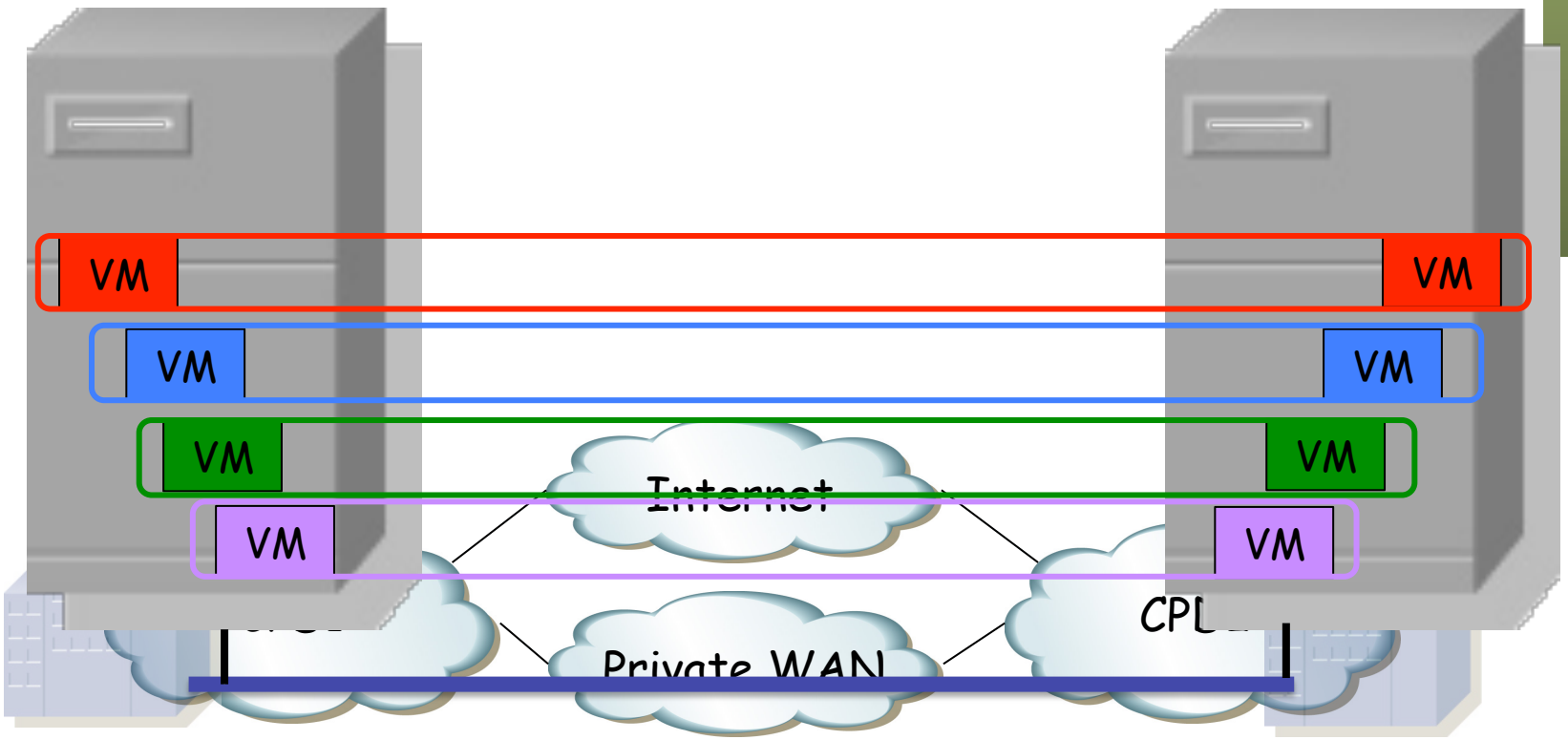
VXLAN

VXLAN

- RFC 7348 “Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks” (agosto 2014)
- RFC Informativa firmada por Cisco, VMware, Intel, Red Hat, Arista y Cumulus Networks
- Diseñado para un entorno de host virtualizado
- Emplea un esquema de overlay de capa 2 sobre capa 3 (o sea, un túnel), en el mismo data center o en otro

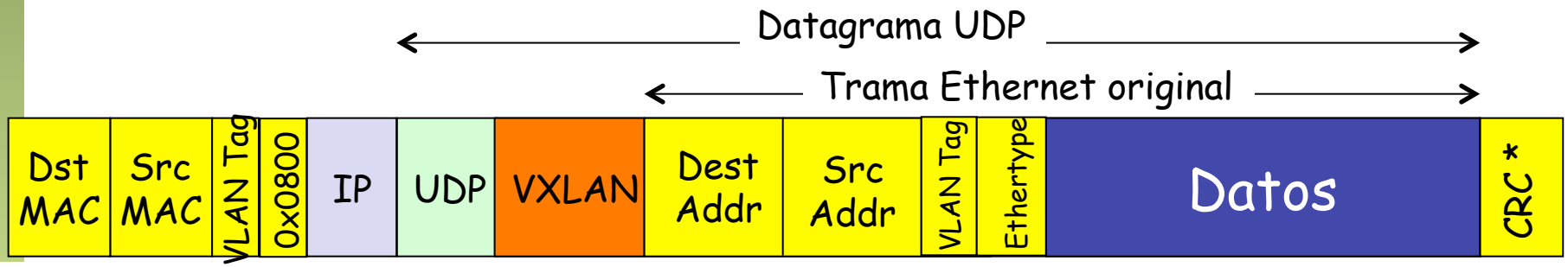
VXLAN

- RFC 7348 “Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks” (agosto 2014)
- RFC Informativa firmada por Cisco, VMware, Intel, Red Hat, Arista y Cumulus Networks
- Diseñado para un entorno de host virtualizado
- Emplea un esquema de overlay de capa 2 sobre capa 3 (o sea, un túnel), en el mismo data center o en otro



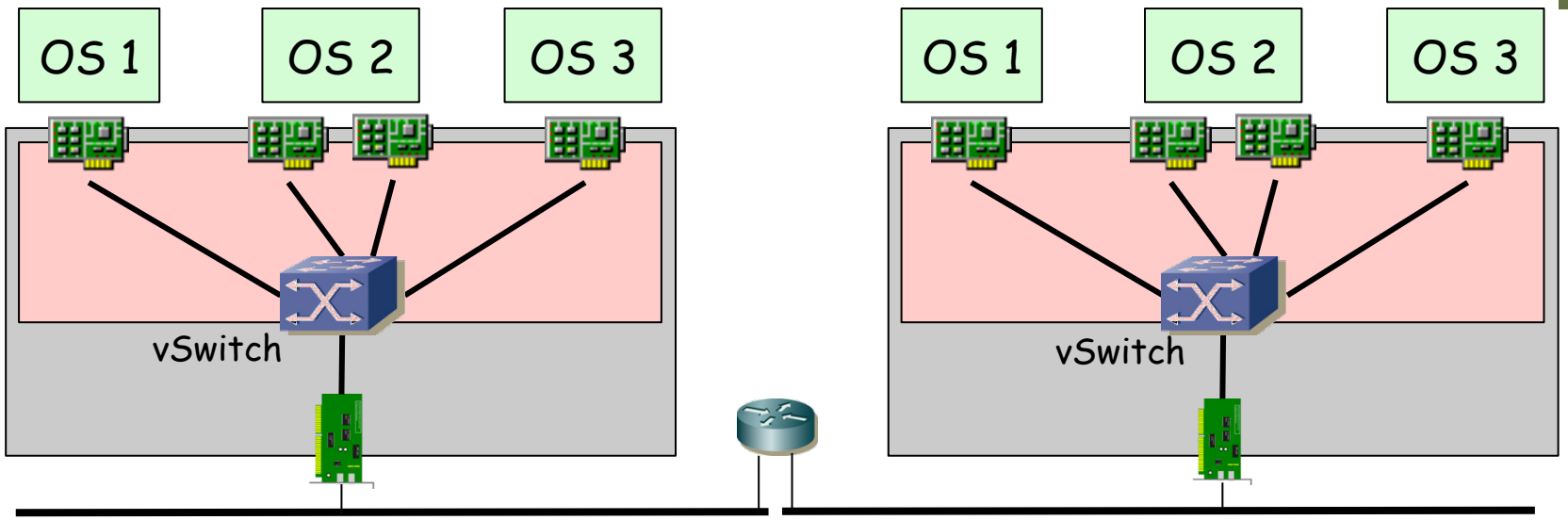
VXLAN

- Emplea un esquema de overlay de capa 2 sobre capa 3 (o sea, un túnel), en el mismo data center o en otro
- En realidad sobre capa 4 pues hace el transporte sobre UDP
- Puerto destino 4789, puerto origen se recomienda un hash de campos de la trama original para facilitar el balanceo de flujos en la red IP
- La cabecera VXLAN es de 8 bytes y fundamentalmente contiene el VNI
- VNI = *VXLAN Network Identifier* (de 24 bits)
- En un entorno de DC con múltiples usuarios permite separar más de los 4094 que permitiría una etiqueta de VLAN
- Los VLAN Tags (trama externa e interna) son opcionales
- Para las máquinas virtuales es transparente



