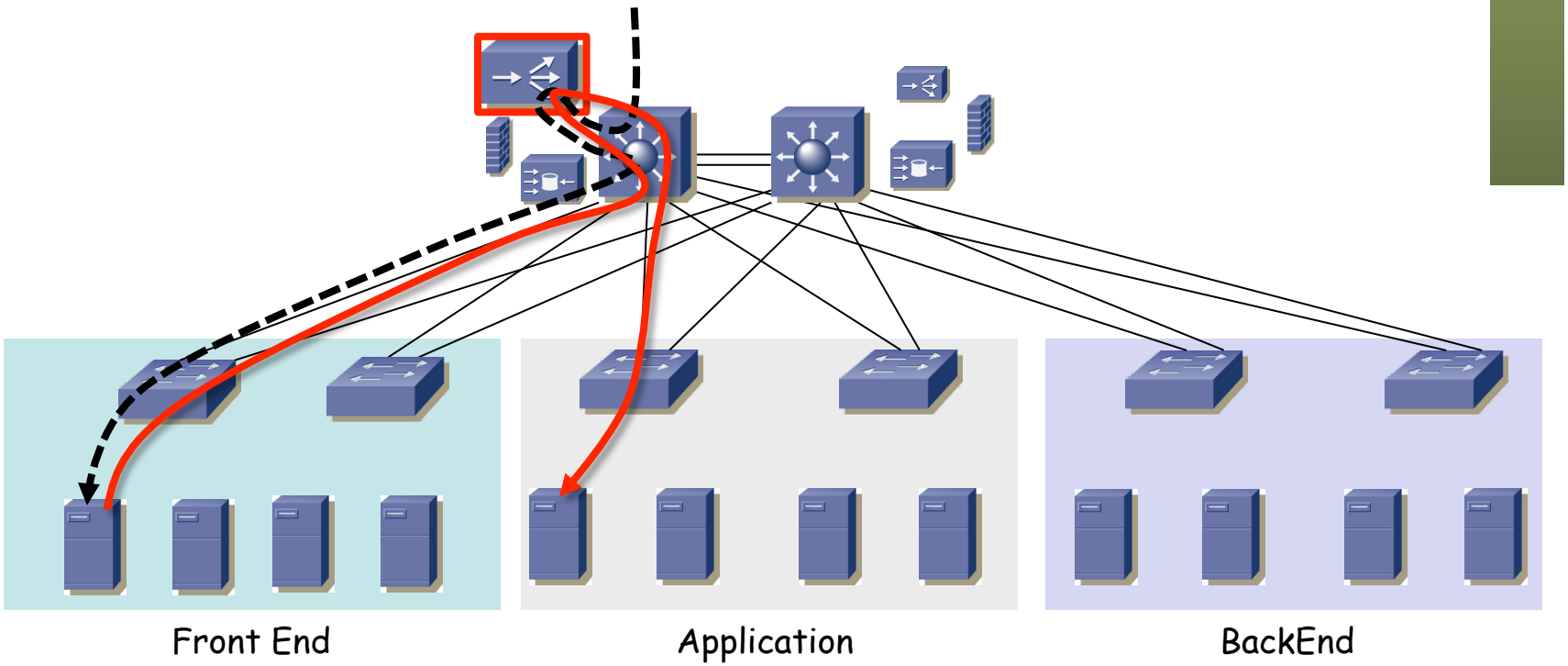
Servicios: ¿Dónde?

- Es común que los *tiers* estén en la capa de acceso
- Con un diseño colapsado los servicios estarían conectados a los conmutadores de agregación
- O pueden ser módulos en los conmutadores de agregación
- Pueden ser compartidos entre las diferentes capas
- Por ejemplo el mismo balanceador
  - Pasa por el balanceador de camino al *front end*
  - Y de nuevo, etc.



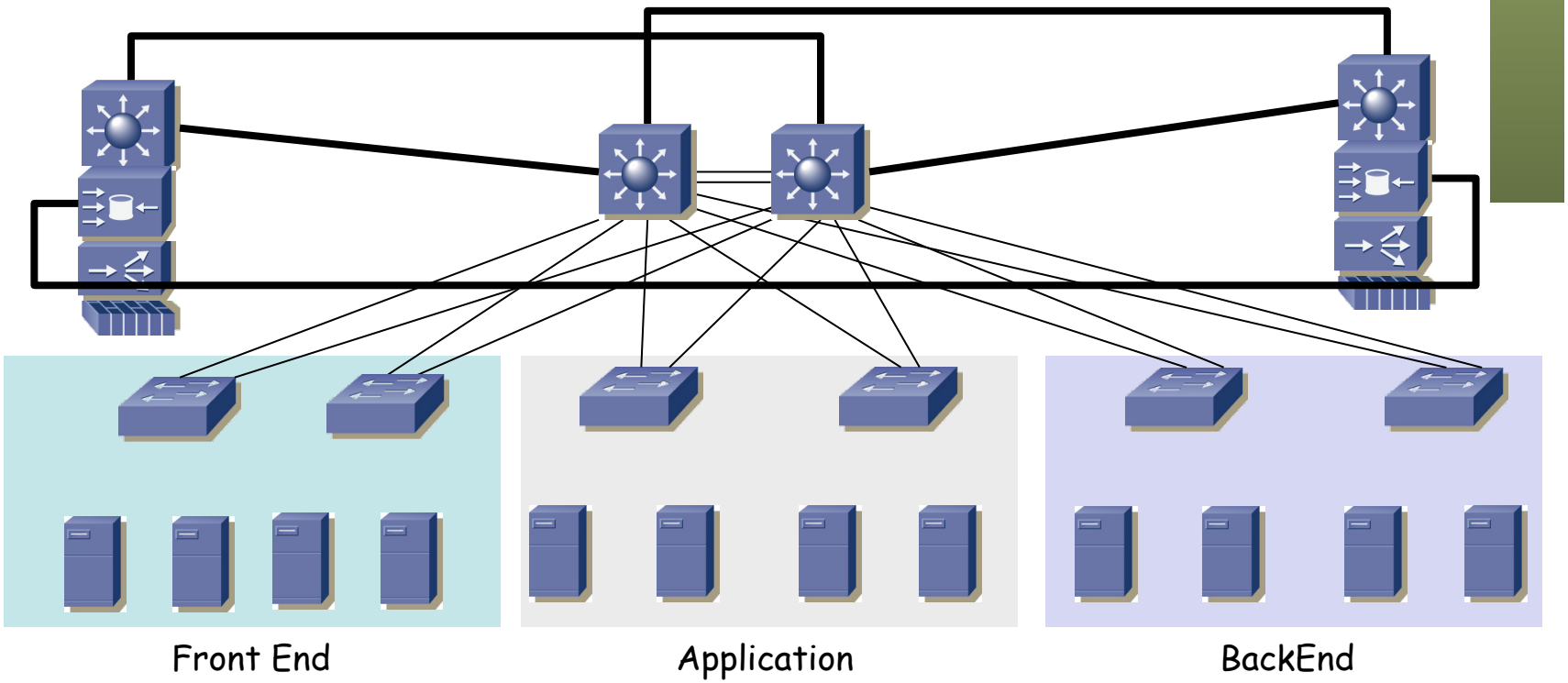
Servicios: ¿Dónde?

