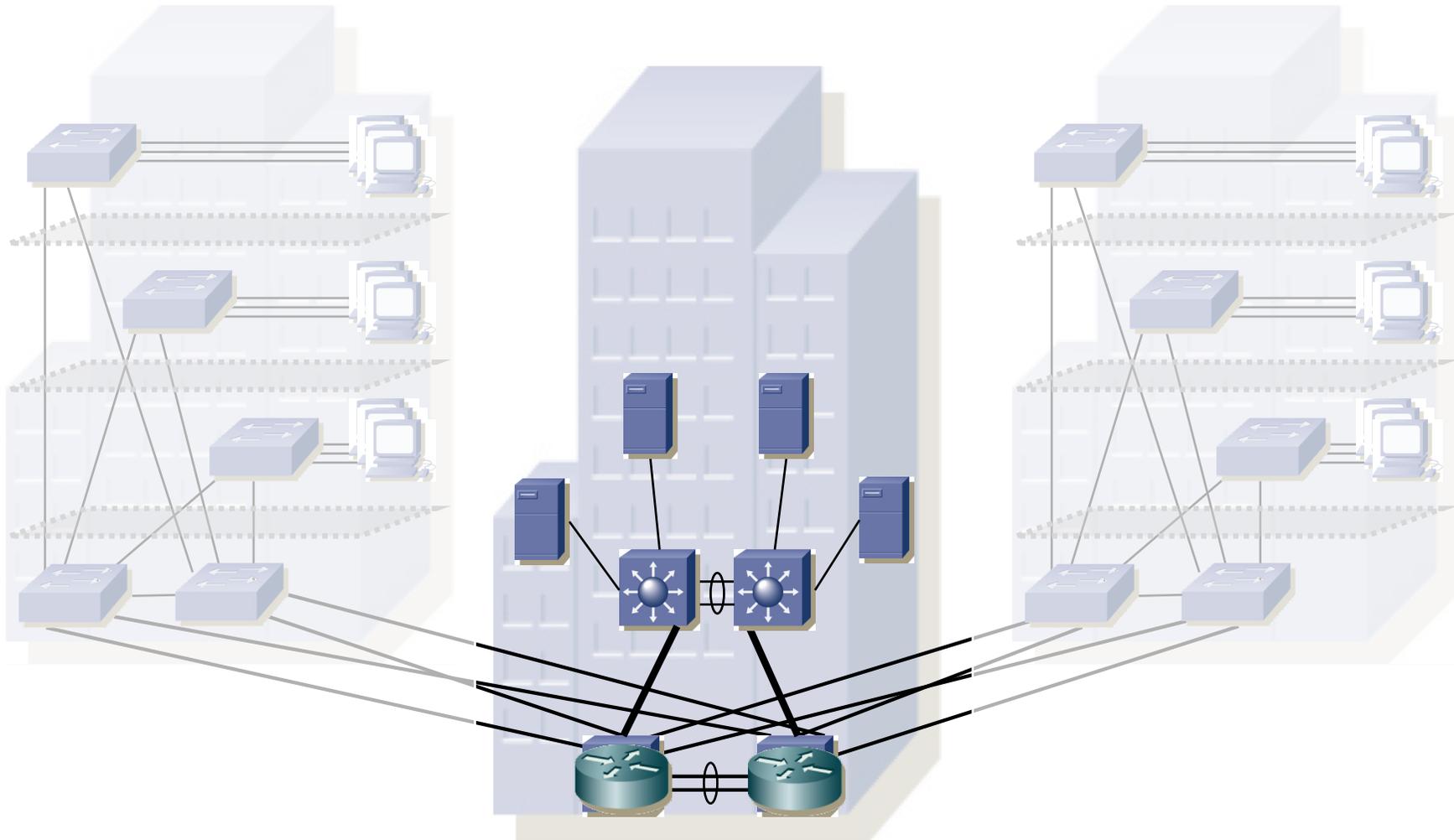


Diseño del data center

Diseño clásico

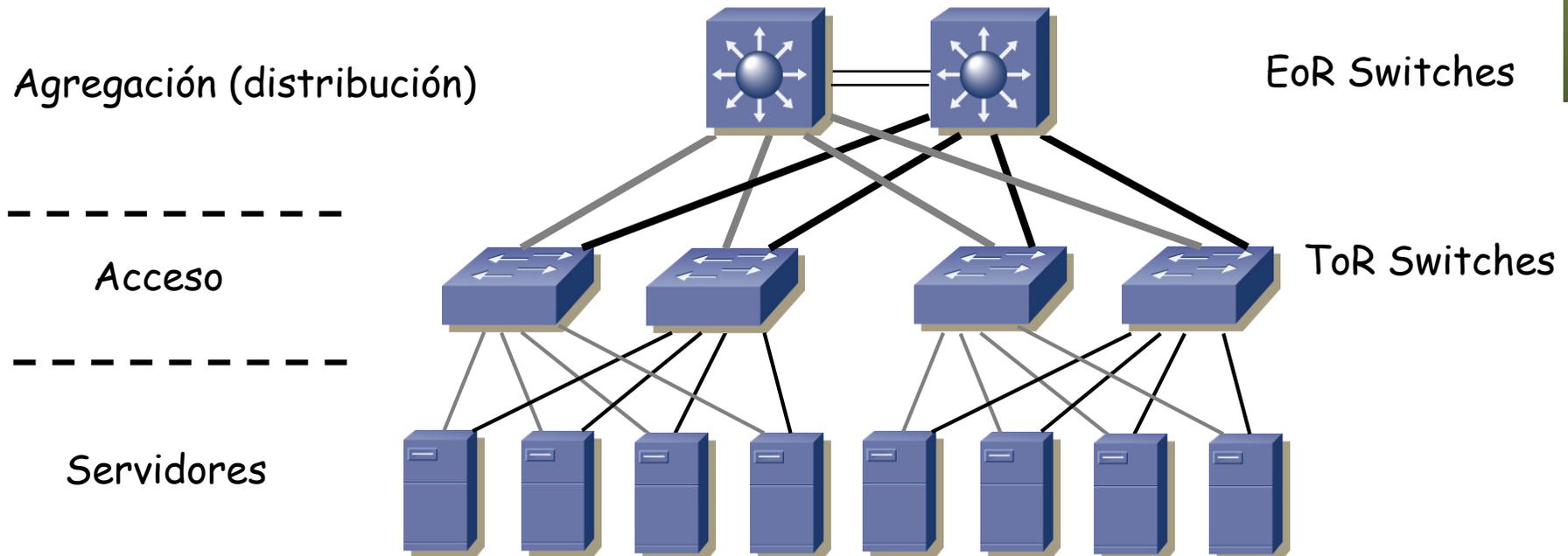
Sala de servidores

- Escalabilidad
- Flexibilidad
- Alta disponibilidad
- Rendimiento
- Eficiencia energética
- Coste