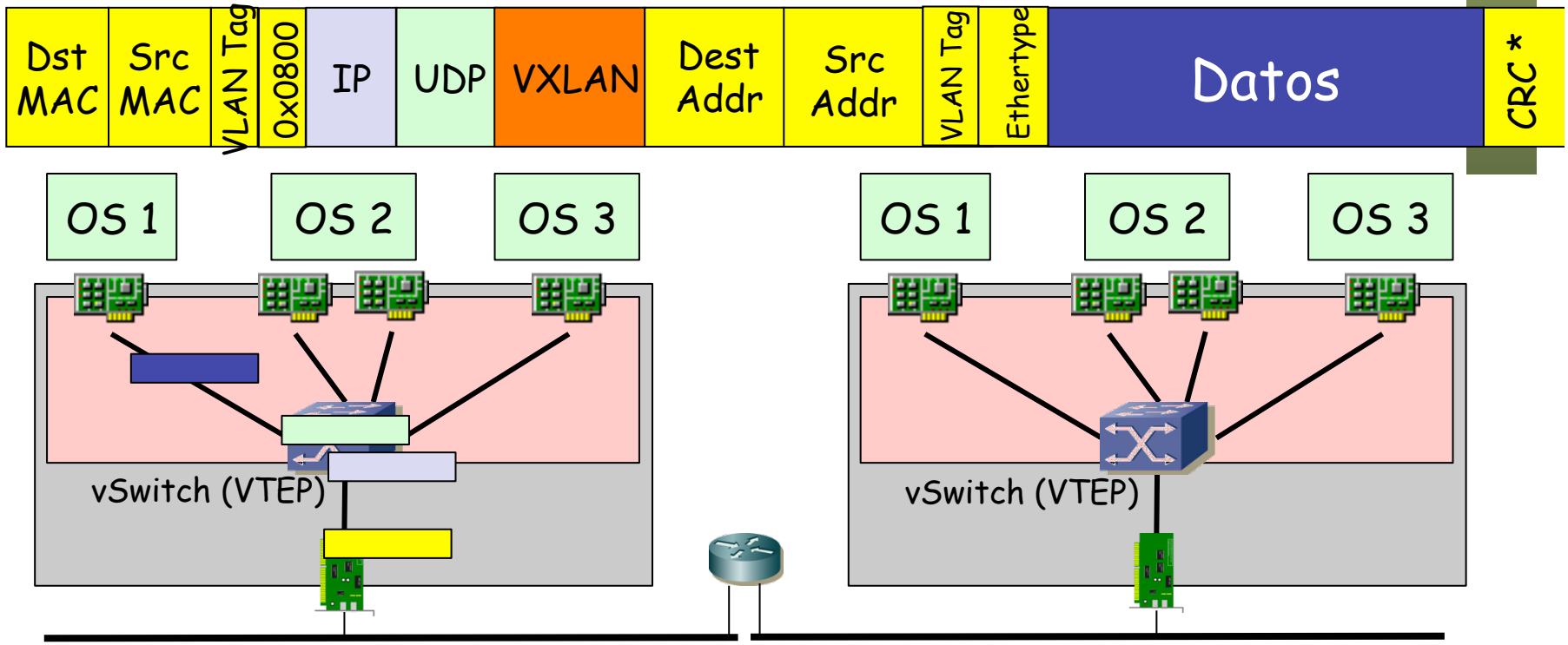
VXLAN: *Data plane*

- Cada overlay se conoce como un “segmento VXLAN”
- Los hosts (VMs) de un segmento VXLAN solo pueden comunicarse entre ellos
- Se pueden repetir las direcciones MAC en distintos segmentos
- El extremo que encapsula la trama original se llama el VTEP (VXLAN Tunnel End Point)
- El VTEP se suele encontrar en el hypervisor (transparente para la VM)
- Podría estar si no en un ToR switch



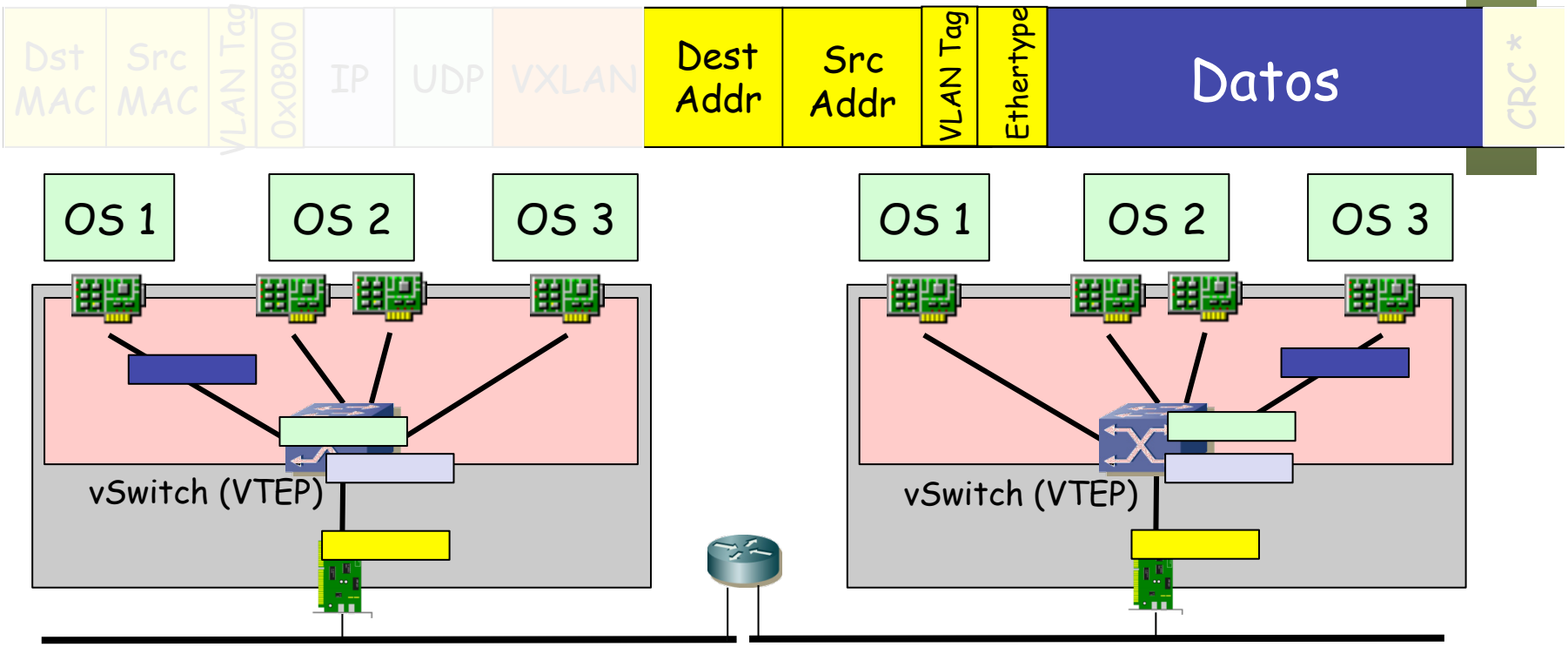
VXLAN: *Data plane*

- La trama Ethernet que envía una VM la recibe el vSwitch
- La encapsula con el VNI (configuración de la VM) en un datagrama UDP
- Averigua la dirección IP del host que contiene la VM con esa MAC destino
- Le envía el paquete IP que contiene la trama
- Por supuesto en una trama Ethernet