- Esto puede ser gracias a que tenga varios interfaces físicos, en las diferentes VLANs
- Porque emplee trunking en su(s) interfaz(-ces)
- Puede incluso dividirse en varios balanceadores “virtuales”
- Compartirlos reduce costes pero aumenta la complejidad y requiere mayor rendimiento de los mismos



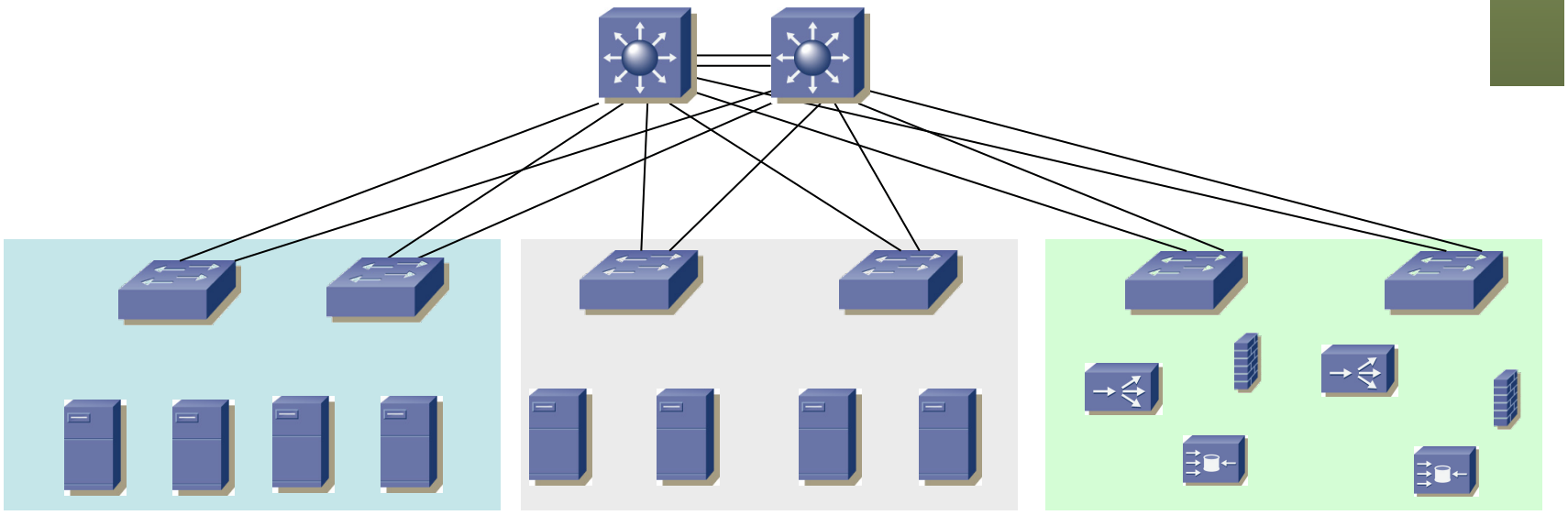
Servicios: ¿Dónde?

- Los equipos pueden ser demasiados para los slots de los conmutadores de agregación
- Demasiados para los puertos de los conmutadores de agregación
- Podemos sacarlos a sus propios conmutadores (*service switches*)
- (…)