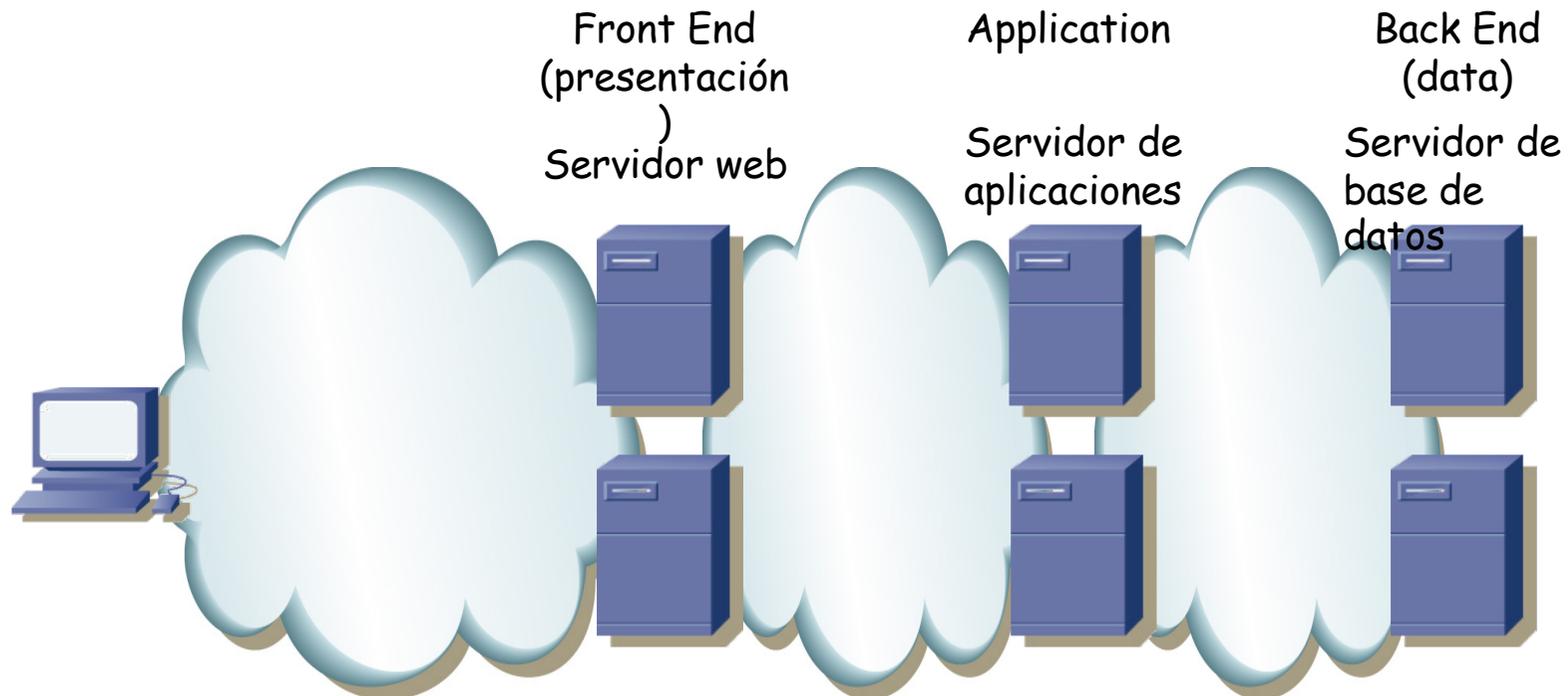
Diseño genérico

- Los conmutadores de la capa de acceso dan alta densidad de puertos
- Los conmutadores de agregación agregan tráfico hacia y desde el acceso y para conectar con los servicios de red
- No hay puntos únicos de fallo
- Enlaces gigabit o 10G a los servidores
- Enlaces gigabit, 10G o LAGs entre los conmutadores
- Todo full duplex, ¡nada de hubs!