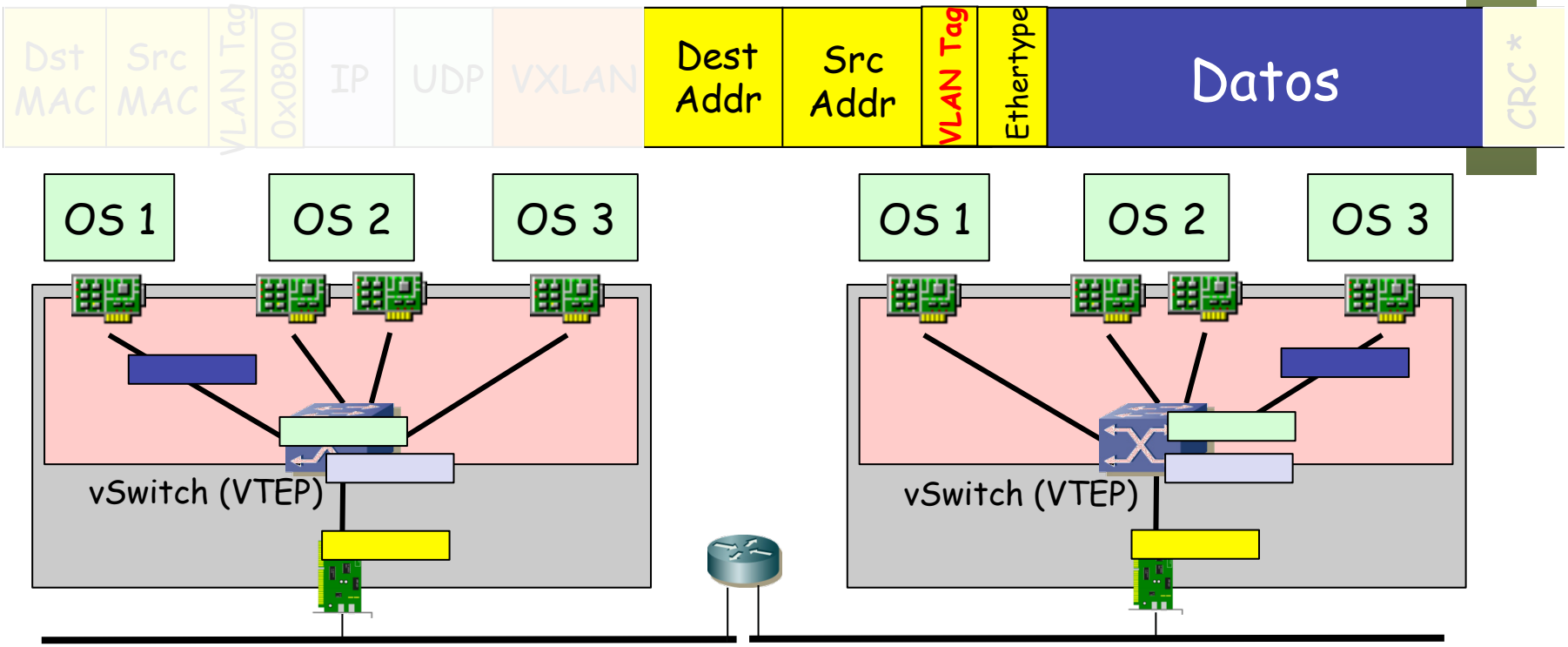
VXLAN: *Data plane*

- Si hay LAGs los switches que repartan flujos en función de capa 3+ pueden repartir estos flujos por los enlaces (si lo hacen por MAC peor)
- En el receptor el proceso es el inverso
- La VM destino nunca ve el paquete VXLAN
- Recibe directamente la trama que envió la VM origen



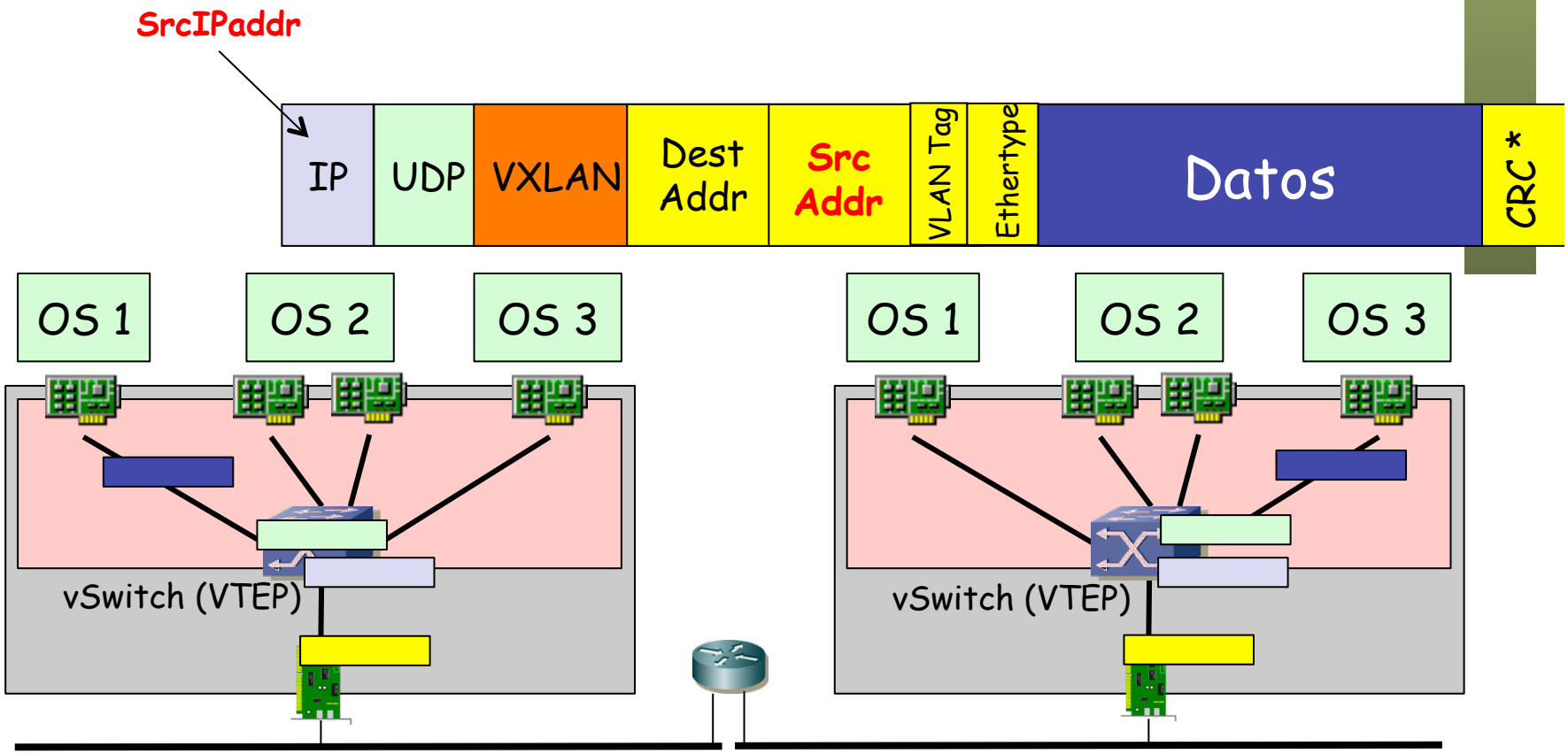
VXLAN: *Data plane*

- El transporte entre las VMs es de las tramas Ethernet
- Se comportan como si estuvieran en la misma VLAN
- ¿O en varias VLANs? A fin de cuentas transporta el V-Tag
- La RFC no lo deja claro y parece más inclinada a retirar esa etiqueta (sección 6.1)



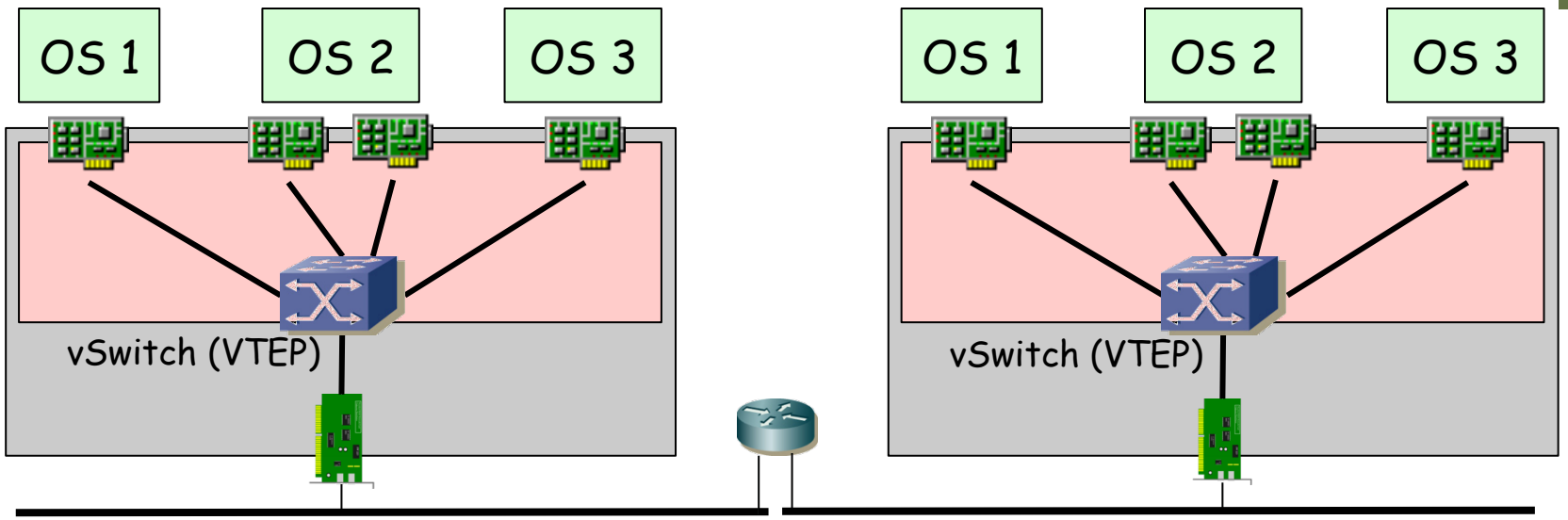
VXLAN: *Control plane*

- Los vSwitch deben aprender la dirección IP del host que hospeda una VM
- En este caso, al recibir un paquete de datos
- Aprende que la dirección MAC origen en el contenido es de un host en la máquina con dirección IP la origen en el continente



