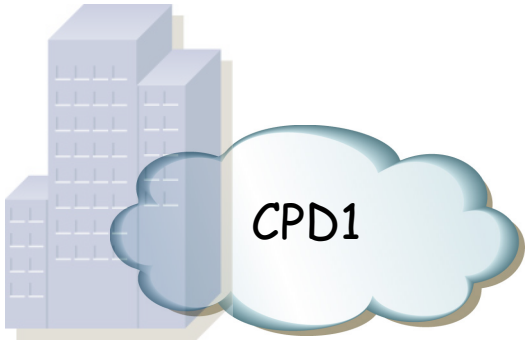
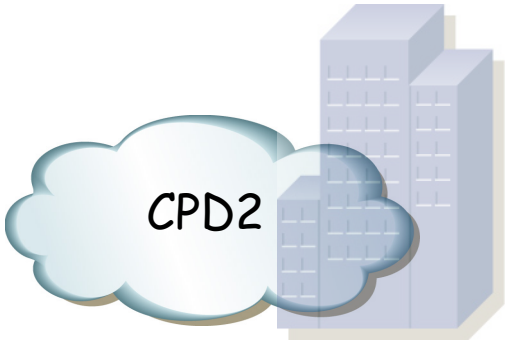
Interconexión de DCs: Introducción

Múltiples DCs

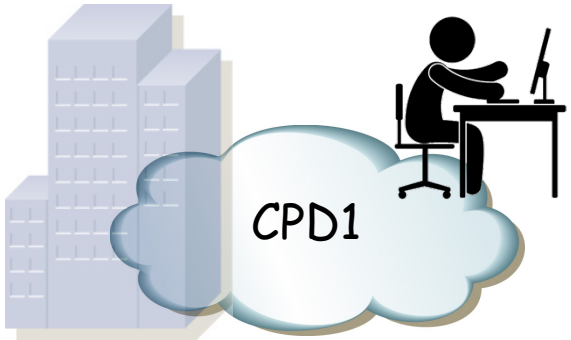
- La palabra clave es “disponibilidad” (*availability*)
- Buscamos protección ante desastres:
  - Tsunamis, huracanes, inundaciones, terremotos, incendios
  - Fallos de larga duración de la red eléctrica (*black-outs*)
  - Violaciones de seguridad
- No es solo una cuestión de disponibilidad física sino que la lógica para coordinarlos debe funcionar correctamente también

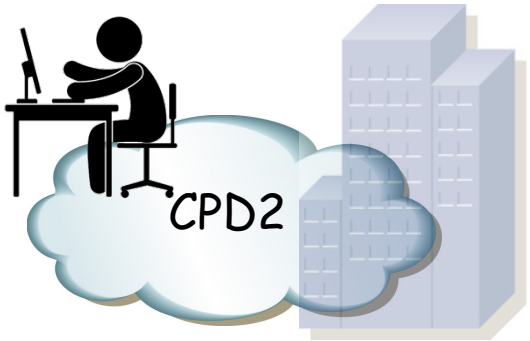




Múltiples DCs

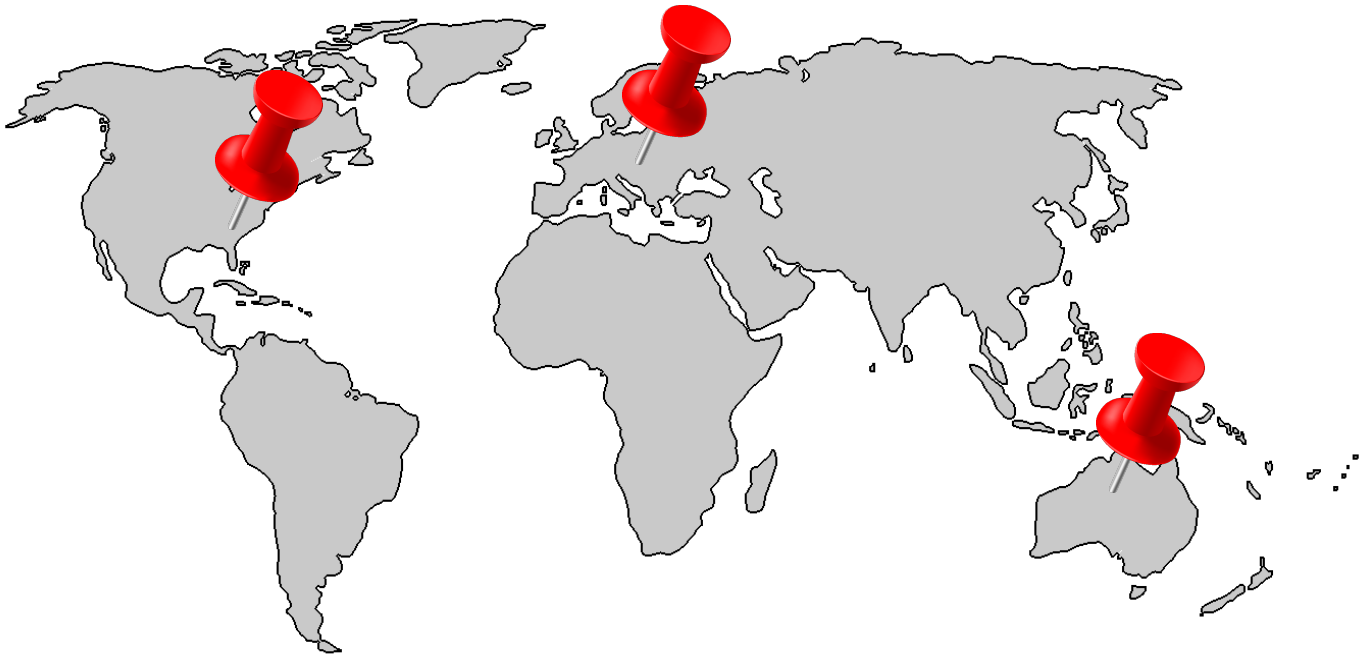
- Pueden trabajar por parejas en modo activo-standby
  - Uno de ellos cursa todas la carga de trabajo
  - El segundo monitoriza el estado del activo
  - Operaciones que modifiquen datos almacenados se sincronizan con el almacenamiento en el de respaldo
- Pueden trabajar en modo activo-activo
  - Necesitamos técnicas de reparto de carga entre los DCs
  - Así como de nuevo técnicas para sincronizar los datos entre ellos





Ubicación de DCs

- Alejados para que un problema “geográfico” no afecte a ambos
- Sin embargo podemos toparnos con limitaciones de retardo máximo para las aplicaciones distribuidas
- Por ejemplo la *replicación síncrona* se basaba en devolver confirmación de haber almacenado el dato cuando se ha escrito en las dos cabinas
- Si están en DCs alejados esto afectará al retardo de transacción
- Eso limita la distancia para reducir el tiempo de respuesta
- También protocolos como FC deben ajustarse para altos retardos (mayor RTT requiere mayor número de créditos para sacar provecho al BW)