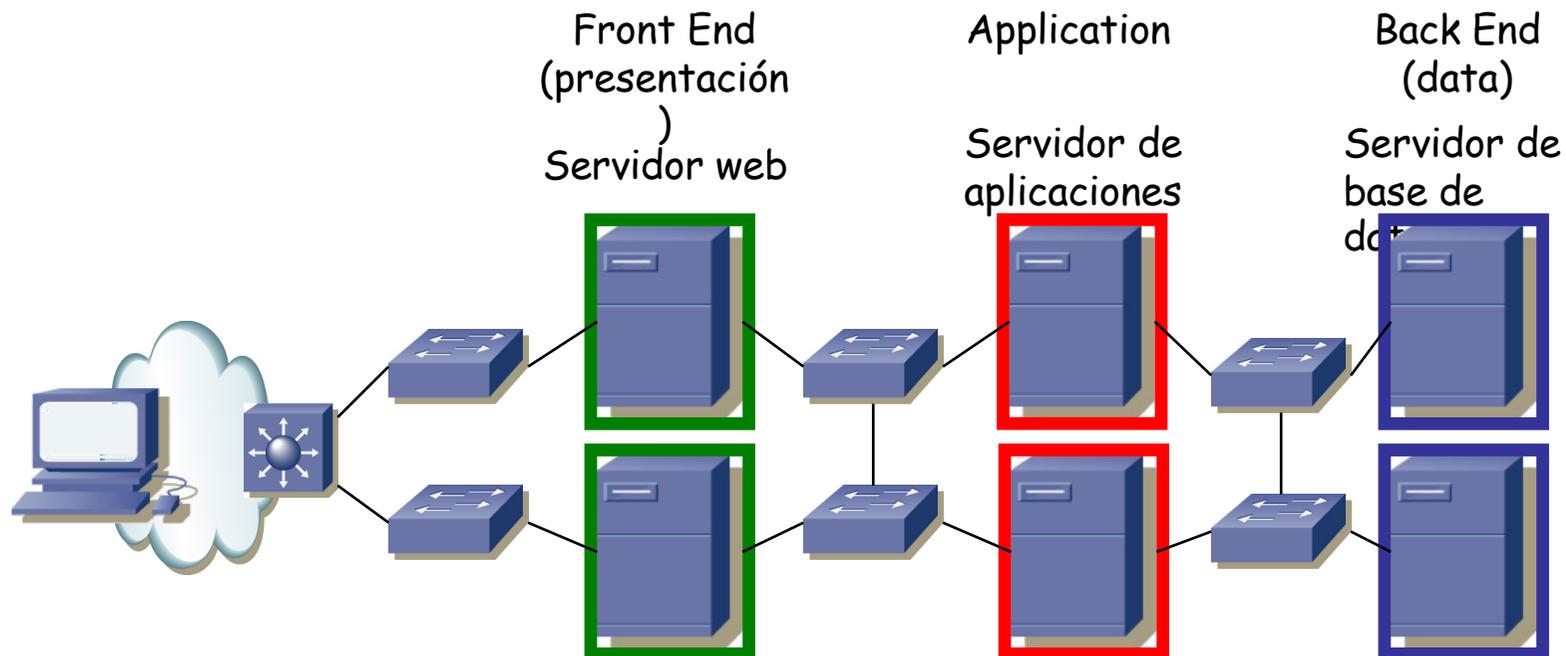
Red para n-tier

- ¿La red para las diversas capas del servicio?
- (...)



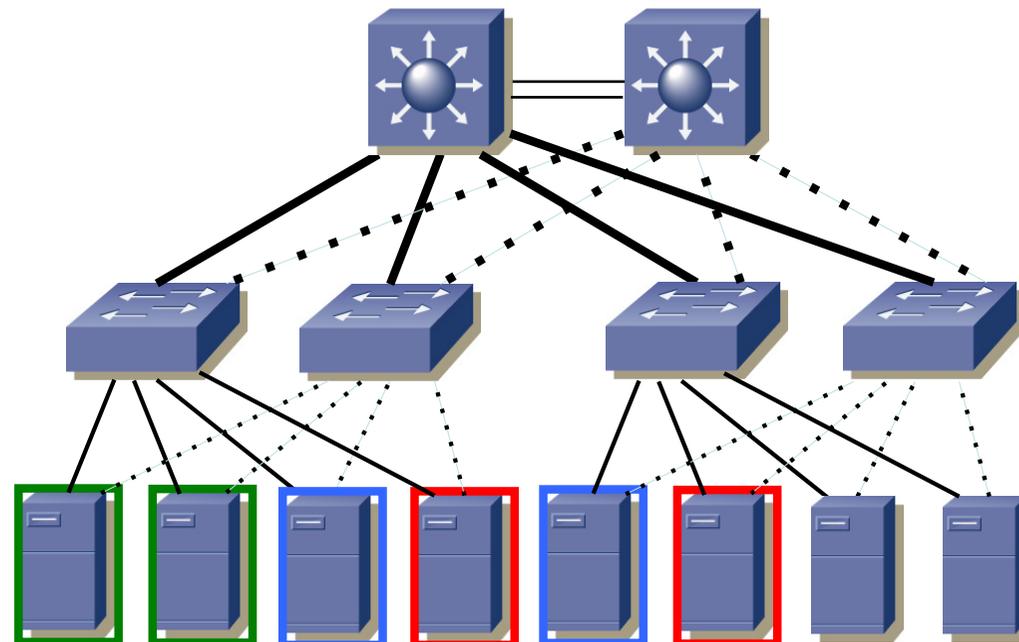
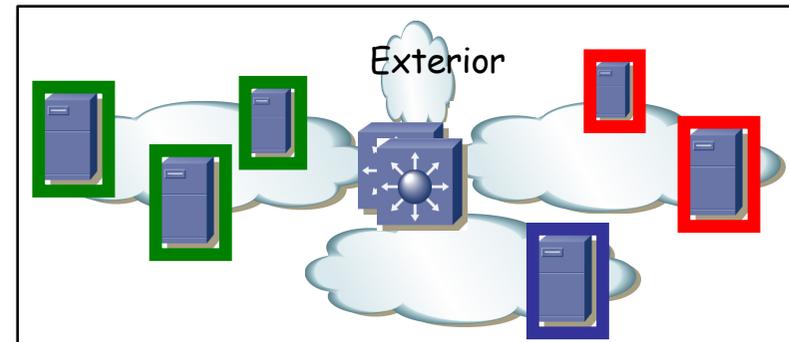
Red para n-tier

- ¿La red para las diversas capas del servicio?
- Es común que los servidores tenga varios interfaces
- Y que empleen uno hacia la capa anterior y otro hacia la siguiente
- ¿Creamos una red para cada segmento? (con sus firewall, balanceadores, etc)



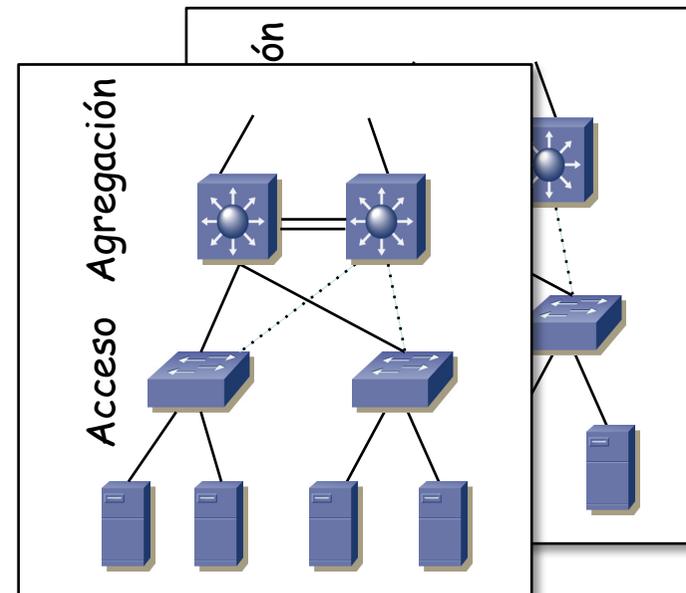
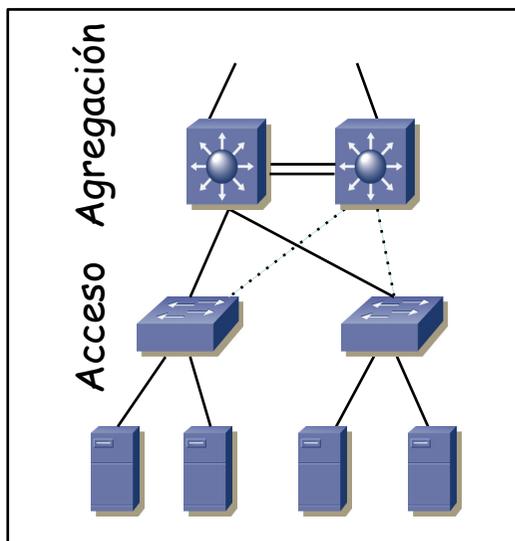
Red para n-tier