VXLAN: *Control plane*

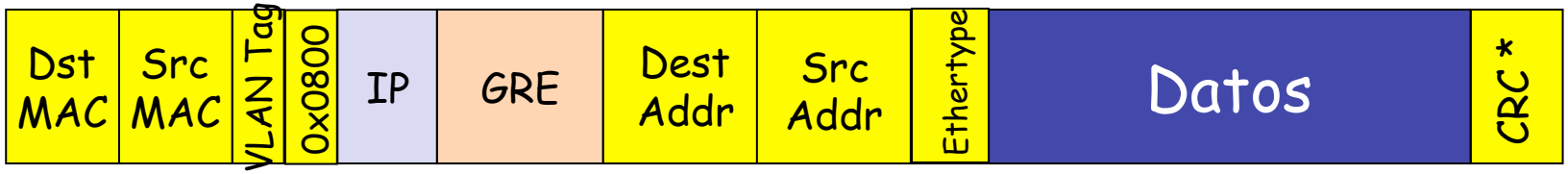
- ¿Y el BUM? Por ejemplo los ARP
- Se envía a un grupo multicast IP (uno por segmento VXLAN)
- Todos los hosts del segmento VXLAN pertenecen a ese grupo
- Esto implica routing multicast en la red IP (algo como PIM-SM)
- El número de grupos multicast soportados por la red puede ser limitado, lo cual llevaría a compartirlos para varios segmentos VXLAN



NVGRE

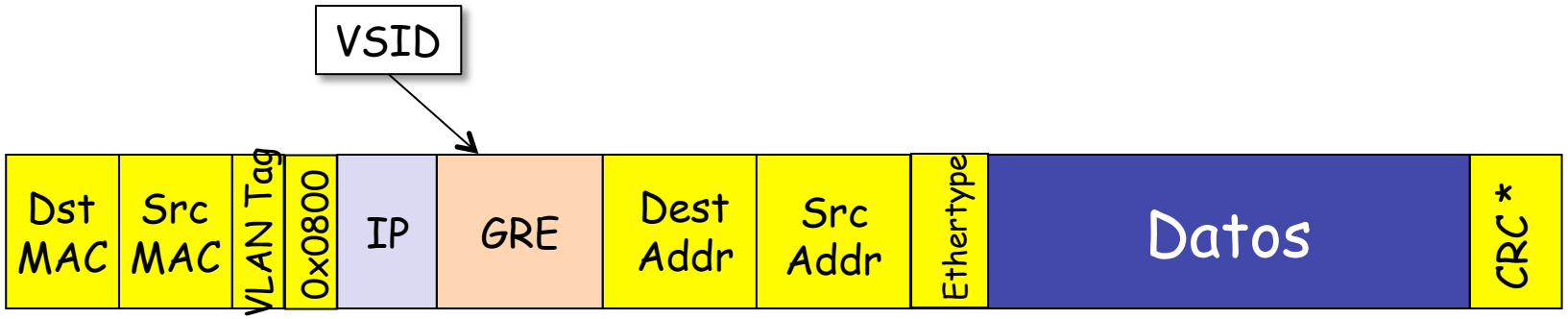
NVGRE

- RFC 7637 "*NVGRE: Network Virtualization using Generic Routing Encapsulation*"
- RFC Informativa (Sept.2015) firmada por Microsoft
- Crea una topología capa 2 virtual sobre una red capa 3
- La trama (sin V-TAG) es encapsulada en el extremo (host, switch virtual, etc) en un paquete GRE y en un paquete IP (protocolo 0x2F)



NVGRE

- El extremo se llama el NVGRE Endpoint
- La cabecera GRE contiene un Virtual Subnet ID (VSID)
  - De 24 bits (parte del campo *key* de GRE)
  - Los 8 bits restantes de la clave se usan para distinguir flujos y poder hacer reparto de carga en routers que entiendan GRE
  - Permite identificar un dominio broadcast capa 2 en un entorno multi-tenant



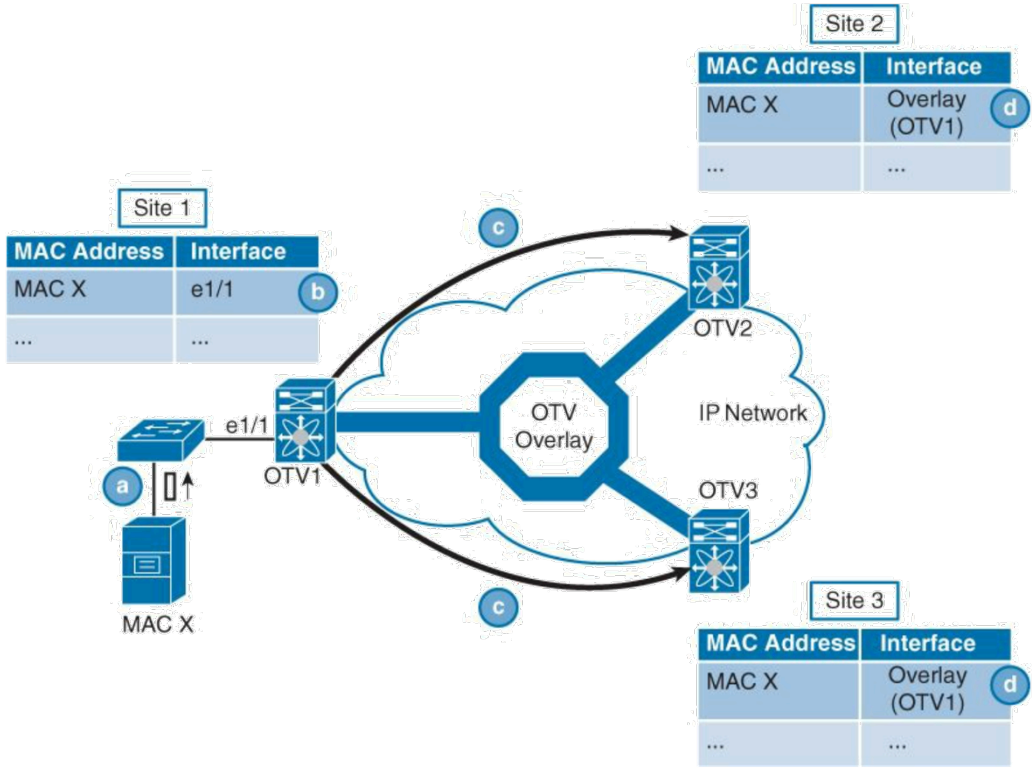
NVGRE

- La RFC no detalla cómo el Endpoint conoce la dirección del destino al que mandar el paquete IP
- Broadcast y multicast
  - Se puede emplean encaminamiento multicast IP con una o más direcciones multicast por VSID
  - Se puede implementar con N-way unicast
- Lo soporta Hyper-V (draft propuesto por Microsoft)
- NICs pueden soportar *offloading* del encapsulado NVGRE

OTV

OTV

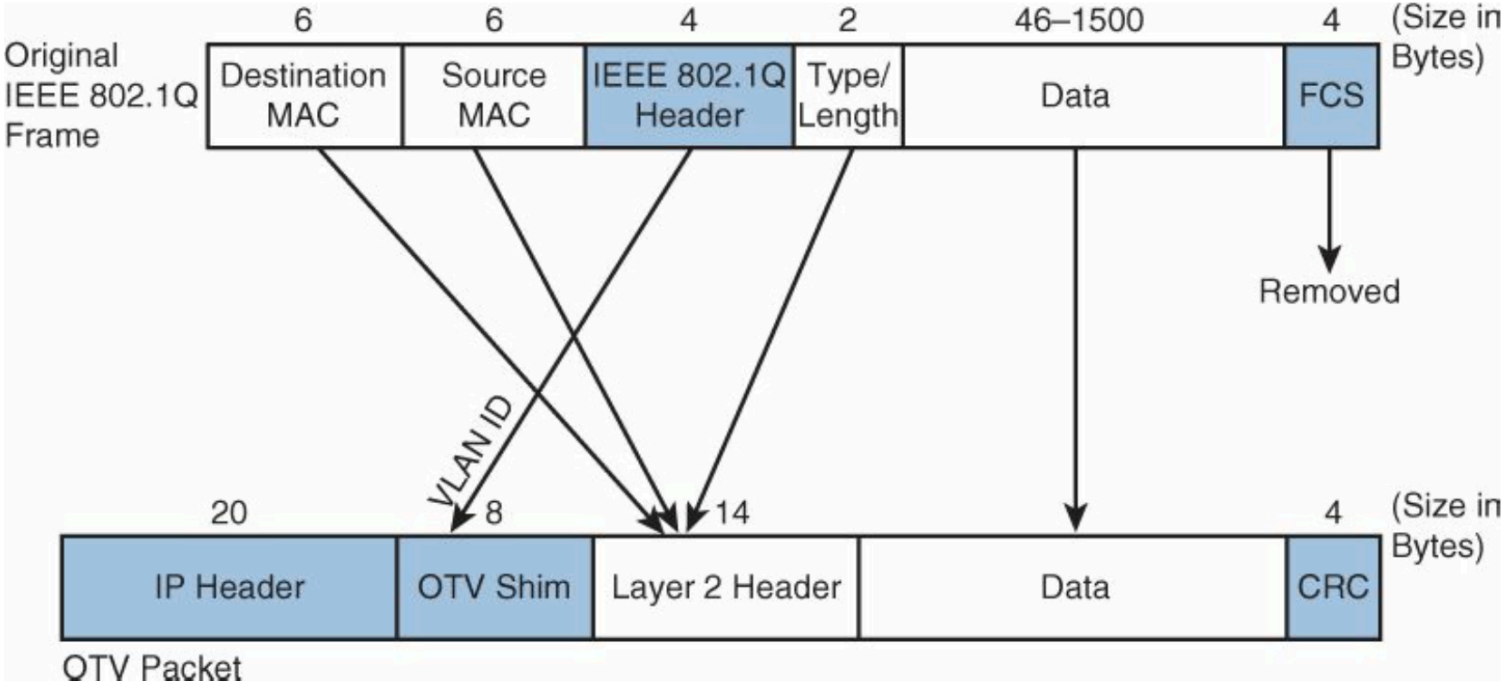
- *Overlay Transport Virtualization*
- Solución propietaria de Cisco
- Conectividad Ethernet a través de una red IP
- No hace aprendizaje de direcciones MAC en el plano de datos
- Emplea IS-IS para intercambiar esa información