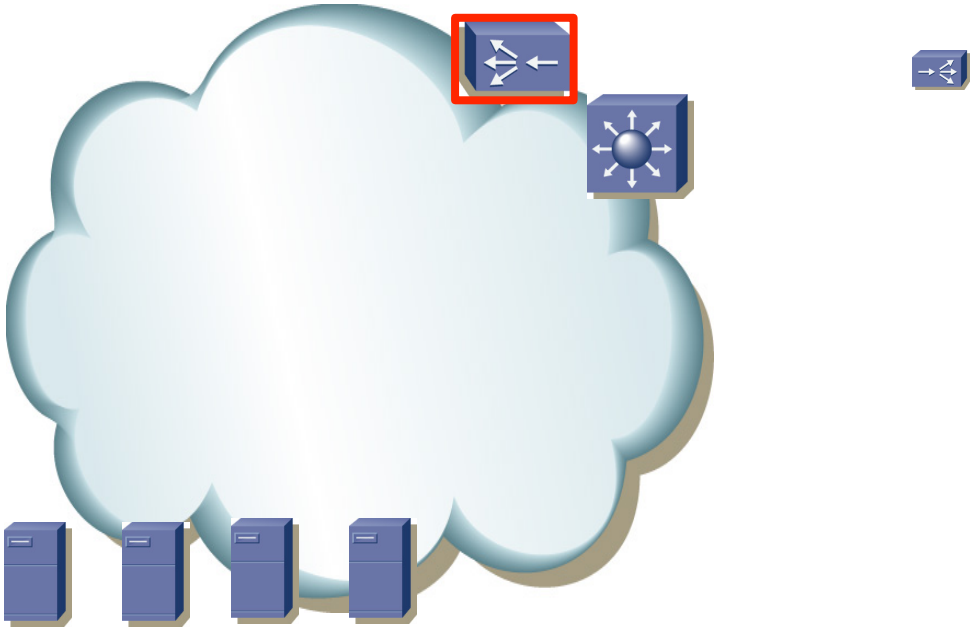
Servicios: ¿Dónde?

- Los equipos pueden ser demasiados para los slots de los conmutadores de agregación
- Demasiados para los puertos de los conmutadores de agregación
- Podemos sacarlos a sus propios conmutadores
- O sacarlos de la capa de agregación a su propia capa de acceso
- Especialmente necesario si son múltiples equipos balanceados



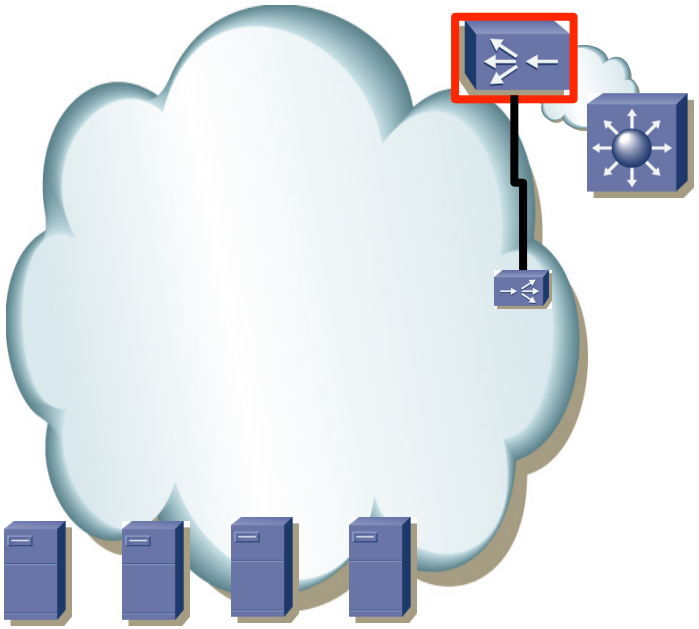
Servicios: ¿ y lógicamente?

- Lo más normal es que estos dispositivos tengan que ser adyacentes en capa 2 a los servidores
- Frecuentemente son el router por defecto de los mismos (…)



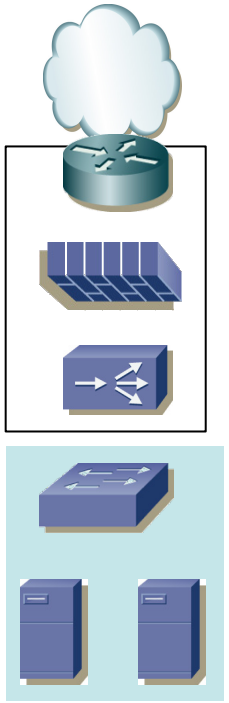
Servicios: ¿ y lógicamente?

- ¿Y el segundo equipo? También, pues debe sustituirle
- Tendrán un protocolo para implementar la redundancia
- Los *heartbeats* frecuentemente no son enrutables
- Suelen ser protocolos propietarios pero cuando son también router por defecto pueden emplear el FHRP
- También pueden tener una VLAN entre ellos o enlace directo para mantener sincronizada información de estado (*stateful failover*)



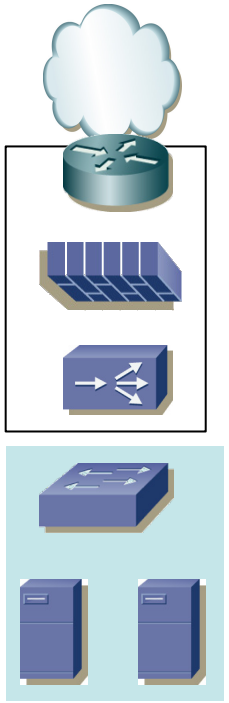
Orden de los servicios

- Recordemos que algunos de los servicios pueden ser módulos en un router/switch
- Tendremos diferentes formas de ordenarlos en el camino hacia los servidores
- Cada forma tendrá ventajas e inconvenientes
- (…)



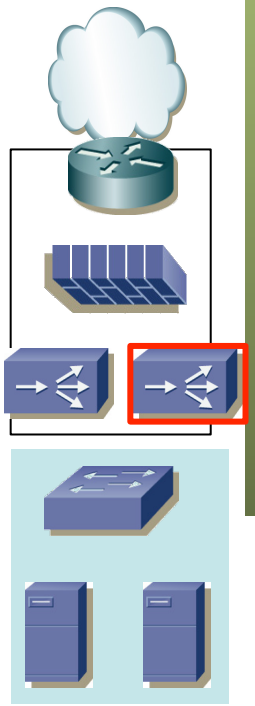
Router-Firewall-Balanceador

- Desde el núcleo, podemos encontrarnos primero con el router
- A continuación el firewall
- Finalmente el balanceador
- Si el balanceador se comporta como un puente entonces el router por defecto será el firewall
- Si el balanceador se comporta como un router se vuelve el router por defecto para los servidores