- Lo normal es emplear VLANs
- Es decir, mediante virtualización en la red tenemos las tres capas de acceso
- ¿Qué funcionalidades necesitamos entonces en la capa de agregación?
- Conmutación capa 2 en la VLAN, capa 3 entre ellas
- En la VLAN árbol de expansión
- Se puede emplear un FHRP (...)



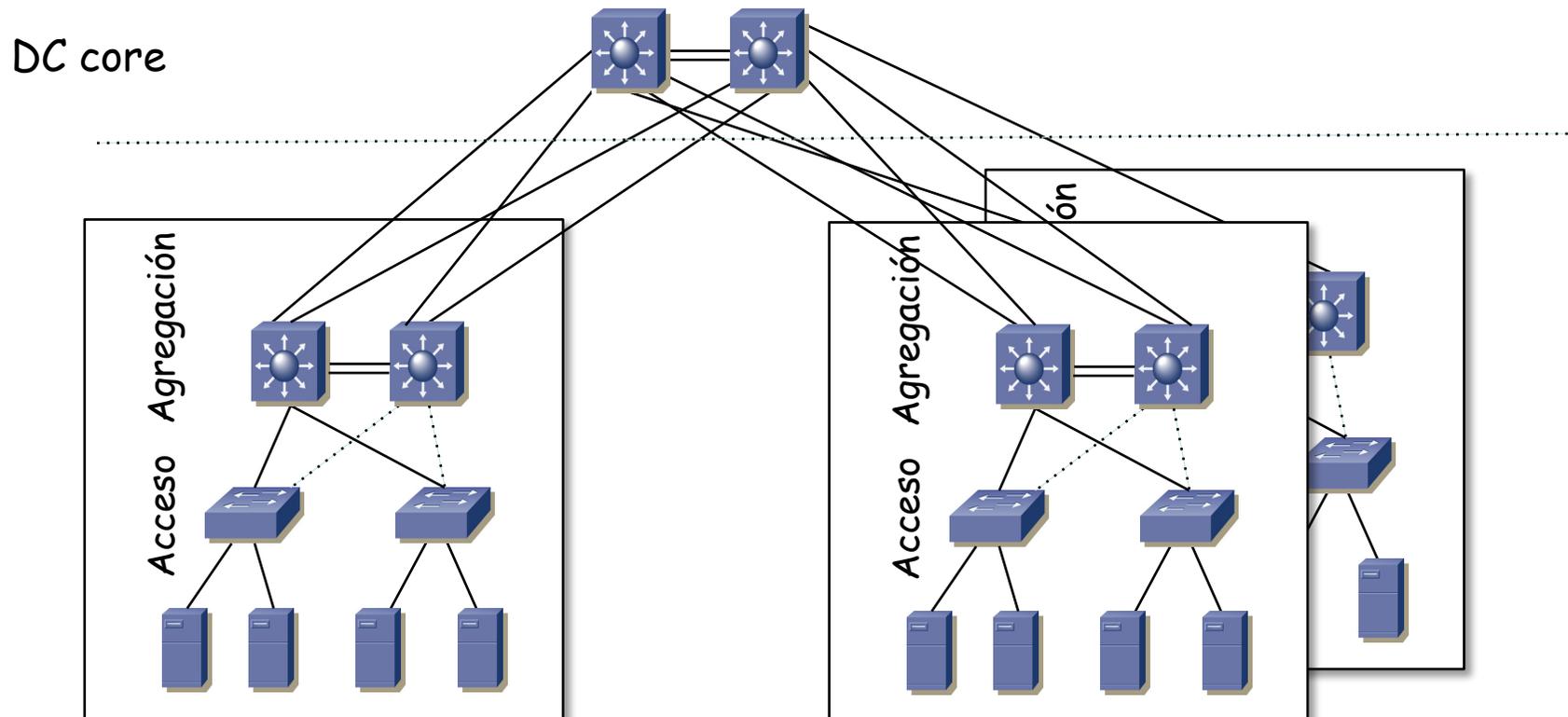
Data center core

- Puede que no sea suficiente con un par de conmutadores de agregación
- Por ejemplo por limitación en número de puertos
- O por políticas que recomienden la separación de capas de agregación
- (...)