G.A.A.Santana,"Data Center Virtualization Fundamentals", Cisco Press

OTV

- Genera paquetes con DF=1 conteniendo una sola trama Ethernet
- La MTU en la red IP debe poder transportar ese paquete IP
- No transporta BPDUs así que aísla los dominios STP



G.A.A.Santana,"Data Center Virtualization Fundamentals", Cisco Press

DNS based Site selection

Balanceo mediante DNS

- *Content routing, request routing, Global Server Load Balancing (GSLB)*
- El "site selector" actúa como servidor de DNS para los clientes
- Monitoriza el estado de los servidores, probablemente a través de los balanceadores
- Servidores de DNS ("*site selectors*") en ambos CPDs
- Usuario emplea un Proxy DNS (acepta *query* recursiva)
- Al resolver el dominio obtiene las direcciones de los dos servidores
- Mide el tiempo de respuesta a ambos y se queda con el menor
- Es decir, se harán las peticiones al topológicamente más cercano

Balanceo mediante DNS

- Los clientes se repartirán por proximidad
- Estos servidores darán la VIP del servicio local
- También la del otro CPD, para ofrecer redundancia
- El cliente entonces tiene las dos direcciones, aunque normalmente empleará la primera que encuentre en la respuesta
- Si el proxy hiciera las peticiones unas veces a un servidor y otras al otro para el mismo cliente
  - Si la petición viene del mismo cliente, tal vez tras bastantes minutos, le está enviando al otro CPD
  - Si existía una sesión fallará, pues la sesión estaba en el anterior CPD
  - Es un problema de *stickines*

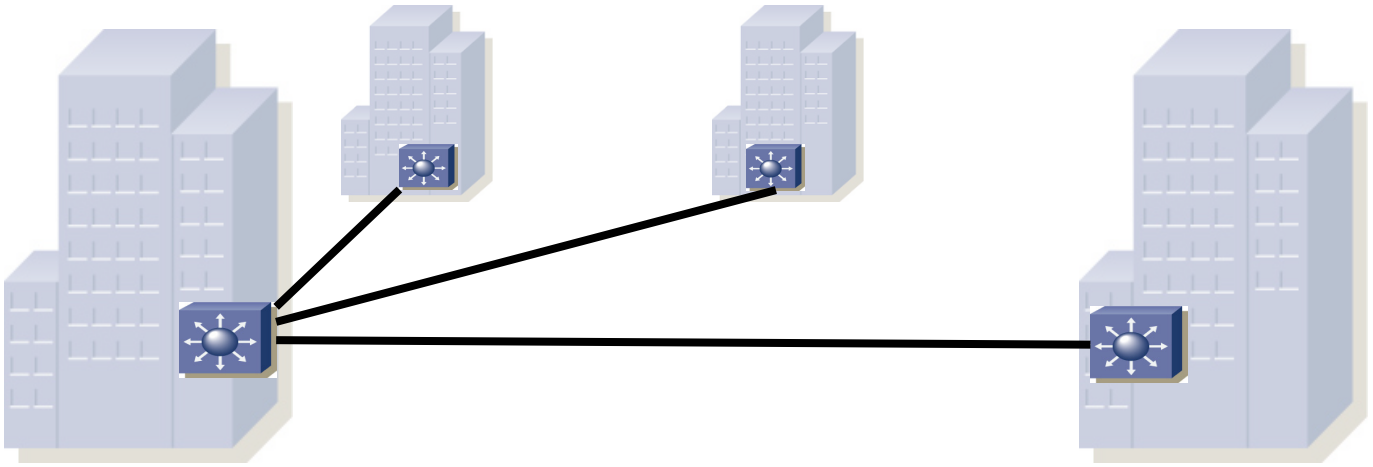
DNS y stickiness

- Activo-backup
  - Seleccionar un CPD como activo
  - Se encamina a los clientes al otro solo cuando falla el primero
  - Los servidores de ambos CPD devuelven la VIP del mismo
  - Se puede repartir carga si se atiende a varios FQDNs
- Source IP Hash
  - Devolver siempre la misma dirección IP al mismo proxy
  - Eso balancea solo en la medida en que las peticiones de DNS de los usuarios vengan de diferente proxy

WAN optimization

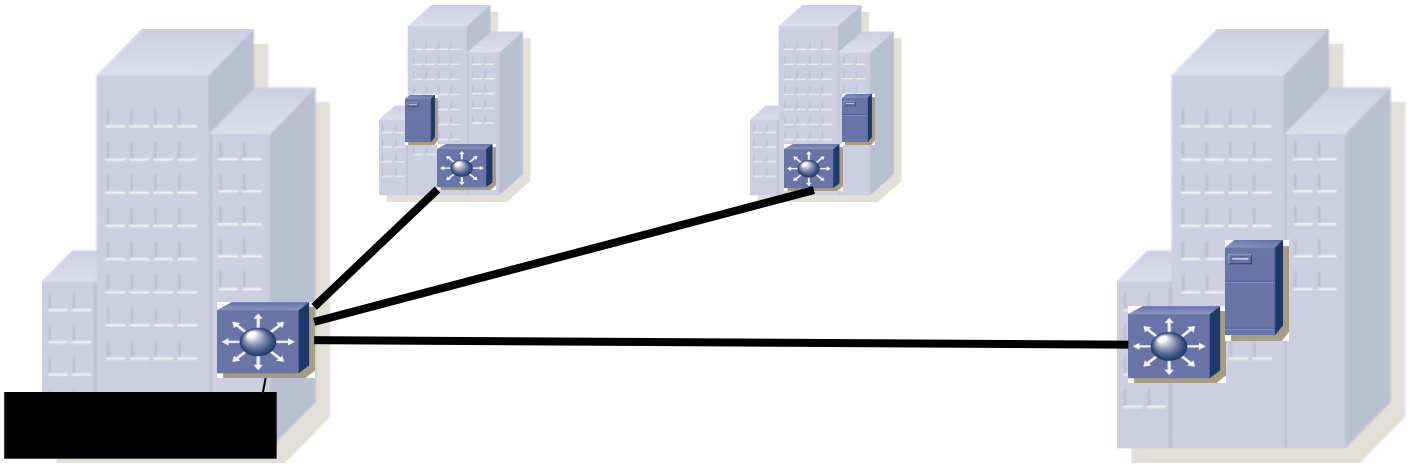
Rendimiento en la WAN

- Muchos protocolos y aplicaciones se han desarrollado para el entorno LAN
- Al emplearlos en el entorno WAN se encuentran con limitaciones debidas a:
  - Bandwidth: la aplicación funciona bien… con enlaces a 100Mbps o a 1Gbps
  - Latencia: la hemos probado en LAN… con RTT de 1-2ms
  - Pérdidas: infrecuentes en la LAN y con RTT pequeño se recuperan rápido
  - Servidor cargado: el de pruebas solo tenía al desarrollador pero el de producción tiene miles de usuarios