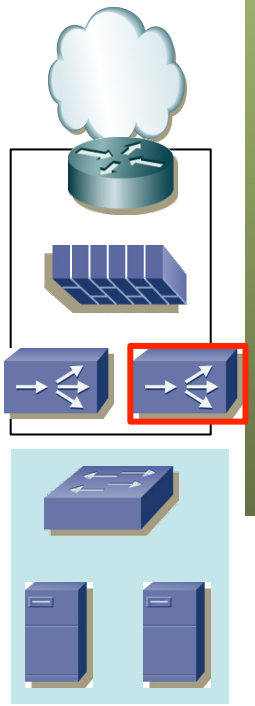
Router-Firewall-Balanceador

- Entre los elementos redundados estará el balanceador
- Si es el router por defecto puede emplear el FHRP
- Pueden configurarse en activo-pasivo o activo-activo
- Activo-pasivo (*active-standby*)
  - Uno de ellos hace todo el trabajo y si falla entra el otro
  - Mantener el estado sincronizado es sencillo (un solo sentido)
  - Es la alternativa más simple
- (…)



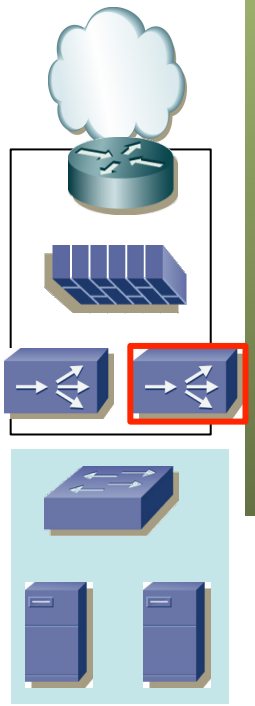
Router-Firewall-Balanceador

- Entre los elementos redundados estará el balanceador
- Si es el router por defecto puede emplear el FHRP
- Pueden configurarse en activo-pasivo o activo-activo
- Activo-pasivo (*active-standby*)
- Activo-activo (*active-active*) con reparto de VIPs
  - Las direcciones virtuales de los servicios se reparten
  - Cada dirección es empleada por un balanceador y el otro es el de respaldo
  - Es como empleado 2 grupos VRRP en la subred
  - Los servidores tendrán de router por defecto al balanceador que gestione como primario la dirección IP de su servicio
- (…)



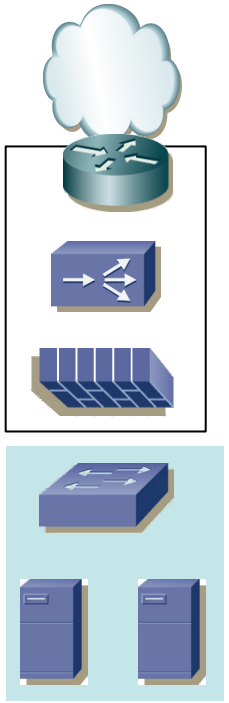
Router-Firewall-Balanceador

- Entre los elementos redundados estará el balanceador
- Si es el router por defecto puede emplear el FHRP
- Pueden configurarse en activo-pasivo o activo-activo
- Activo-pasivo (*active-standby*)
- Activo-activo (*active-active*) con reparto de VIPs
- Activo-activo con VIPs replicadas
  - Las direcciones IP de los servicios están activas en los dos
  - Hay que conseguir que el mismo cliente (toda su sesión) vaya siempre al mismo equipo
  - Esto es complejo pues los equipos *upstream* son conmutadores capa 2 y/o 3 que no entienden de sesiones
  - Normalmente eso requiere repartir a los clientes entre las dos instancias de la dirección IP virtual



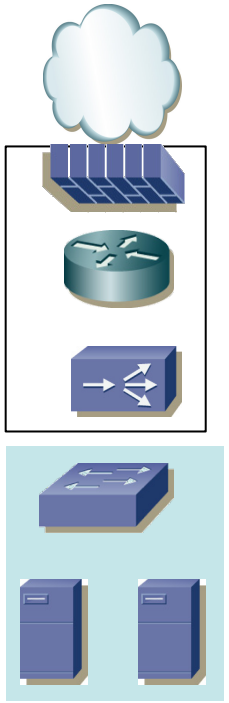
Router-Balanceador-Firewall

- En este caso tras el router está el balanceador y detrás el firewall
- El firewall debe permitir que el balanceador verifique el estado de los servidores (*health probes*)
- Esto implica configuración
- En esta configuración el router por defecto es el firewall



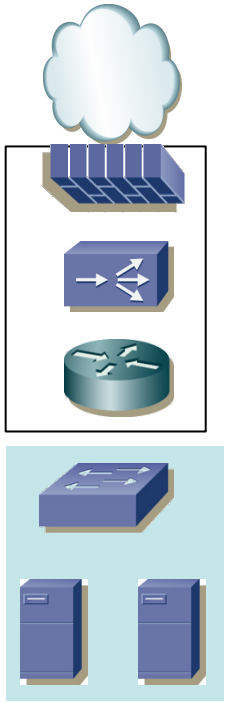
Firewall-Router-Balanceador

- En este caso la entrada es por el firewall
- Es probable que requiera funcionalidades extra de router como por ejemplo integrarse en el IGP
- Es más difícil securizar cada *tier* pues están todos al otro lado del firewall, enrutados sin pasar por el fw
- Según cómo opere el balanceador el router por defecto es él o el router