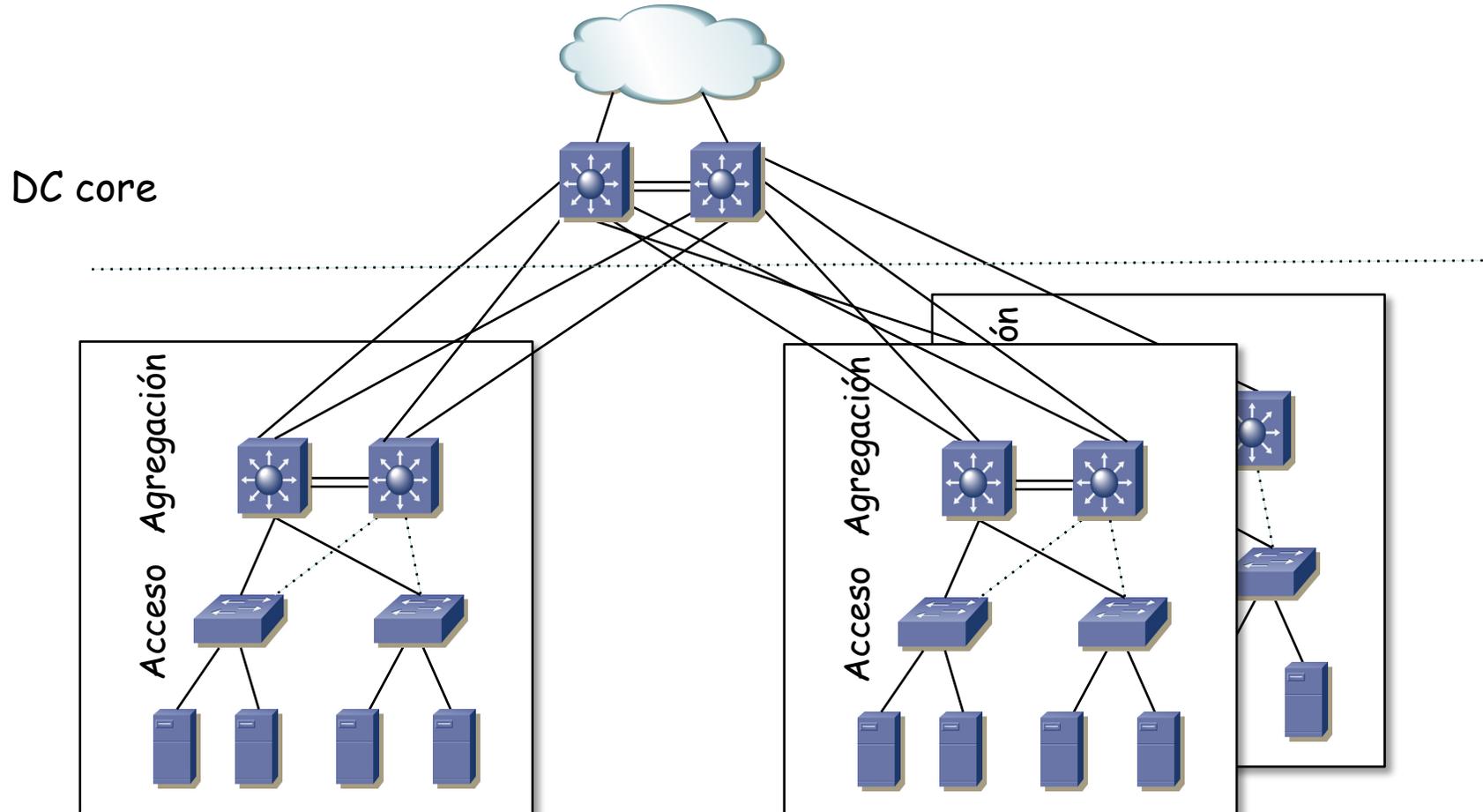
Data center core

- Puede que no sea suficiente con un par de conmutadores de agregación
- Por ejemplo por limitación en número de puertos
- O por políticas que recomienden la separación de capas de agregación
- En ese caso añadimos una capa de *core* del data center



Data center core

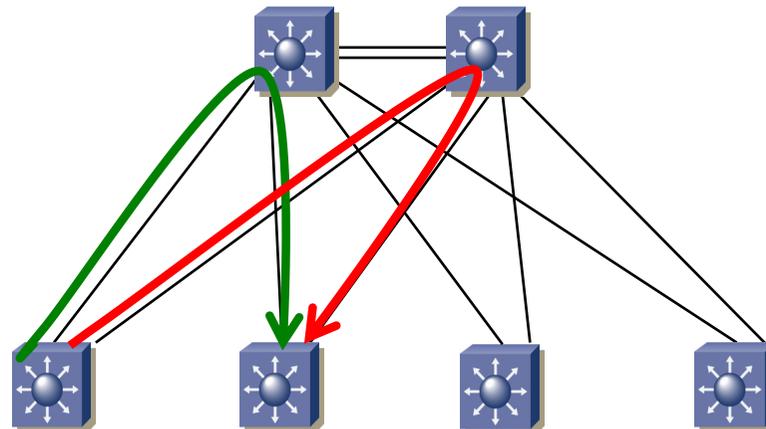
- Enlaces 10GE
- Enlaces agregación-core en capa 3
- Se conectará al core de la red Campus
- Pueden ser directamente los equipos del core del campus pero podemos estar limitados por número de puertos



L3 y ECMP

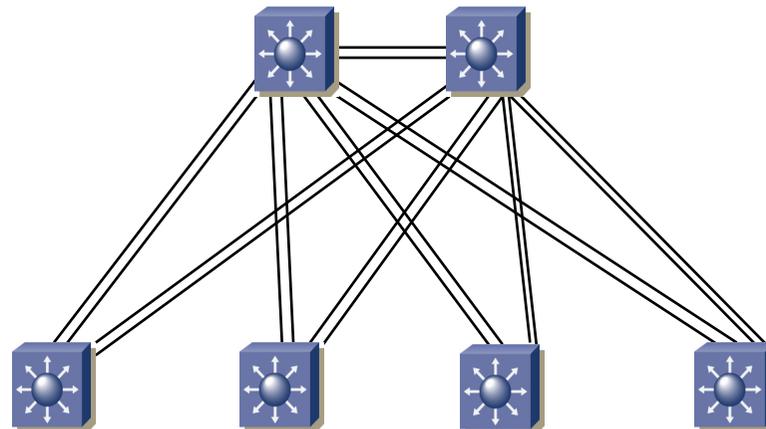
L3 y ECMP

- Escenarios en los que tenemos enlaces capa 3
- Puede ser entre core y agregación o entre agregación y acceso
- Empleamos un protocolo de encaminamiento dinámico
- En muchas ocasiones estos protocolos pueden calcular varias rutas del coste mínimo
- Podemos aumentar la capacidad con LAGs (...)



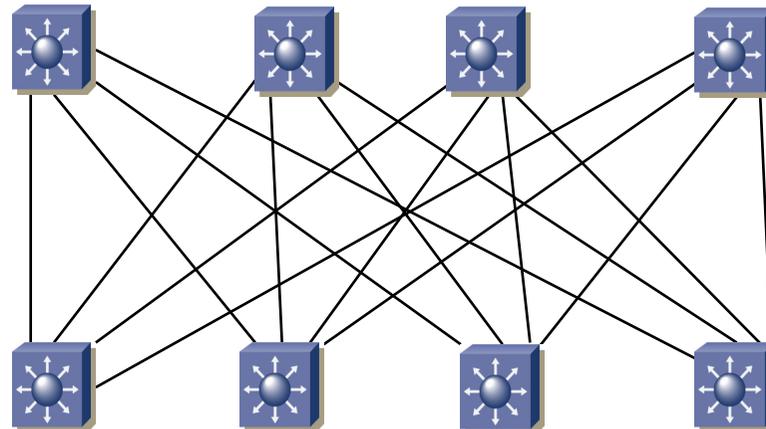
L3 y ECMP

- Escenarios en los que tenemos enlaces capa 3
- Puede ser entre core y agregación o entre agregación y acceso
- Empleamos un protocolo de encaminamiento dinámico
- En muchas ocasiones estos protocolos pueden calcular varias rutas del coste mínimo
- Podemos aumentar la capacidad con LAGs (hemos puesto parejas de enlaces pero podrían ser más)
- (...)