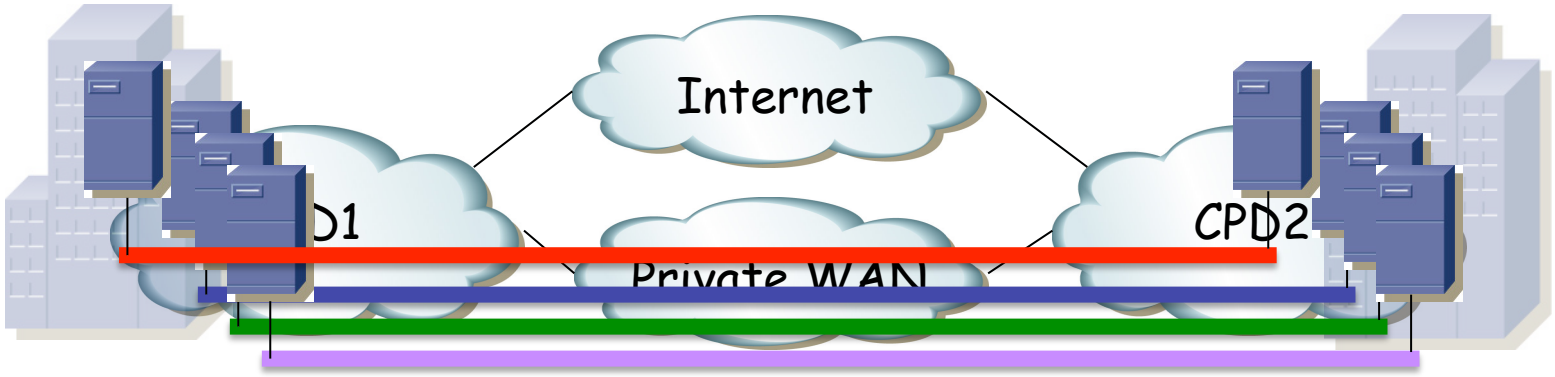


Múltiples DCs o sedes

- Habitualmente la interconexión se recomienda en capa 3
- Eso limita los problemas de capa 2 a cada DC
- Sin embargo muchas aplicaciones con funcionalidades de clustering requieren adyacencia en capa 2
  - Heartbeats o información de estado se envía multicast/broadcast
  - Nodos que comparten dirección IP y dirección MAC
- La movilidad de servidores (físicos o virtuales) requiere mantener la pertenencia a la misma VLAN
- O el crecimiento nos puede llevar a otro edificio
- Es decir, podemos necesitar extender las VLANs entre DCs
- Todo esto aplica tanto a interconexión de CPDs como de sedes remotas

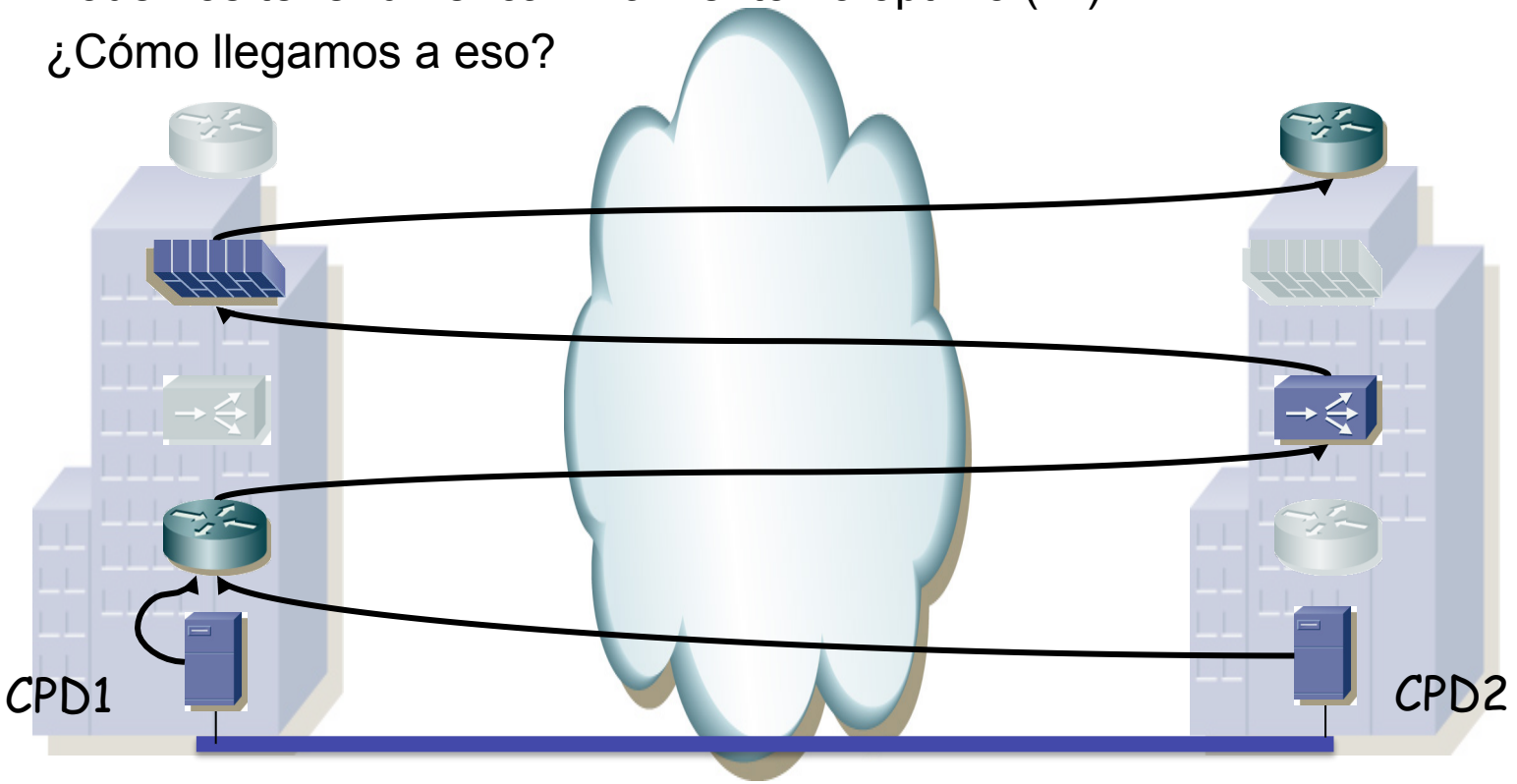
Multi-tenancy

- Múltiples clientes (miles) en un data-center
- Cada cliente debe tener una visión de la infraestructura como si estuviera solo y tuviera control total
- Virtualización en la red permite separar la red de esos usuarios