Firewall-Balanceador-Router

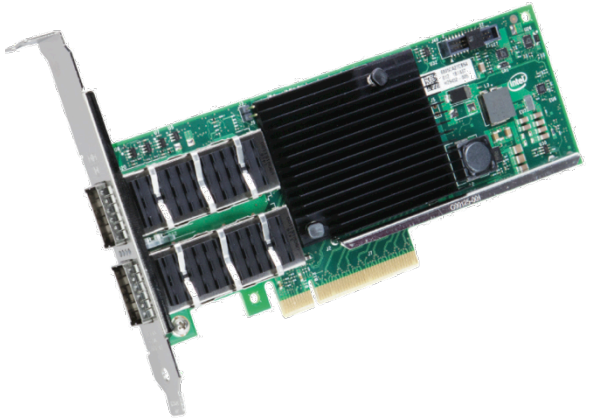
- El router como router por defecto para los servidores
- Eso permite usar funcionalidades habituales suyas como un FHRP, QoS, relay DHCP, etc.
- El balanceador no puede emplear una técnica que le requiera conectividad L2 con los servidores
- Los *health probes* que envíe el balanceador deben ser enrutables
- De nuevo pasar por el firewall entre cada capa requiere volver upstream



NICs Ethernet para servidor

Tareas en la NIC

- Por un enlace 10GE pueden llegar en 1 segundo más de 14 millones de tramas de 64 bytes
- Eso da a la CPU unos 67ns para procesar cada una
- Las CPUs tienen serios problemas para procesar en ese tiempo cabeceras TCP/IP
- Una NIC puede incluir electrónica para llevar a cabo ciertas tareas de TCP/IP descargando a la CPU
- La NIC puede incluir ASICs, Network Processors o un procesador con un sistema operativo de tiempo real
- A 400Gbps una trama cada 1,67ns lo cual está en el rango de los mejores tiempos de acceso a memoria

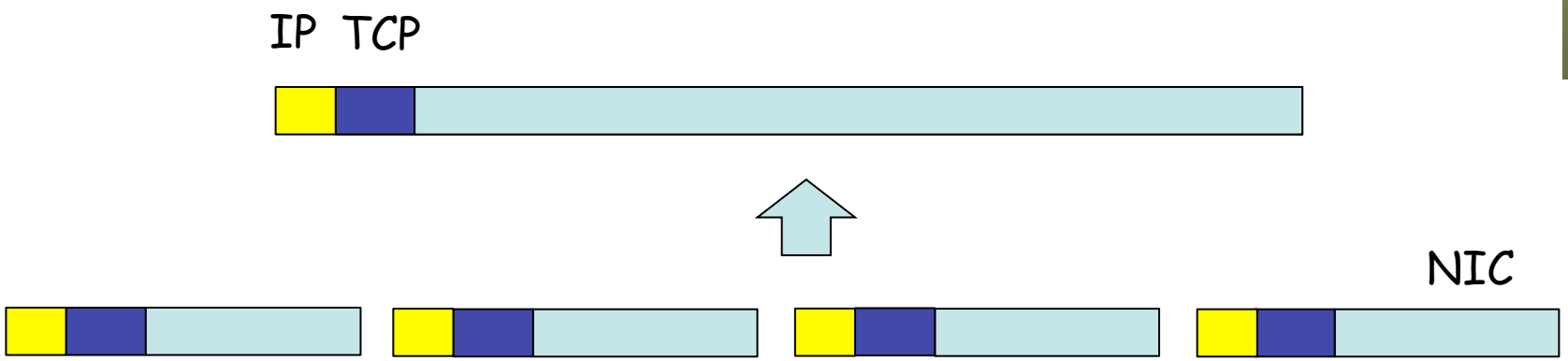


Integración en el bus

- DMA
  - *Direct Memory Access*
  - Transferencia desde la NIC a memoria sin requerir a la CPU
- Coalescencia de interrupciones
  - Las NICs solían generar una interrupción por paquete
  - Alto coste para la CPU
  - Por ejemplo los mainframes tienen CPUs dedicadas a atender I/O
  - La coalescencia hace que la NIC genere una interrupción para un grupo de paquetes en vez de por cada uno

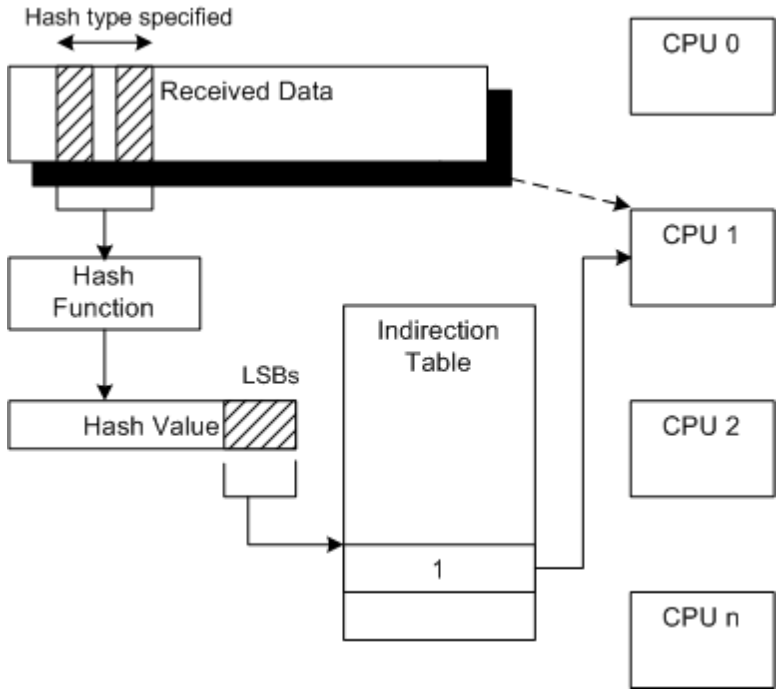
LRO

- Large Receive Offload
- La NIC une varios segmentos TCP en uno solo
- Crea unas cabeceras TCP e IP para ese nuevo segmento
- Reduce el número de interrupciones y procesado de cabeceras en el kernel