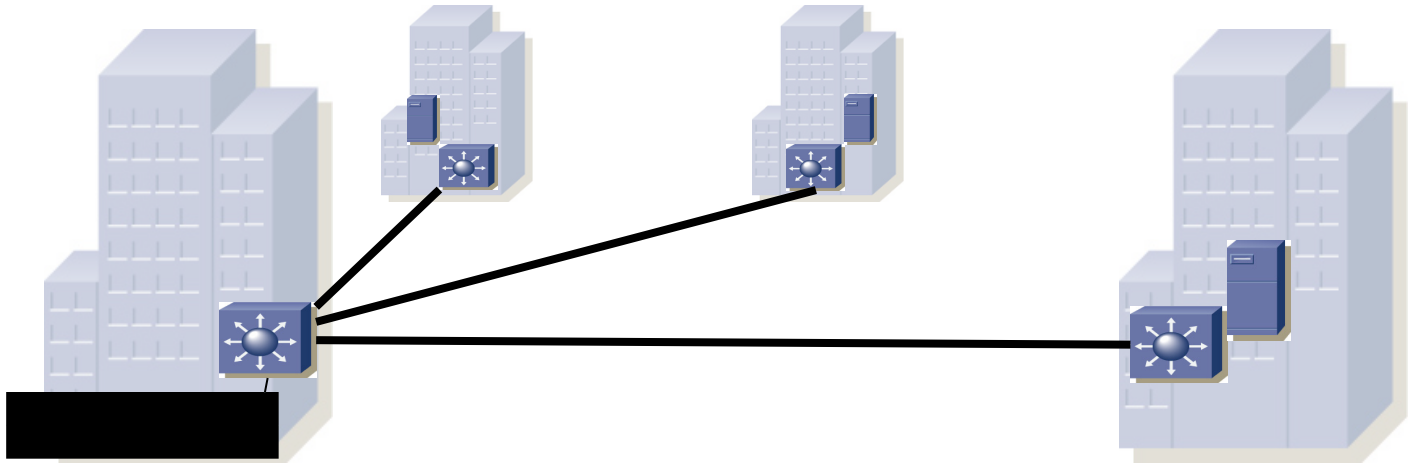
WAN vs LAN Bandwidth

- En los 90s accesos WAN de bajas velocidades (<512Kbps) y caros
- Una oficina remota puede no justificar el coste o simplemente no haber disponibilidad en esa región
- Los servidores tenían que estar cerca de los usuarios
- Eso quiere decir gran cantidad de servidores, más caro
- Más complicado gestionarlos, securizarlos, hacer backups, actualizaciones, etc
- Administración y mantenimiento más complejo y caro
- La aplicación debe compartir la capacidad con otras



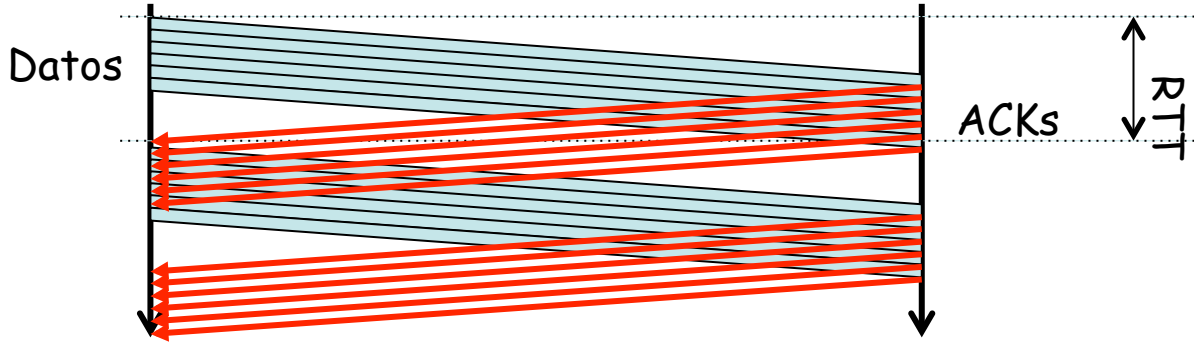
WAN vs LAN Bandwidth

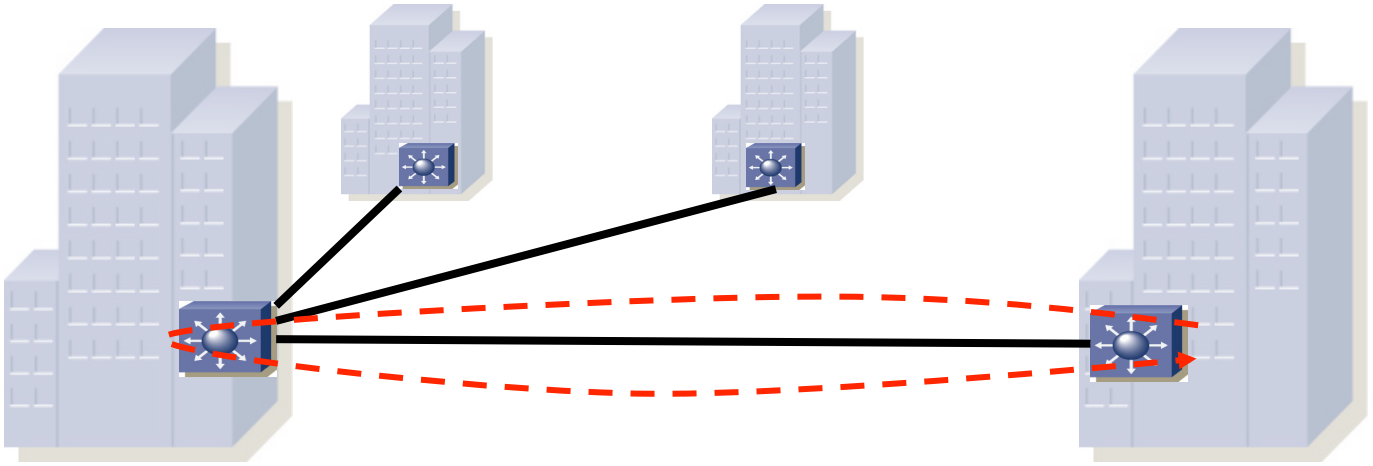
- Hoy en día accesos de baja o alta velocidad según la localización o las necesidades
- DSL, cable modem, fibra, metro Ethernet, UMTS, satélite, etc
- Por el enlace WAN: acceso a ficheros, backups, e-mail, acceso a Internet, streaming, impresión, aplicaciones de gestión, aplicaciones empresariales, autentificación, sesión remota, etc.
- Intentamos centralizar los servicios
- Han aparecido equipos que buscan acelerar el comportamiento de las aplicaciones en base a modificar los flujos de aplicación
- Evidentemente son muy dependientes del tipo de aplicaciones y uso de las mismas



Latencia (retardo)

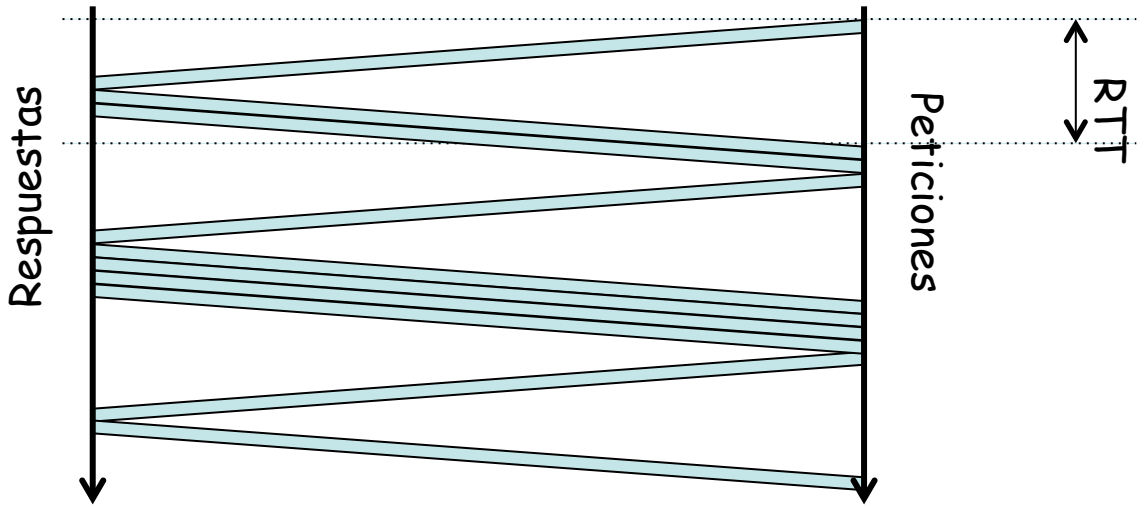
- El RTT tiene un impacto en la velocidad de transferencia y en la interactividad
- En protocolos de ventana deslizante estamos limitados por el tamaño de la ventana entre el RTT
- Esto no es solo TCP sino también protocolos de nivel de aplicación (por ejemplo SMB)
- Añadir capacidad a los enlaces no mejora el throughput (limitado por ventana)

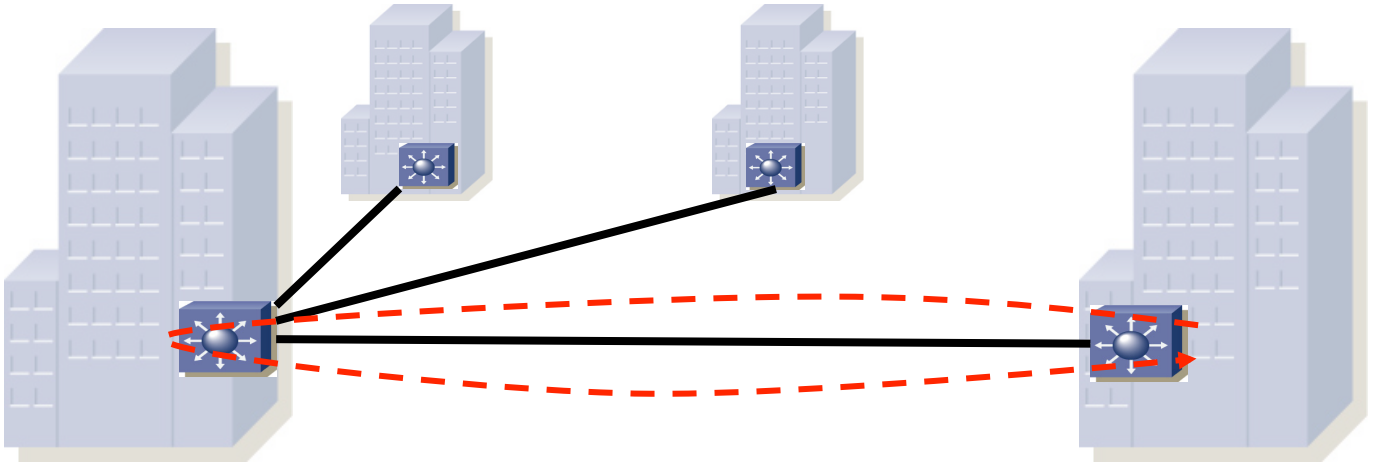




Latencia (retardo)