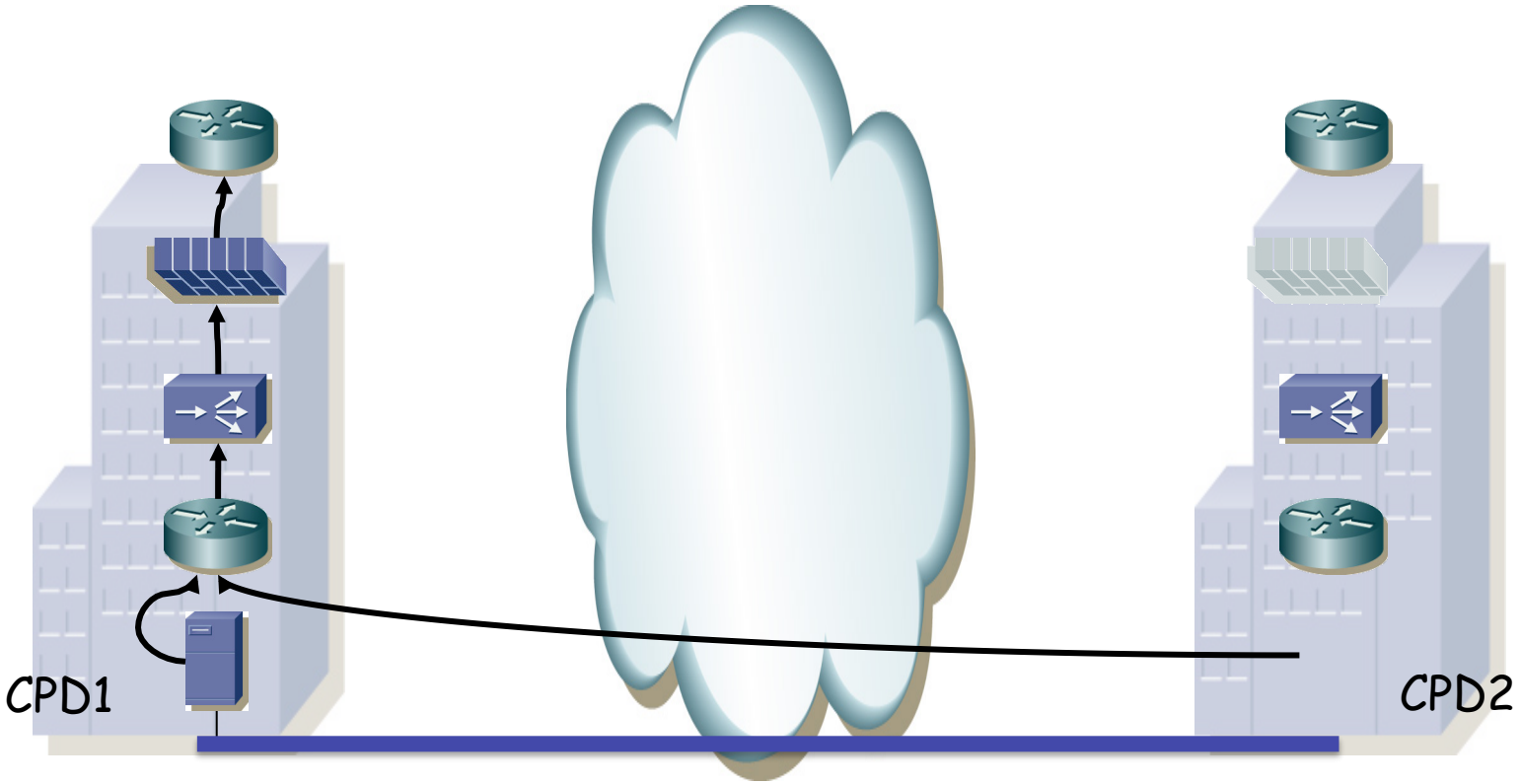
Problemas con extensión L2

- Entre los DCs hay que controlar el Broadcast, Unknown unicast y Multicast (BUM)
- ¿STP?
  - Problemas de escalabilidad
  - Fallo en la raíz afecta a los dos DCs
  - Si hay más de una interconexión seguramente desactive una
- Podemos tener un encaminamiento no óptimo (…)
- ¿Cómo llegamos a eso?



Tromboning

- ¿Cómo llegamos a eso?
- Por ejemplo porque hemos movido una máquina virtual (…)
- Una posible solución es emplear un FHRP entre los routers de ambos CPDs
- Y cortamos el tráfico del FHRP entre los CPDs para que ambos routers se elijan como maestros (activo/activo)

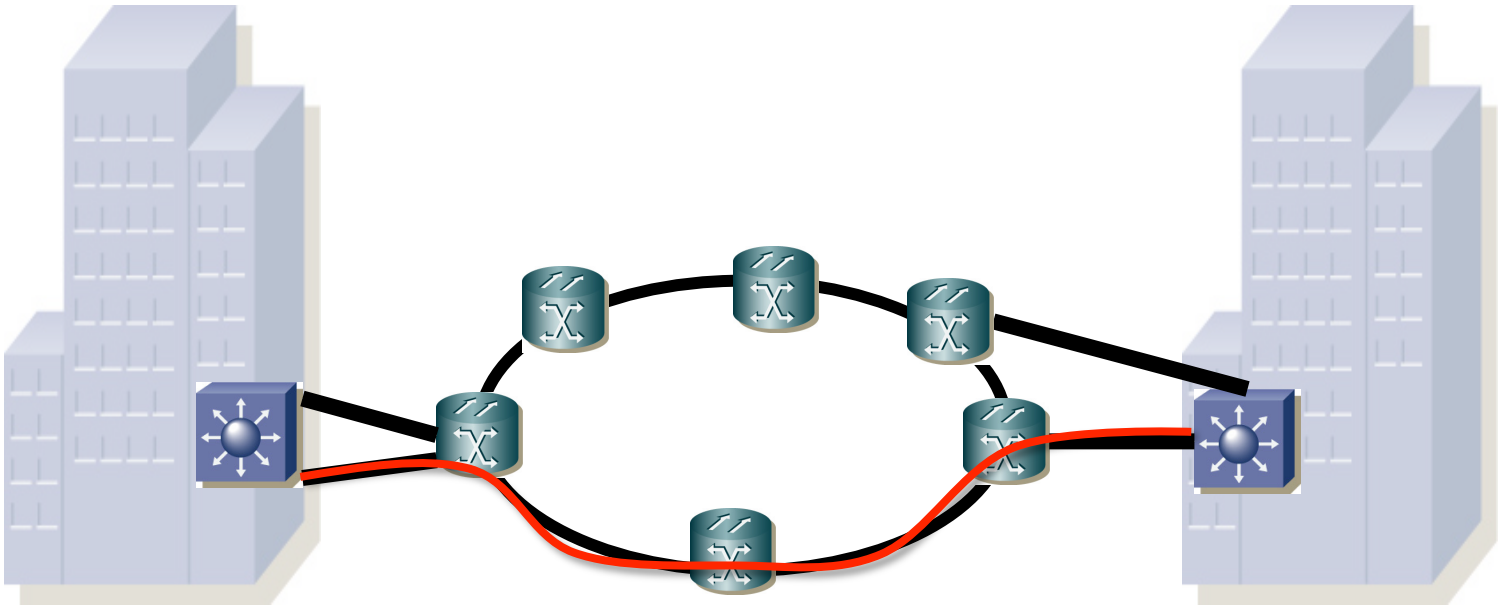


Interconexión del *storage*

- La interconexión entre los DCs puede ser solo para sincronizar el almacenamiento
- "SAN extension"