RSS

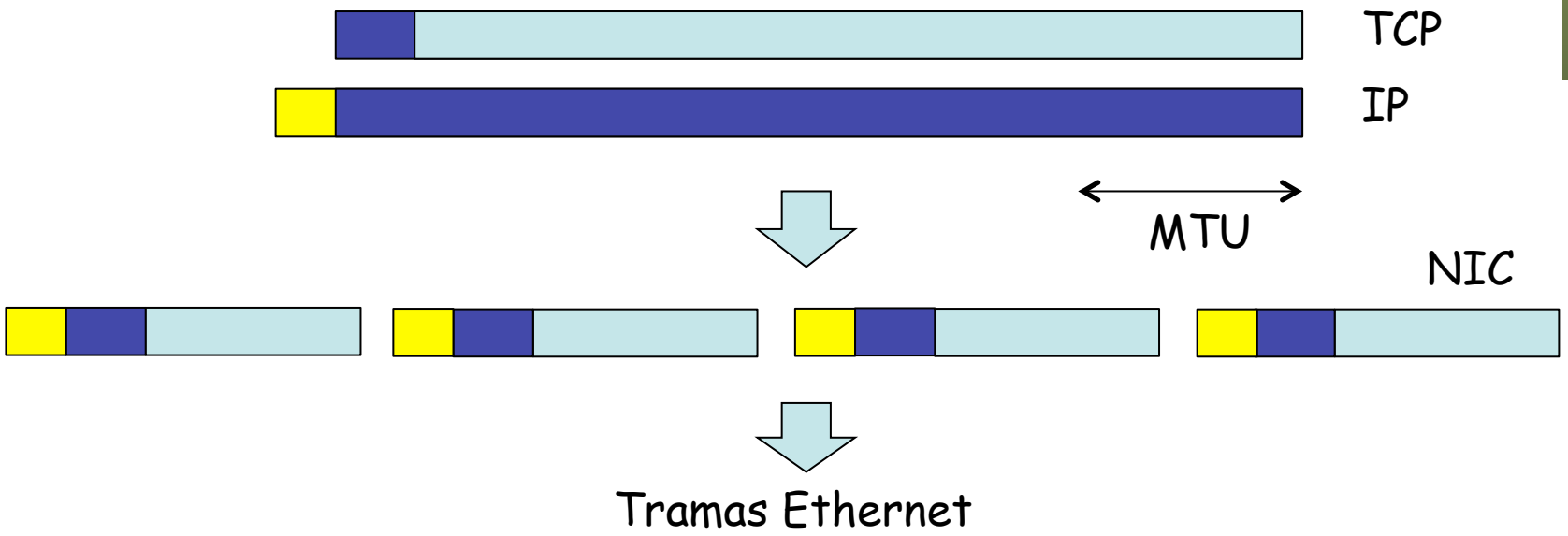
- Receive Side Scaling
- NIC calcula un hash sobre el paquete recibido y con él decide a qué CPU manda la interrupción
- Permite paralelizar entre varias CPUs el procesado del tráfico recibido



https://msdn.microsoft.com/library/windows/hardware/ff556942(v=vs.85).aspx

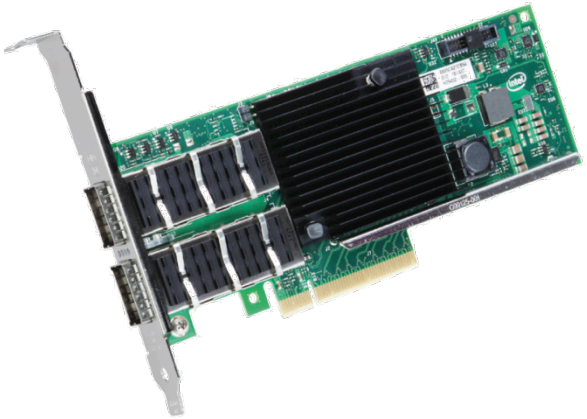
LSO

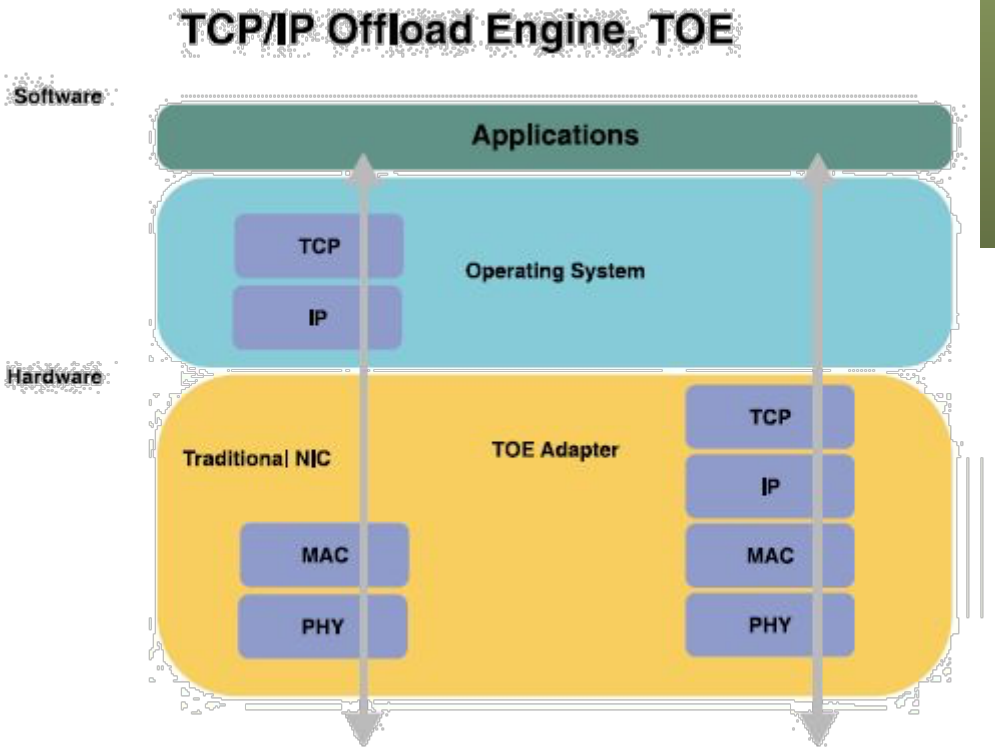
- Large Segment Offload, TCP Segmentation Offload
- TCP entrega a la NIC paquetes más grandes que la MTU
- La propia NIC hace la segmentación de nivel TCP
- Eso le obliga a crear nuevas cabeceras TCP e IP, descargando de ello a la CPU
- Requiere que la NIC sepa segmentar el protocolo (solo TCP)
- Problemas con encriptación (IPSec)
- Genera ráfagas de tráfico