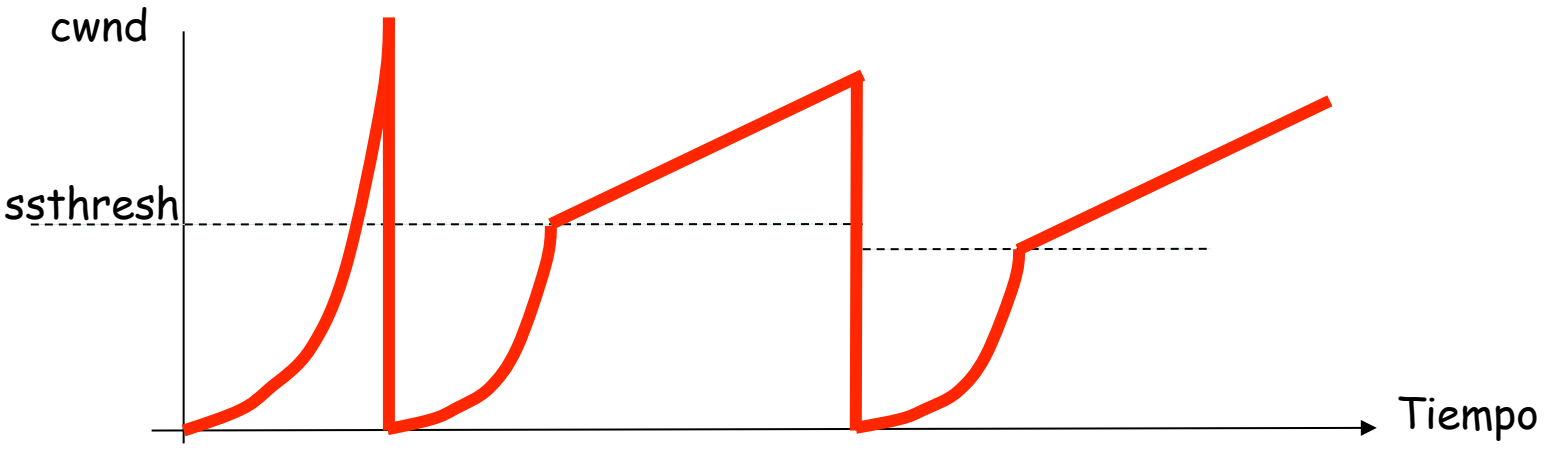
- Aplicaciones muy *chatty* consumen muchos RTTs con pocos datos
- No están limitadas por la ventana del protocolo sino porque necesita la respuesta a una petición antes de poder hacer la siguiente





Pérdidas (TCP)

- Si la ventana de control de flujo es suficientemente grande
- Y la transferencia dura suficiente
- TCP aumenta cwnd hasta congestionar el camino
- Se producen pérdidas y baja agresivamente la tasa de envío
- Y repite (más lento)

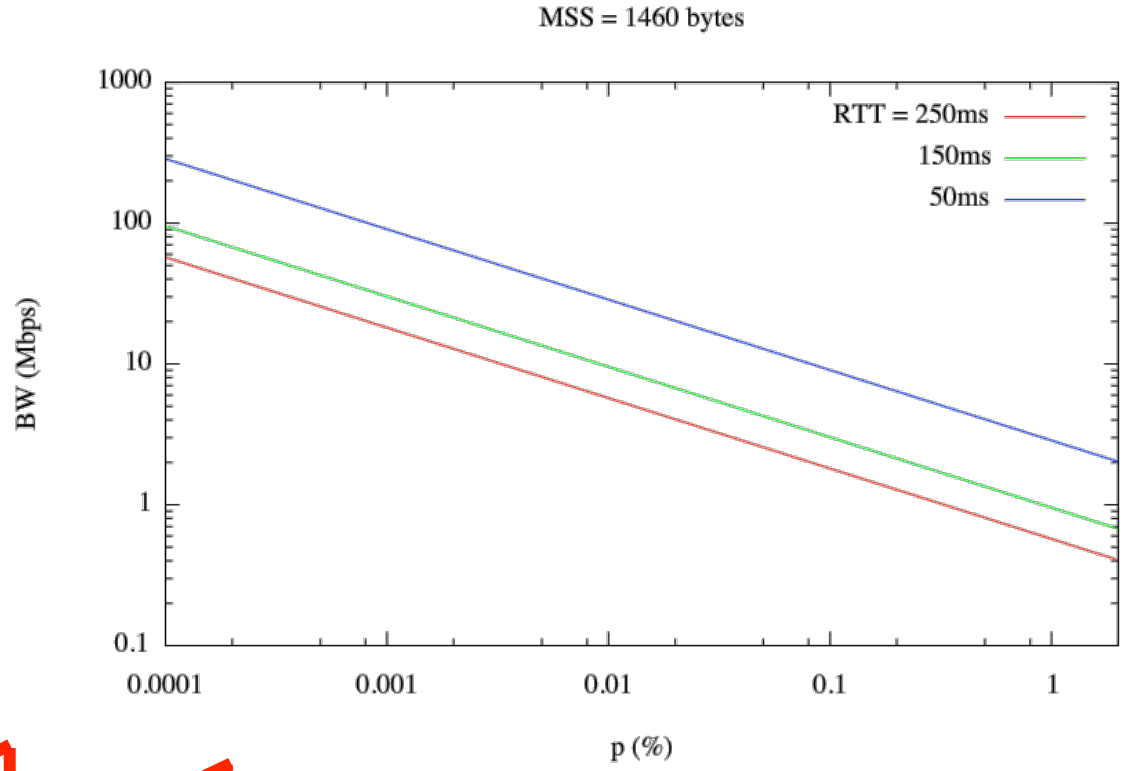


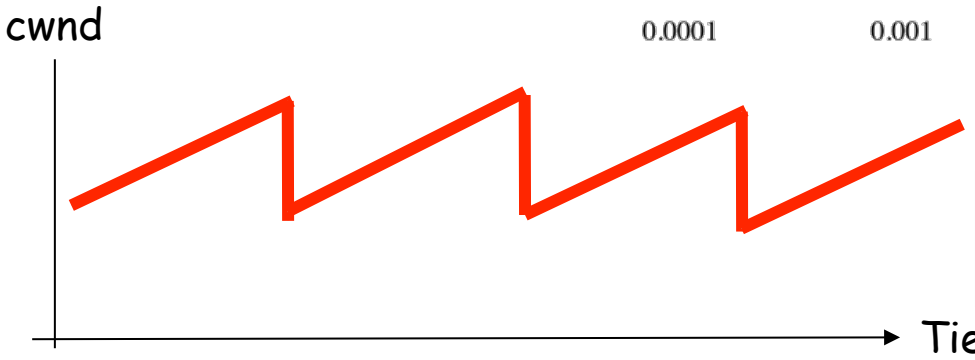
Pérdidas (TCP)

- Aproximación al throughput promedio para conexiones largas (TCP Reno) con una tasa *p* constante y pequeña de pérdidas
- TCP es muy sensible (agresivo) ante las pérdidas

$$BW = \frac{MSS}{RTT}\sqrt{\frac{3}{2p}}$$

M. Mathis
(ACM SIGCOMM'97)





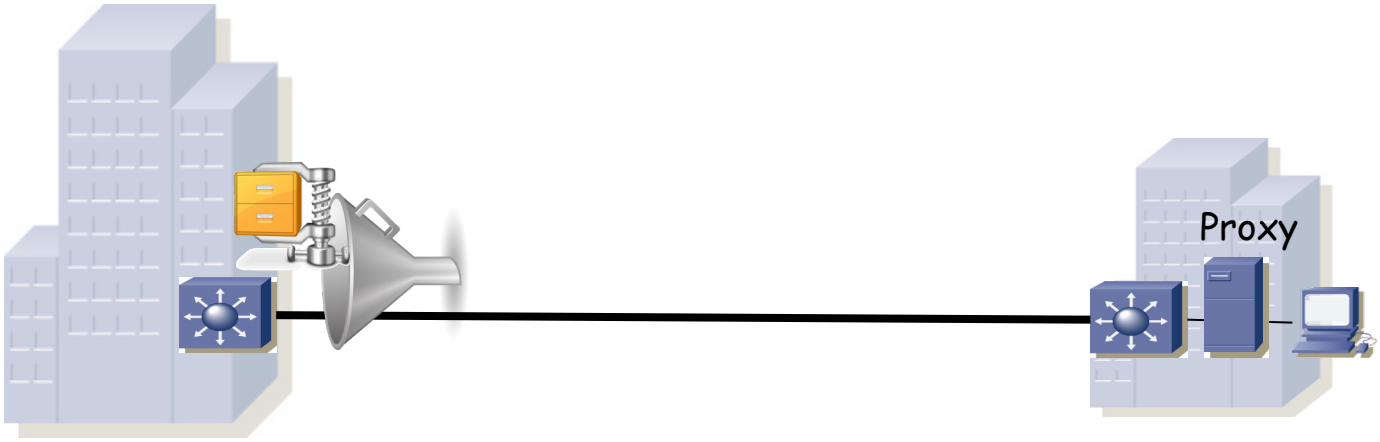
TCP Reno en
congestion avoidance

WAN optimization

- Técnicas/equipos que pretenden aliviar los problemas de rendimiento anteriores
- Sin recurrir a “¡más madera!” (o “más bandwidth”)
- Que como vemos ni siquiera es siempre la solución