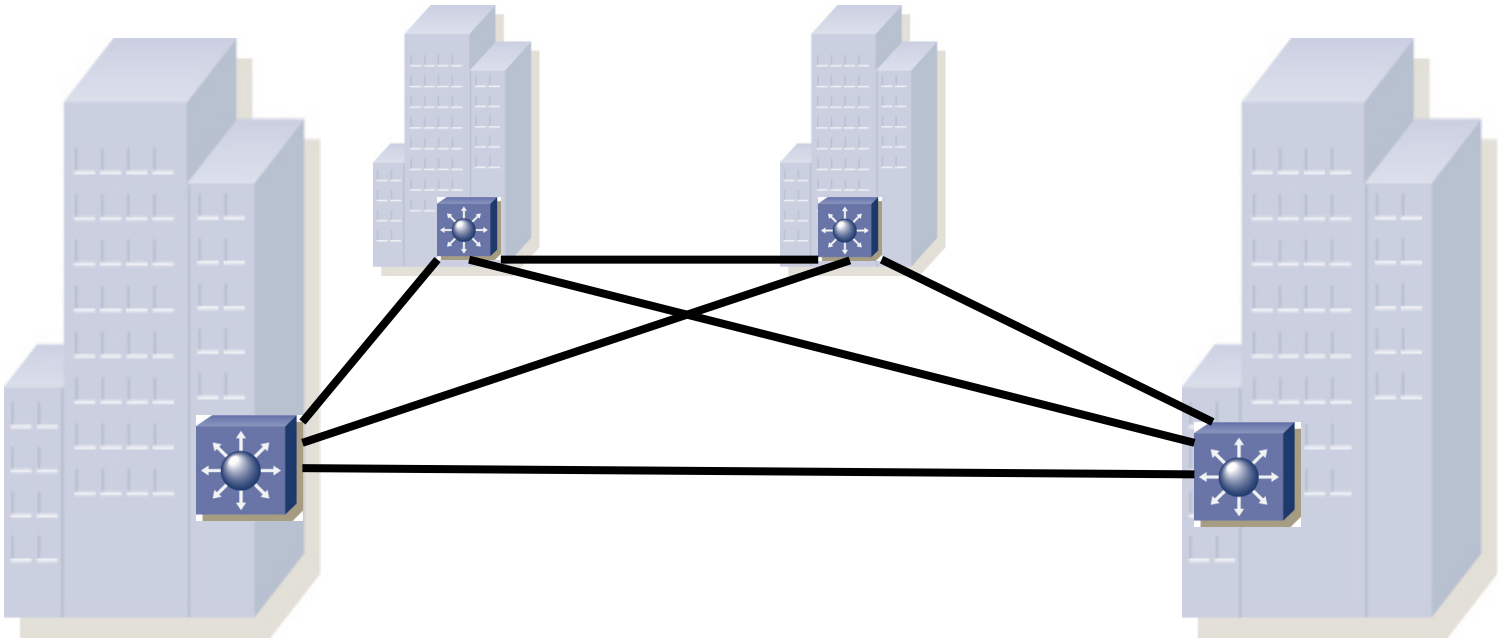
Interconexión por fibra

- Se puede emplear *fibra oscura*
- Puede transportar múltiples wavelengths (CWDM, DWDM)
- O se podría transportar una o varias wavelengths por una red de conmutación óptica
- Esta red puede dar protección
- La distancia sigue limitada pues da continuidad óptica (no hay OEO)
- Y es probable que queramos redundancia en el acceso a ella



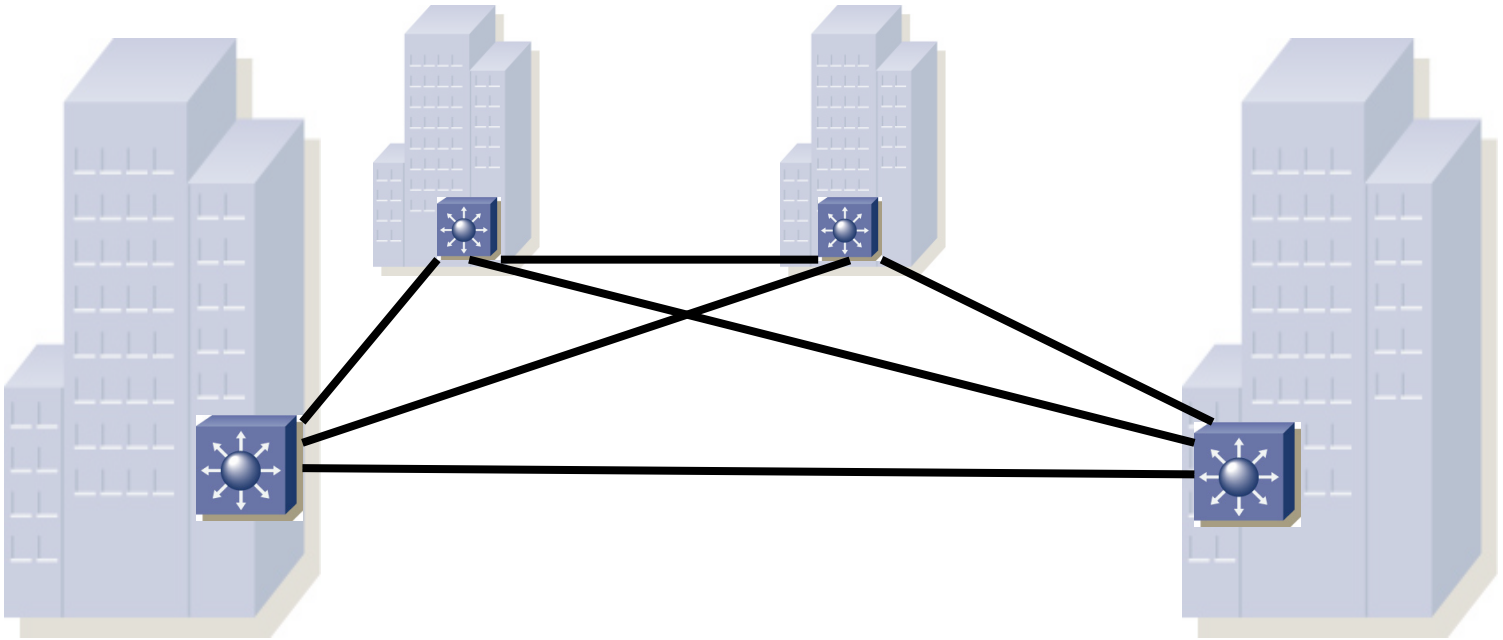
Múltiples *Sites*

- Independientemente de cómo se resuelva el transporte WAN
- ¿Cómo queda la interconexión "física"?
- Podemos tener un esquema *Hub&Spoke*
- También podemos tener un *mesh*
- En este caso hay que resolver esos bucles (STP, SPB, TRILL)
- Estamos hablando de *point-to-point VPNs*