L3 y ECMP

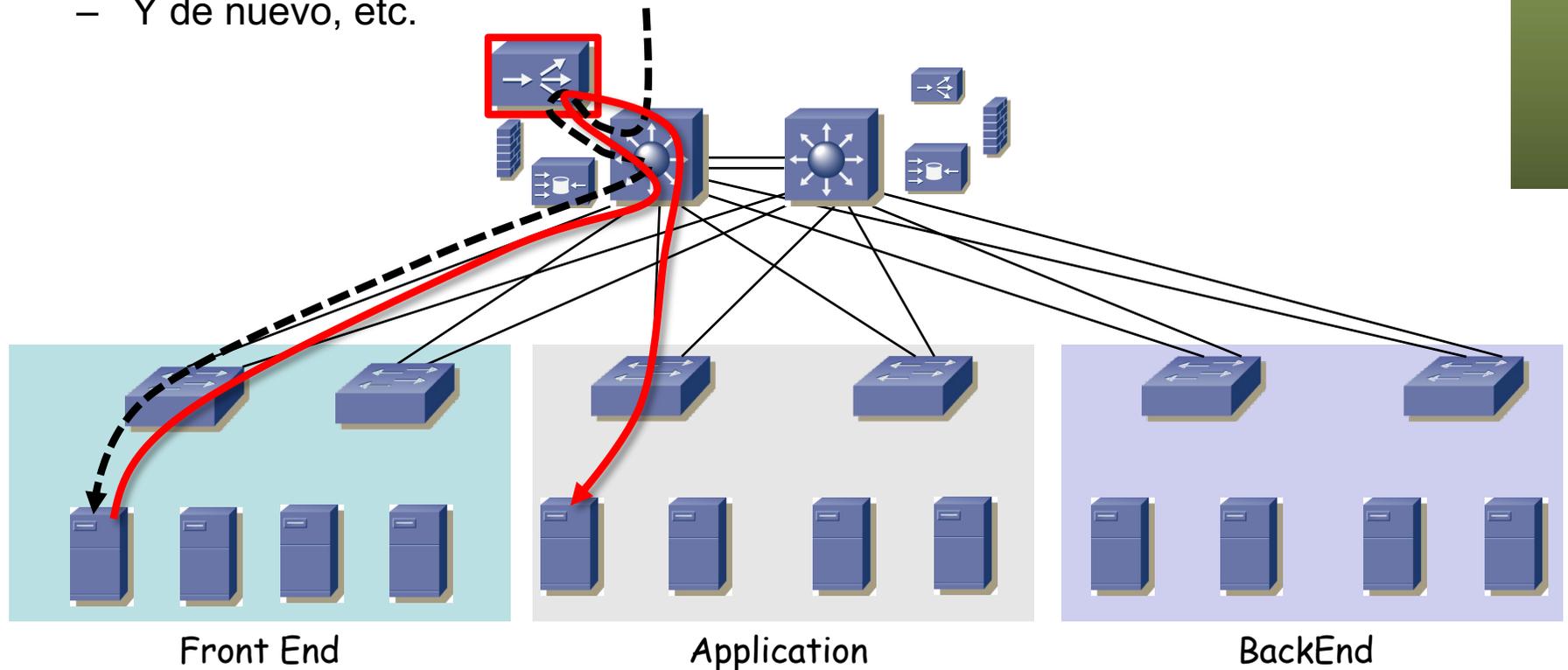
- Y/o podríamos aumentar el número de caminos aumentando el número de conmutadores en la capa superior
- En este ejemplo hay 4 caminos de igual coste entre cada pareja de conmutadores
- Si las capas son core y agregación recordemos que por debajo están los conmutadores de acceso
- Si las capas son agregación y acceso recordemos que por debajo están los servidores
- Estamos limitados por el número máximo de puertos disponibles en los equipos



Ubicación de los servicios

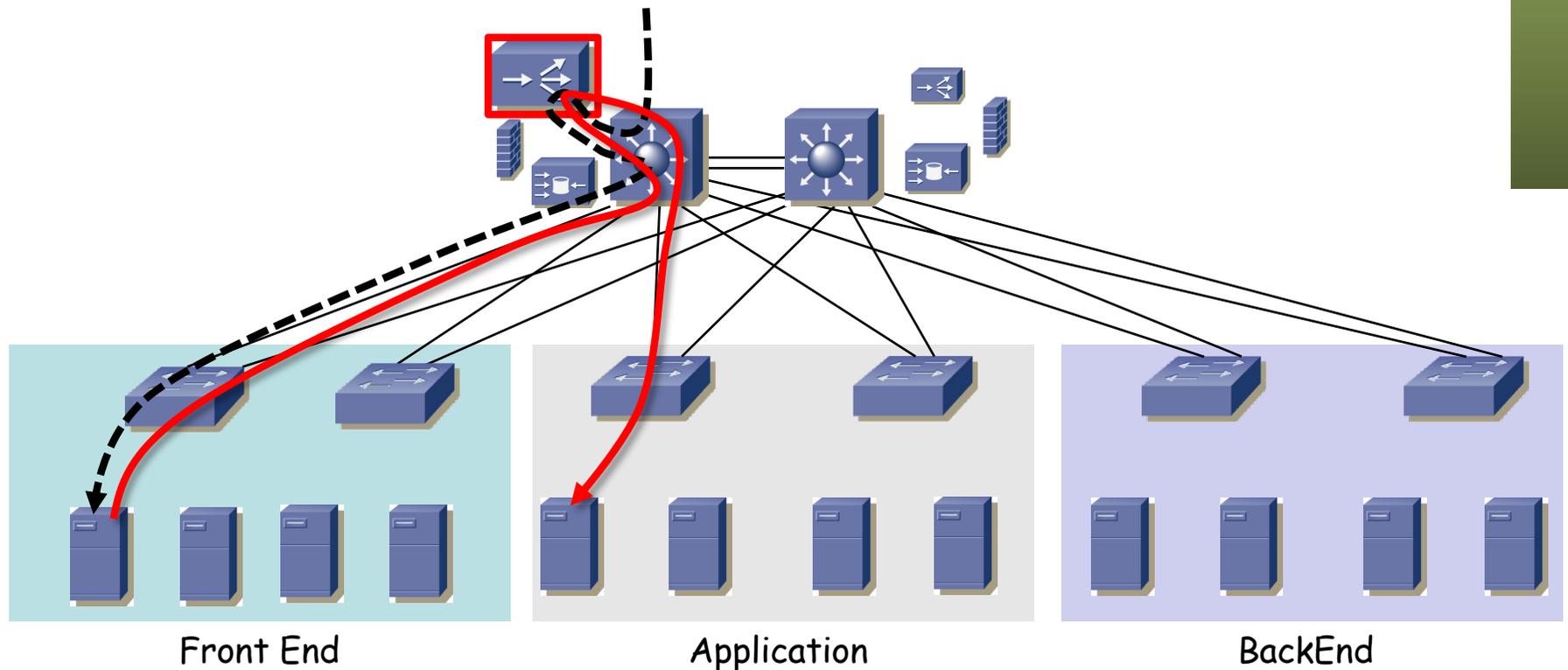
Servicios: ¿Dónde?