TOE

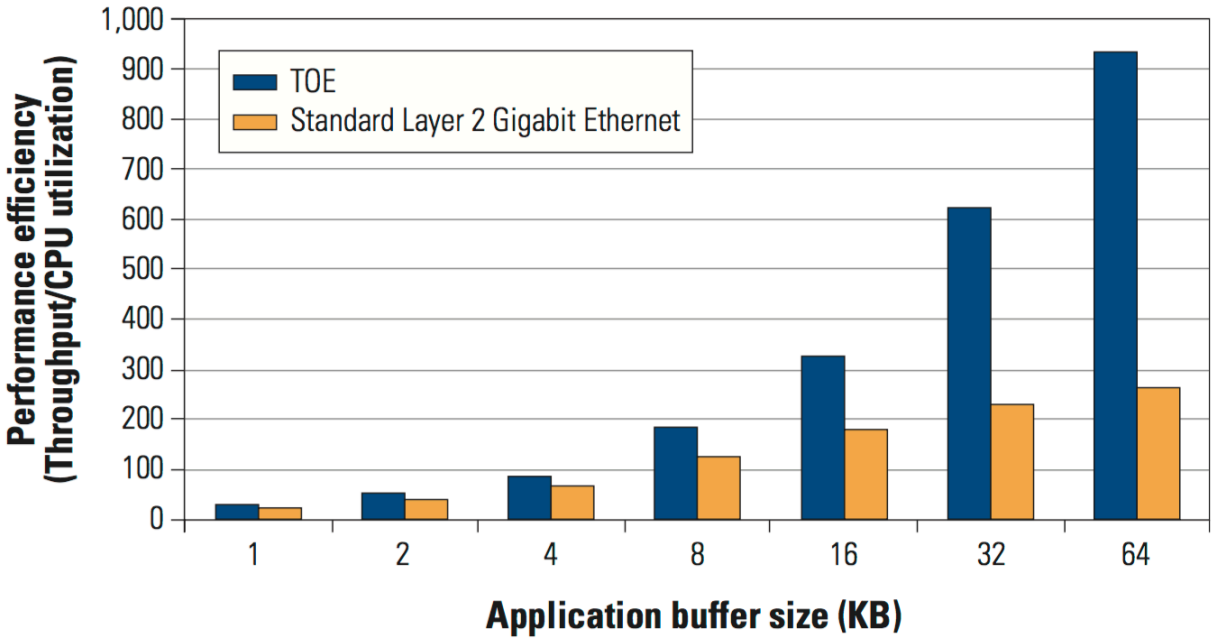
- *TCP/IP Offload Engine*
- Los datos pueden pasar directamente de la aplicación a la NIC
- La NIC puede emplearse para todas las tareas de la fase de transferencia y emplear la CPU para el establecimiento y terminación
- O se puede emplear la NIC para todo
- Requiere soporte del sistema operativo





TOE

- Puede mejorar el throughput
- Reduce la carga sobre la CPU



http://www.dell.com/downloads/global/power/ps3q06-20060132-Broadcom.pdf

Jumbo frames

- No están estandarizadas, la MTU estándar sigue siendo de 1500bytes
- Motivos para limitarlo
  - NICs tenían memoria limitada
  - Se quería limitar el tiempo que una estación tenía capturado el medio transmitiendo
  - El CRC es menos efectivo cuanto más grande es la trama
- Hoy en día (…)

Jumbo frames

- No están estandarizadas, la MTU estándar sigue siendo de 1500bytes
- Motivos para limitarlo
  - NICs tenían memoria limitada
  - Se quería limitar el tiempo que una estación tenía capturado el medio transmitiendo
  - El CRC es menos efectivo cuanto más grande es la trama
- Hoy en día no son problemas reales:
  - Decenas o centenares de Megabytes en la NIC
  - No tenemos medio compartido (ni coaxial ni hubs)
  - El CRC de Ethernet soporta más de 11 Kbytes de trama