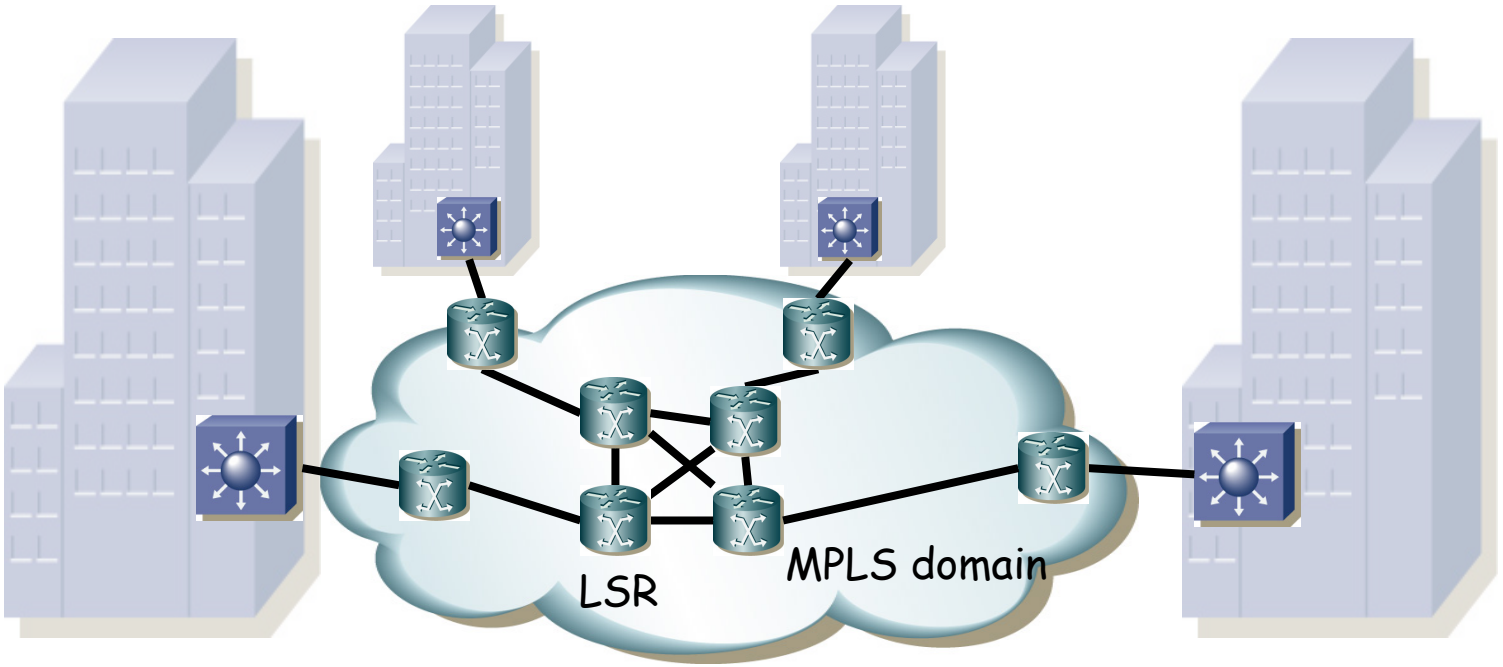
Circuitos

- Estos enlaces, en lugar de wavelengths, pueden ser algún tipo de "circuitos" o "circuitos virtuales"
  - SONET/SDH
  - ATM
  - Frame Relay
- Cualquiera de ellos permite transportar Ethernet o IP
- Serían L2 VPNs o L1 VPNs
- Hoy en día es habitual la solución MPLS



Interconexión MPLS

- En lugar de wavelengths o PVCs tenemos LSPs
- Recordemos que podemos encapsular Ethernet sobre MPLS (EoMPLS)
  - RFC 4448 “Encapsulation Methods for Transport of Ethernet over MPLS Networks”
- De hecho se suele decir que tenemos “AToM” o “Any Transport over MPLS”
- Los equipos de usuario van a poder ser capa 2 o capa 3

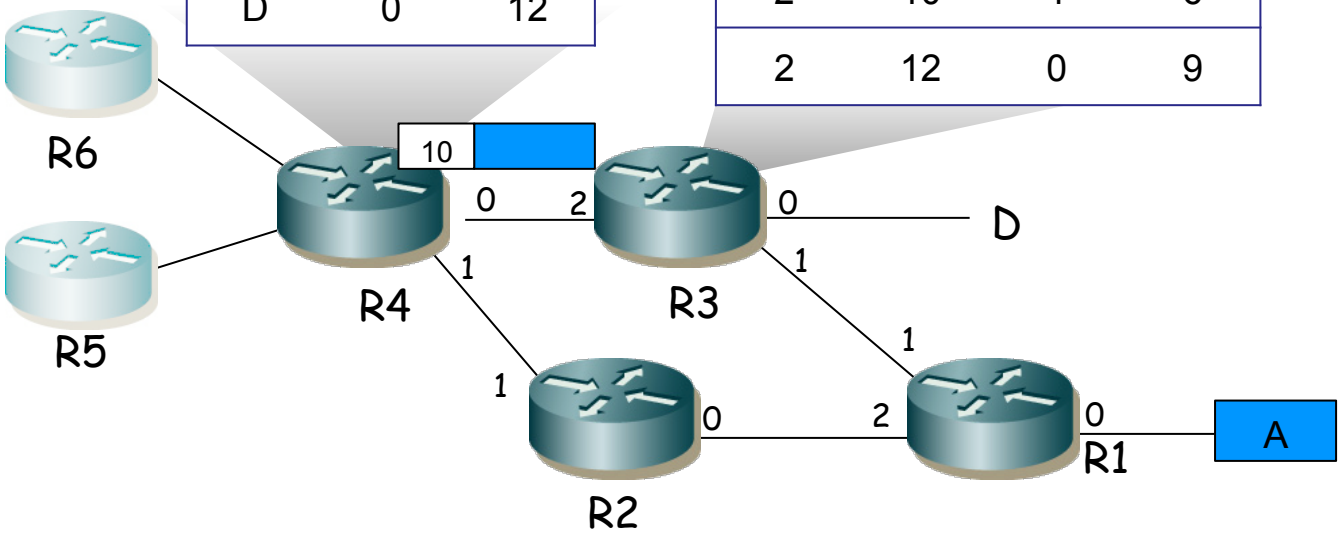


MPLS (recordatorio)

MPLS "forwarding"