- 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.



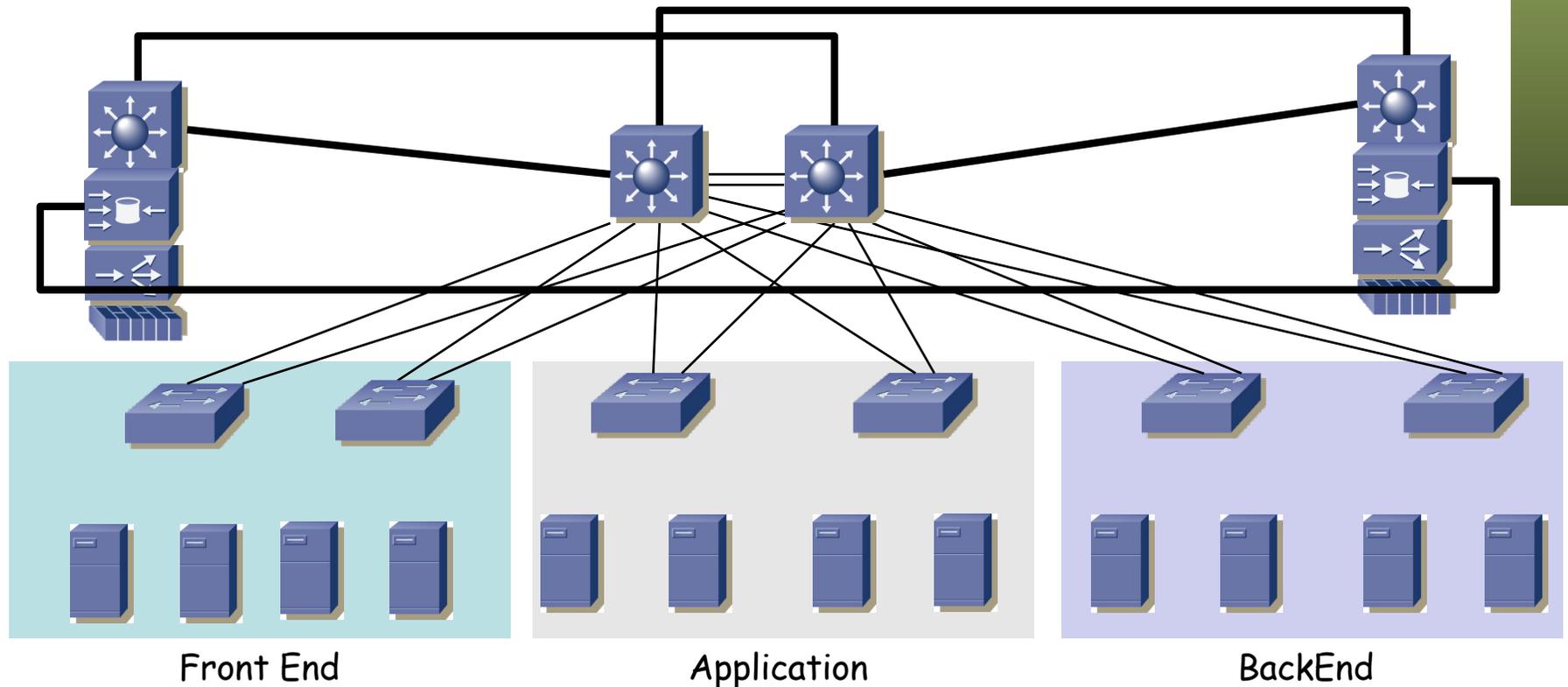
Servicios: ¿Dónde?

- 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



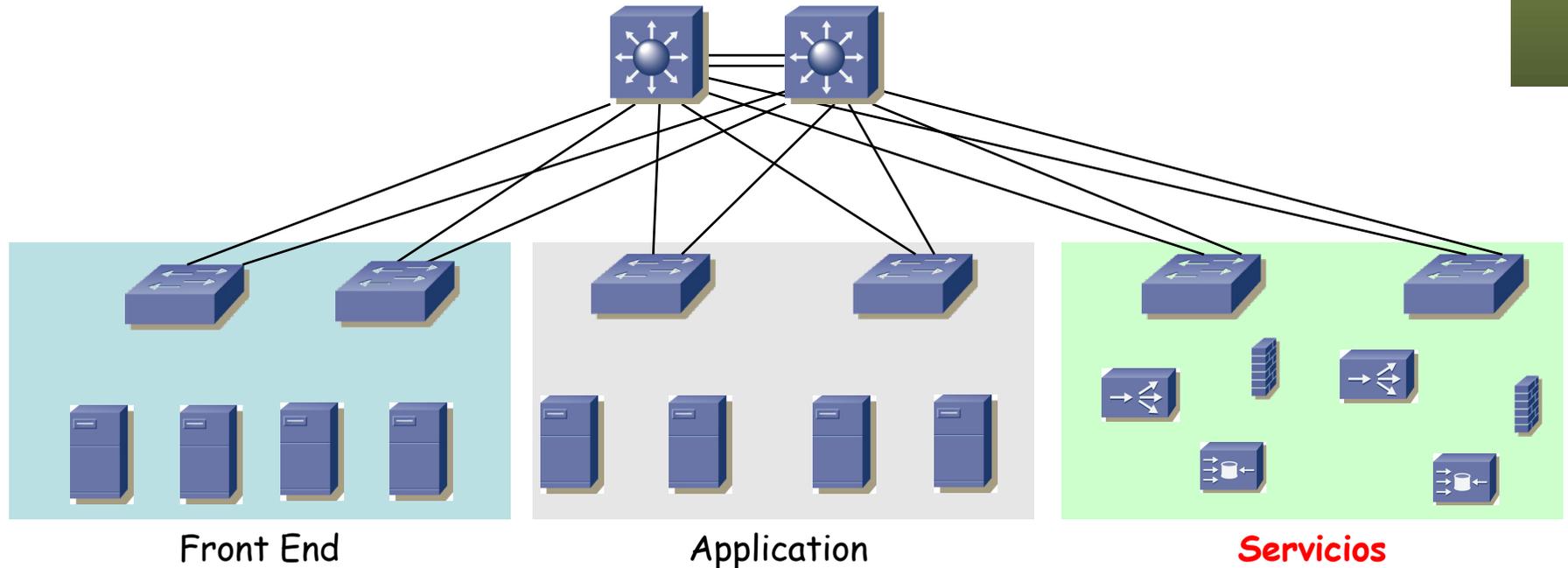
Servicios: ¿Dónde?