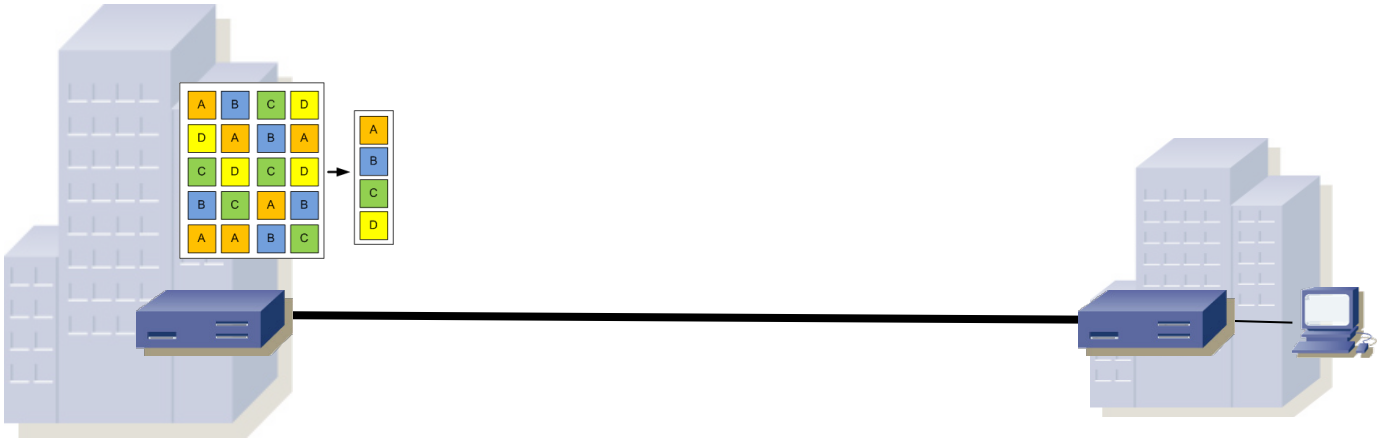
Compresión

- Compresión sin pérdidas de los datos a enviar
- Por ejemplo, si el navegador lo soporta, el servidor web puede enviar comprimido
- En caso de no aceptarlo el cliente, podría ser anunciada la opción por un proxy
- Según las aplicaciones, puede ser útil en ambos sentidos



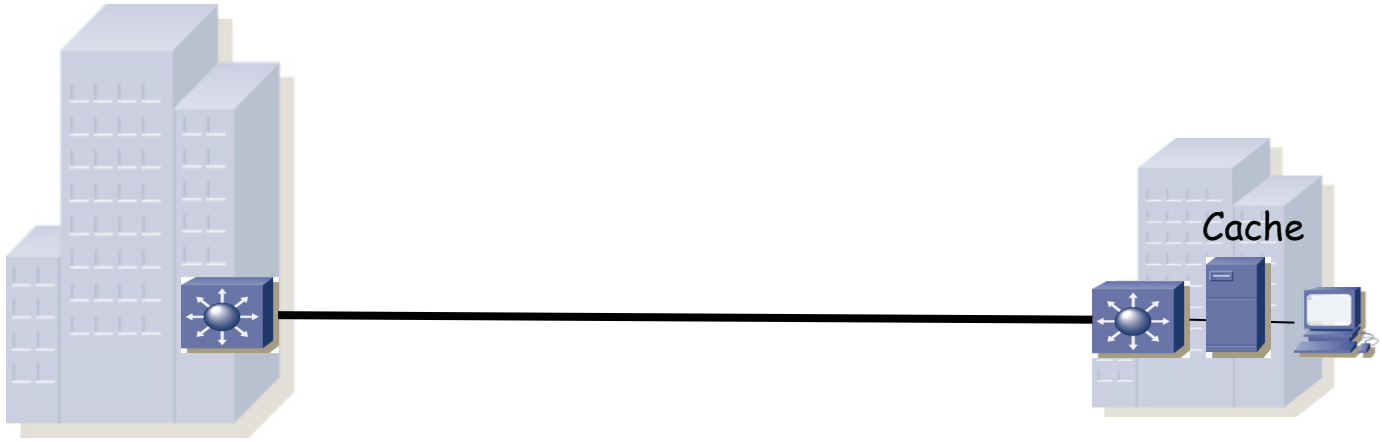
Data deduplicación

- Detectar patrones que se han enviado antes
- No enviarlos de nuevo sino un identificador
- El otro extremo recrea el patrón
- Puede ser a nivel de no volver a enviar el mismo fichero
- O detectar cambios en trozos en el mismo y solo enviar los cambios
- Puede ser entre diferentes aplicaciones: por ejemplo descargar un fichero para luego adjuntarlo a un e-mail
- No es sencillo por los cambios en los protocolos de transporte y aplicación
- Los firewalls no son útiles con el tráfico desduplicado



Latencia: Caches

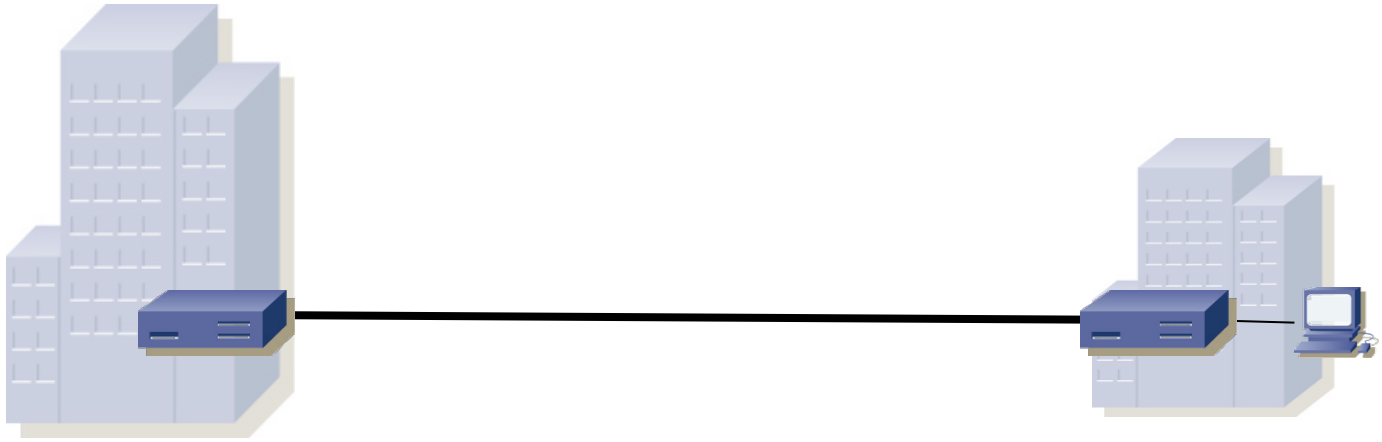
- Sencillas para protocolos como HTTP que están diseñados con ellas en mente
- Factibles pero más complejas para protocolos de compartición de ficheros (por ejemplo SMB)
- Caches para diferentes aplicaciones pueden acabar manteniendo los mismos objetos, ejemplo:
  - Usuario descarga un fichero de un email
  - Lo carga a un directorio compartido (SMB)
  - Lo sube a una web (HTTP)



Aceleración

Application Acceleration

- Conoce el protocolo de aplicación
- Se anticipa a las acciones del usuario
- Por ejemplo puede hacer un *read-ahead* cuando la aplicación está leyendo un documento
- O puede conocer las secuencias típicas de mensajes e intentar optimizarlas



Aceleración

TCP acceleration

- Equipos con versiones antiguas o no optimizadas de TCP, las ecualiza
- El optimizador puede modificar el tráfico implementando nuevos mecanismos de control de congestión, mayores valores de ventana (*virtual window scaling*), etc
- O por ejemplo rompiendo la conexión en tres
- Tiene más información pues ve el tráfico total
- Con deduplicación y compresión en el segmento WAN