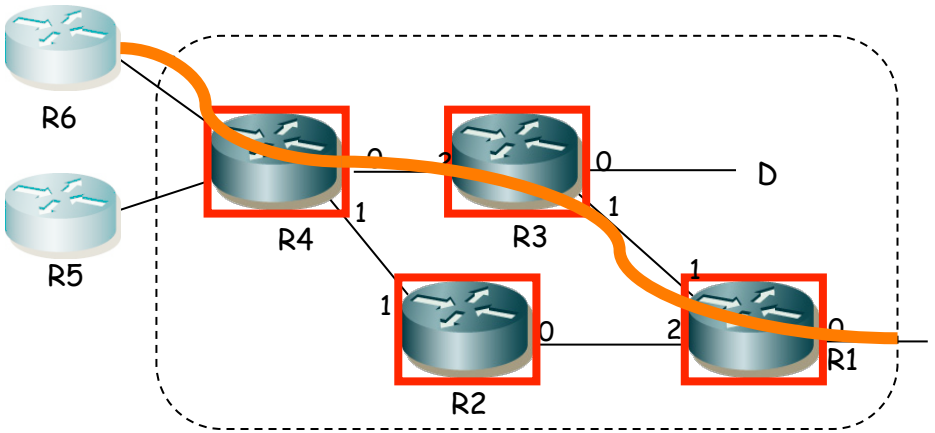
Dst	Out iface	Out label
A	0	10
D	0	12

In iface	In label	Out iface	Out label
2	10	1	6
2	12	0	9



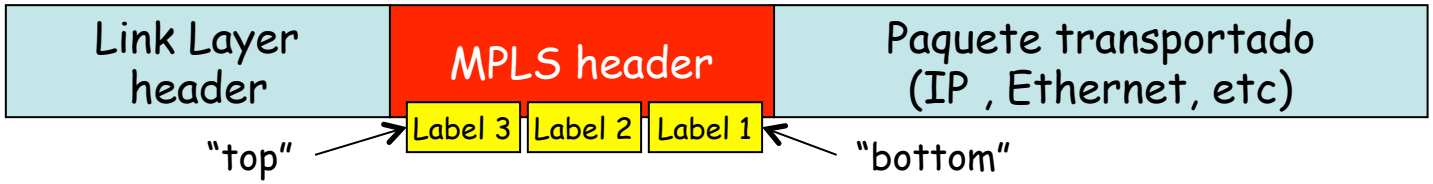
Terminología

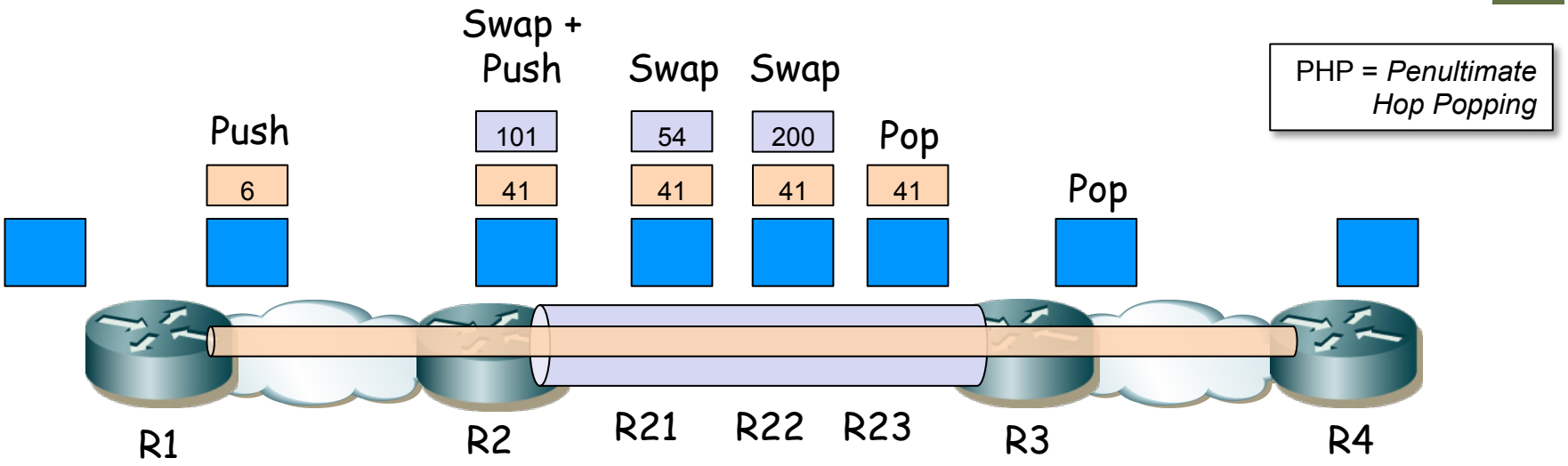
- "MPLS domain": conjunto contiguo de nodos MPLS bajo una misma administración
- "MPLS ingress node": nodo frontera de un dominio en su tarea como entrada de tráfico al mismo
- "MPLS egress node": nodo frontera de un dominio en su tarea como salida de tráfico del mismo
- "Label": etiqueta numérica, corta, longitud fija, identifica a un FEC localmente a un enlace
- "Label Switching Router (LSR)": nodo MPLS capaz de reenviar en base a etiquetas
- "Label Switched Path (LSP)": camino a través de LSRs



Label Stack

- La parte “superior” (“top”) de la pila comienza a continuación de la cabecera de nivel de enlace
- La parte “inferior” (“bottom”) de la pila está junto a la cabecera de nivel de red
- El procesado se basa siempre en la etiqueta exterior (“top”)