Jumbo frames

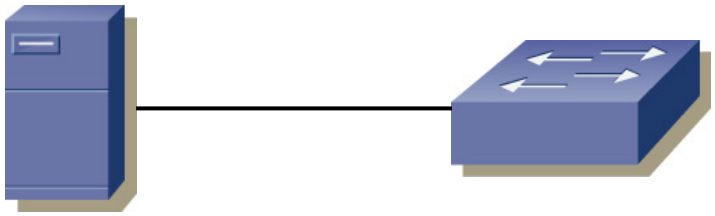
- Diversos estándares han ido aumentando el tamaño máximo de la trama (802.1Q, 802.1ad, MPLS, FCoE, etc)
- A estas últimas en ocasiones se las llama “Baby Giant”
- Jumbo frames suelen estar cerca de los 9 Kbytes (que se puedan transportar bloques de datos de 8Kbytes + encapsulados varios)
- ¿Postivo?
  - Cuanto más grandes menor ratio de cabeceras y menos interrupciones
  - Menos carga de procesado de cabeceras en equipos de red y hosts
- ¿Negativo?
  - (…)

Jumbo frames

- Diversos estándares han ido aumentando el tamaño máximo de la trama (802.1Q, 802.1ad, MPLS, FCoE, etc)
- A estas últimas en ocasiones se las llama “Baby Giant”
- Jumbo frames suelen estar cerca de los 9 Kbytes (que se puedan transportar bloques de datos de 8Kbytes + encapsulados varios)
- ¿Postivo?
  - Cuanto más grandes menor ratio de cabeceras y menos interrupciones
  - Menos carga de procesado de cabeceras en equipos de red y hosts
- ¿Negativo?
  - Mayores tramas sufren mayor retardo así que no son adecuadas para todos los servicios
  - Mayores tramas pueden llenar antes los buffers de los conmutadores
  - Todos los equipos del camino deben soportarlas
  - Posibles problemas con implementaciones que esperan 1500 bytes

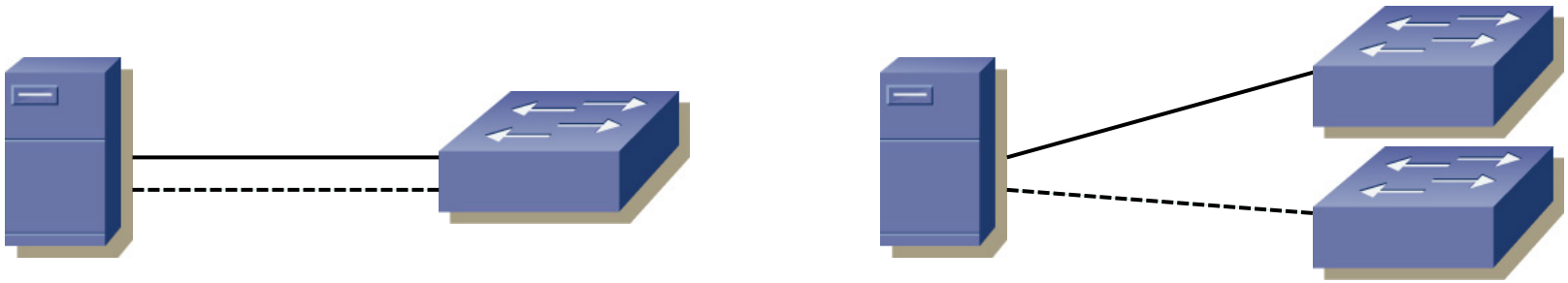
Server multihoming

- *NIC teaming / bonding / aggregation*
- Un servidor conectado a un conmutador presenta puntos únicos de fallo: la NIC, el cable, el conmutador
- Estas soluciones requieren colaboración del driver y normalmente también del sistema operativo
- Tenemos varias mejoras posibles (con una segunda o más NICs)
- (…)



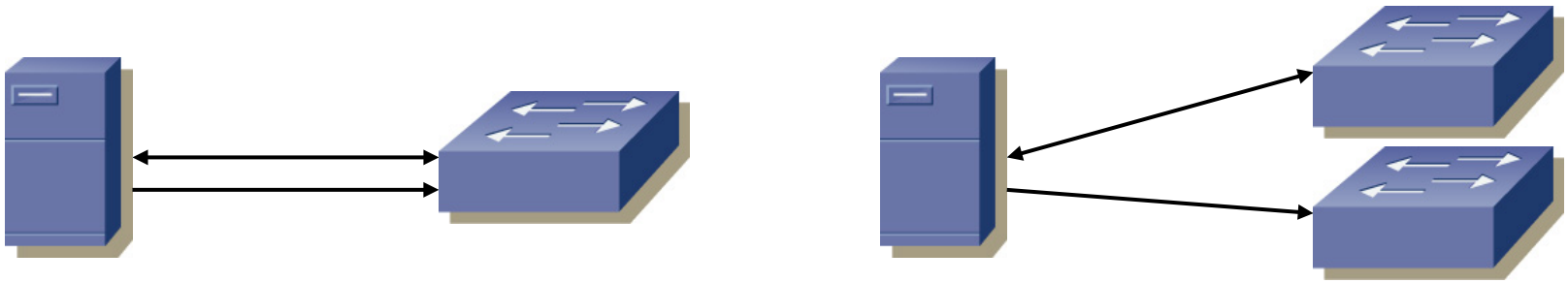
Server multihoming

- Un segundo enlace, modo activo-pasivo
  - Si falla el primero (la NIC, el conmutador o el cable) se activa el segundo con la misma dirección MAC e IP
  - Se desaprovecha el segundo enlace



Server multihoming

- Un segundo enlace, modo activo-pasivo
- O se usan los dos enlaces para transmitir pero solo se recibe por uno
- Cada interfaz suele enviar con diferente dirección MAC origen para no tener *MAC flapping* en el conmutador



Server multihoming

- Un segundo enlace, modo activo-pasivo
- O se usan los dos enlaces para transmitir pero solo se recibe por uno
- O se forma un LAG (802.3ad / 802.1AX)
  - Permite usar la capacidad de ambos enlaces
  - Normalmente requiere colaboración por parte del switch
  - Si se quiere redundancia de switch hay que hacer una agregación en la que un extremo son 2 conmutadores