- 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*)
- (...)



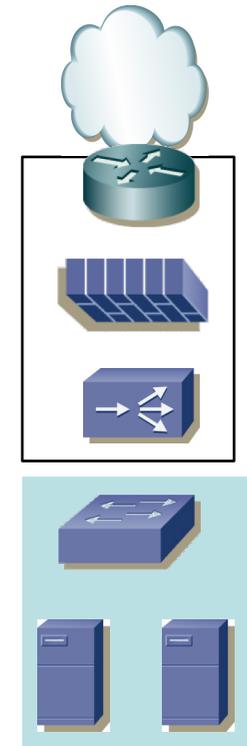
Servicios: ¿Dónde?

- 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



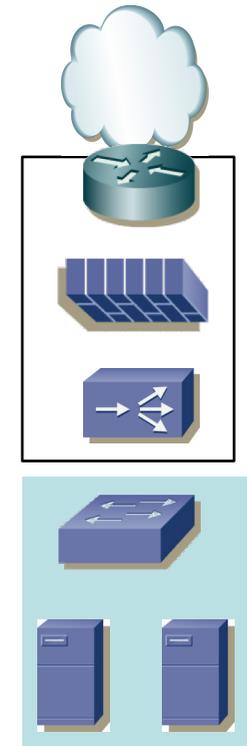
Orden de los servicios

- 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
- (...)



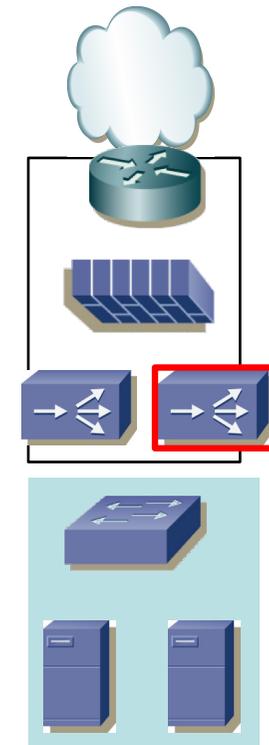
Router-Firewall-Balanceador

- 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



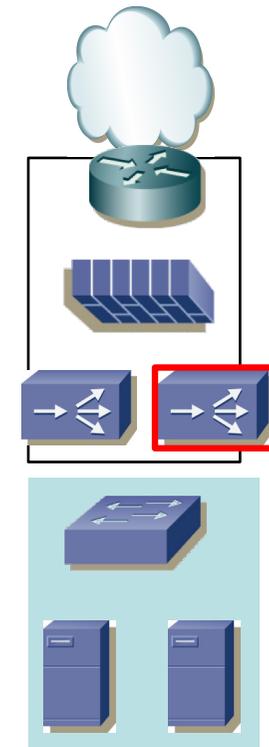
Router-Firewall-Balanceador

- 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
- (...)



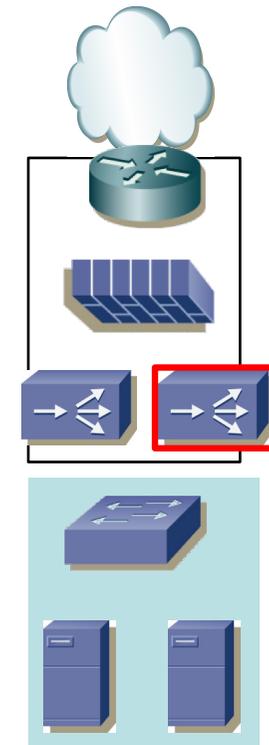
Router-Firewall-Balanceador

- 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 emplear 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
- (...)



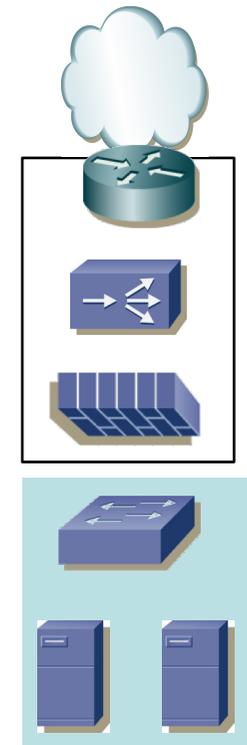
Router-Firewall-Balanceador

- 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



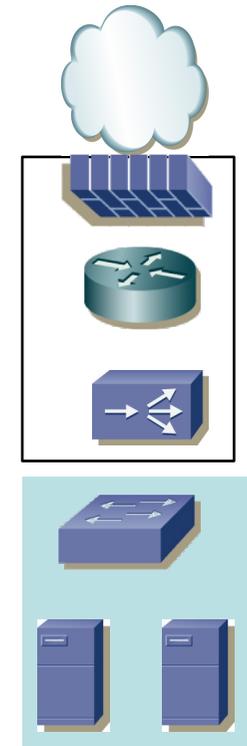
Router-Balanceador-Firewall

- 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
- Si el firewall actúa como router los *health probes* del balanceador deben ser enrutables



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