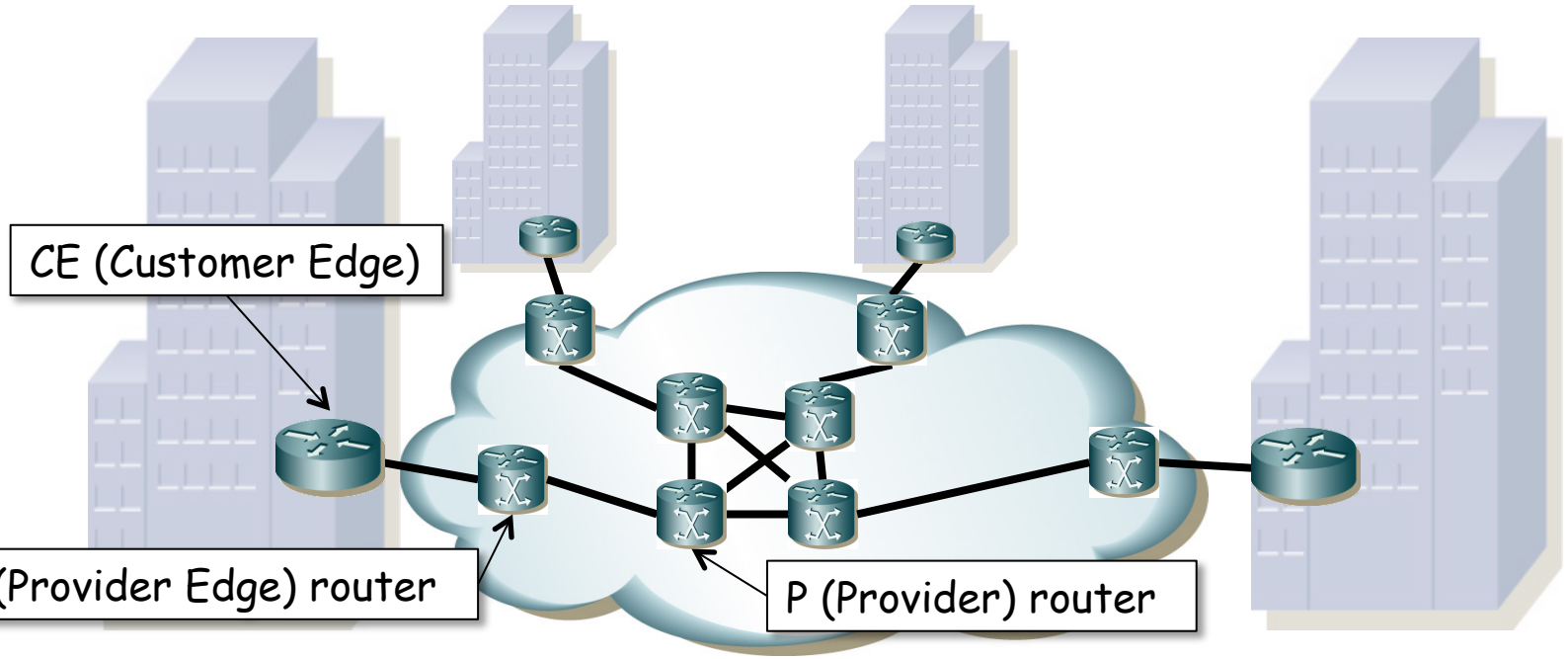




Layer 3 MPLS VPNs

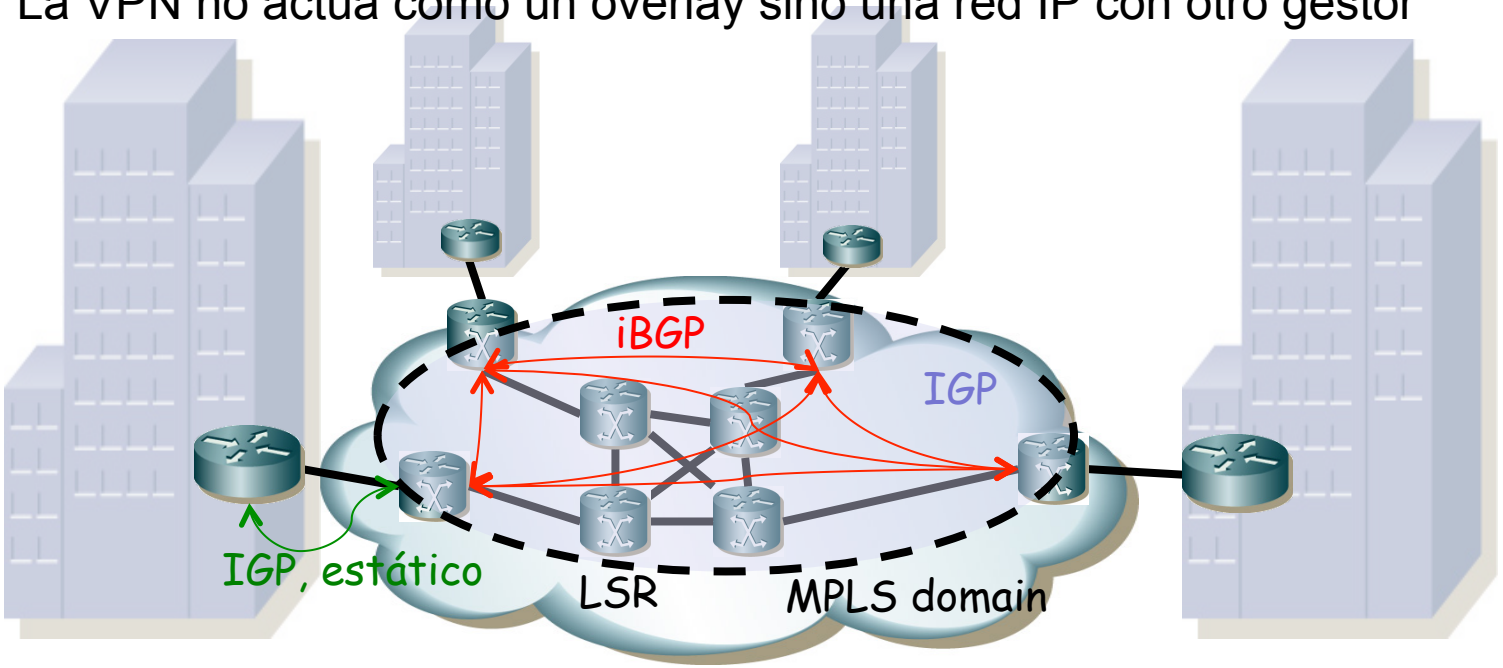
Layer 3 VPNs

- RFC 4364 “BGP/MPLS IP Virtual Private Networks (VPNs)”
- VPN para el transporte de paquetes IP entre sedes
- El backbone del proveedor de servicio es una red IP MPLS
- RFC 4760 “Multiprotocol Extensions for BGP-4”
- Extensiones a BGP-4 para poder transportar información de otros protocolos de nivel de red: IPv6, IPX, L3VPN, etc
- En este caso, en lugar de transportar rutas IPv4 transportará rutas “VPN-IPv4”



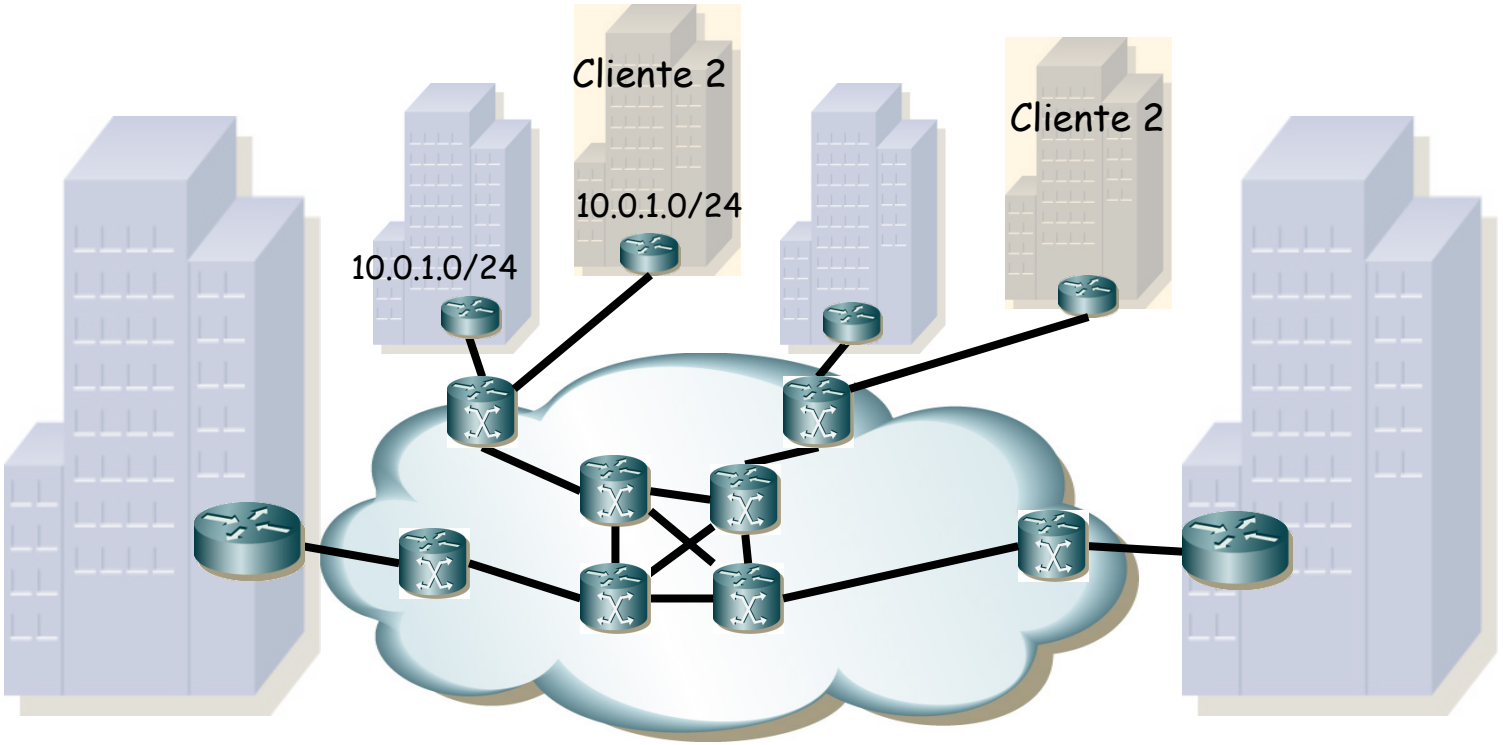
L3VPN: Routing

- Los CEs anuncian sus rutas a los PEs (con un IGP o rutas estáticas)
- Los PEs emplean MP-BGP para intercambiarse esas rutas (iBGP)
- El PE la distribuye al CEs del mismo cliente (de la misma VPN)
- Los P y PE corren un IGP para tener alcanzabilidad interna
- Los CE son routers convencionales, no necesitan ninguna configuración de VPN ni emplean MPLS
- Los CEs no intercambian información de routing entre ellos, no son adyacentes
- La VPN no actúa como un overlay sino una red IP con otro gestor



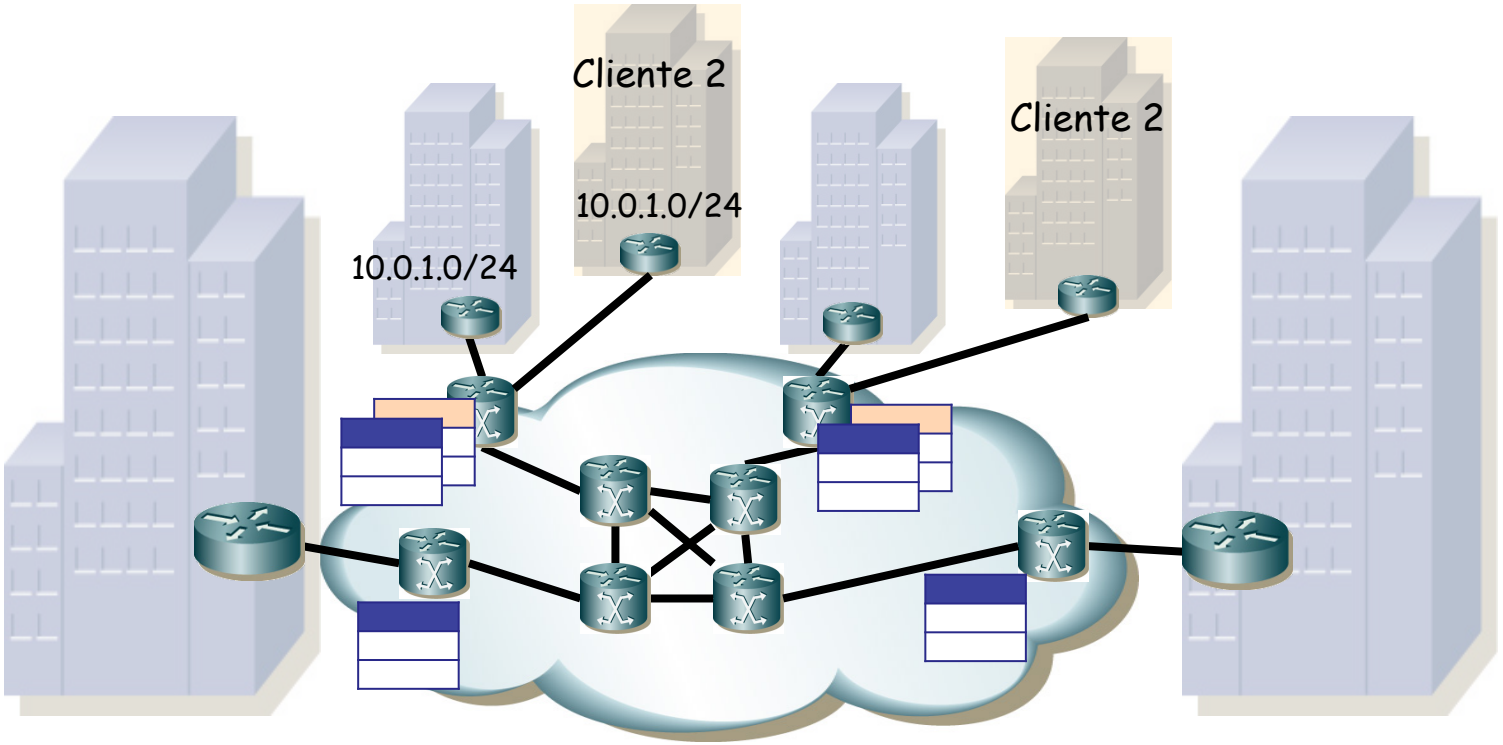
L3VPN: Routing

- Dos VPNs pueden emplear espacios de direcciones IP que se solapen
- Los anuncios VPN-IPv4 de esas subredes mediante BGP incluyen un identificador (*Route Distinguisher = RD, 8 bytes)* que las diferencia
- Cada service provider tiene su espacio de valores RD
- Los P no ven las rutas de las VPNs (evita problemas de escalabilidad)
- ¿Cómo se enruta si hay direcciones duplicadas y los routers centrales no ven esas rutas?



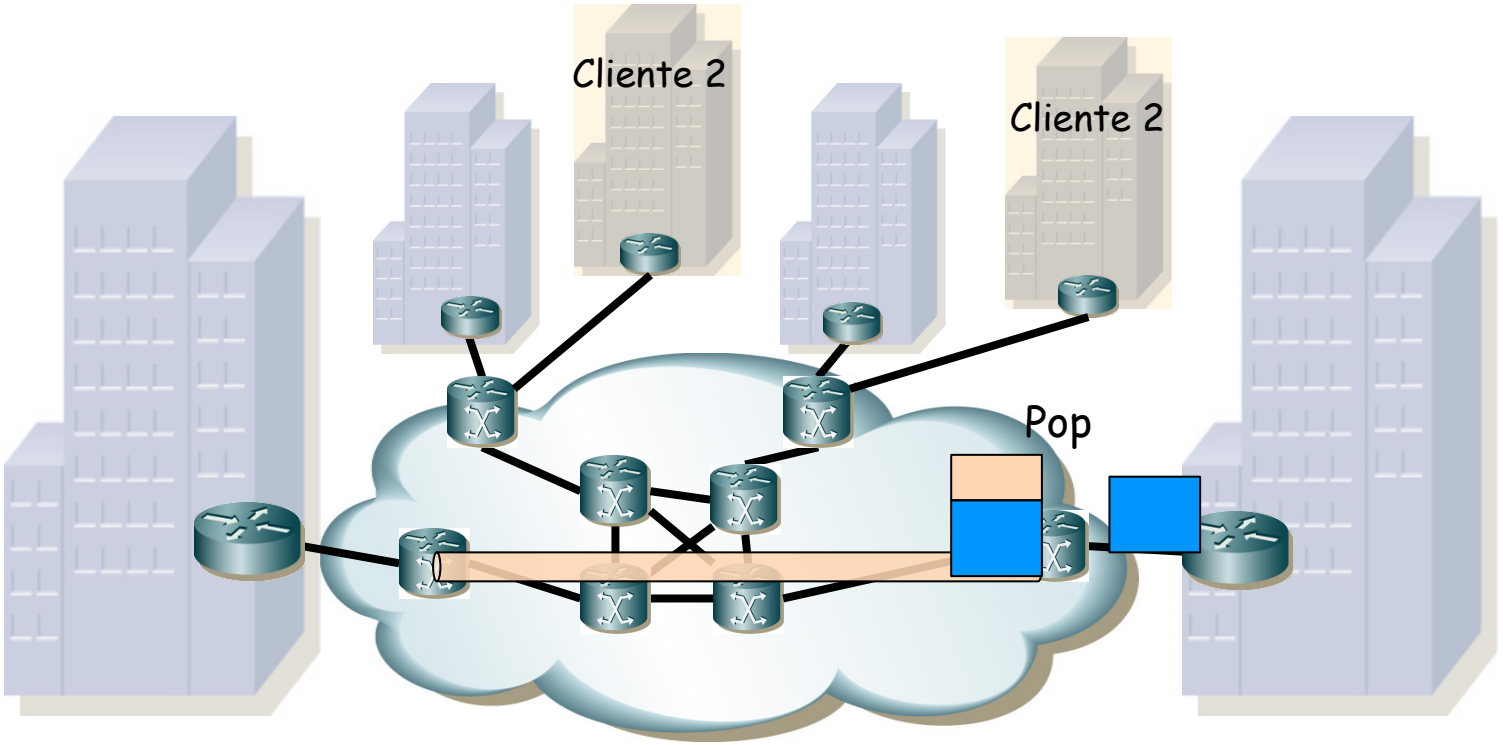
L3VPN: Forwarding

- Cada PE mantiene una tabla de rutas para cada VPN o *VPN/Virtual Routing and Forwarding tables (VRFs) y a*demás una tabla por defecto
- Cara VRF está asociado a un valor o más de *"Route Target" (RT)*
- Al recibir un paquete IP de un cliente consulta la VRF correspondiente
- Las rutas VPN-IPv4 se anuncian con un valor de RT a todas las VRF con ese valor
- Incluye una etiqueta MPLS



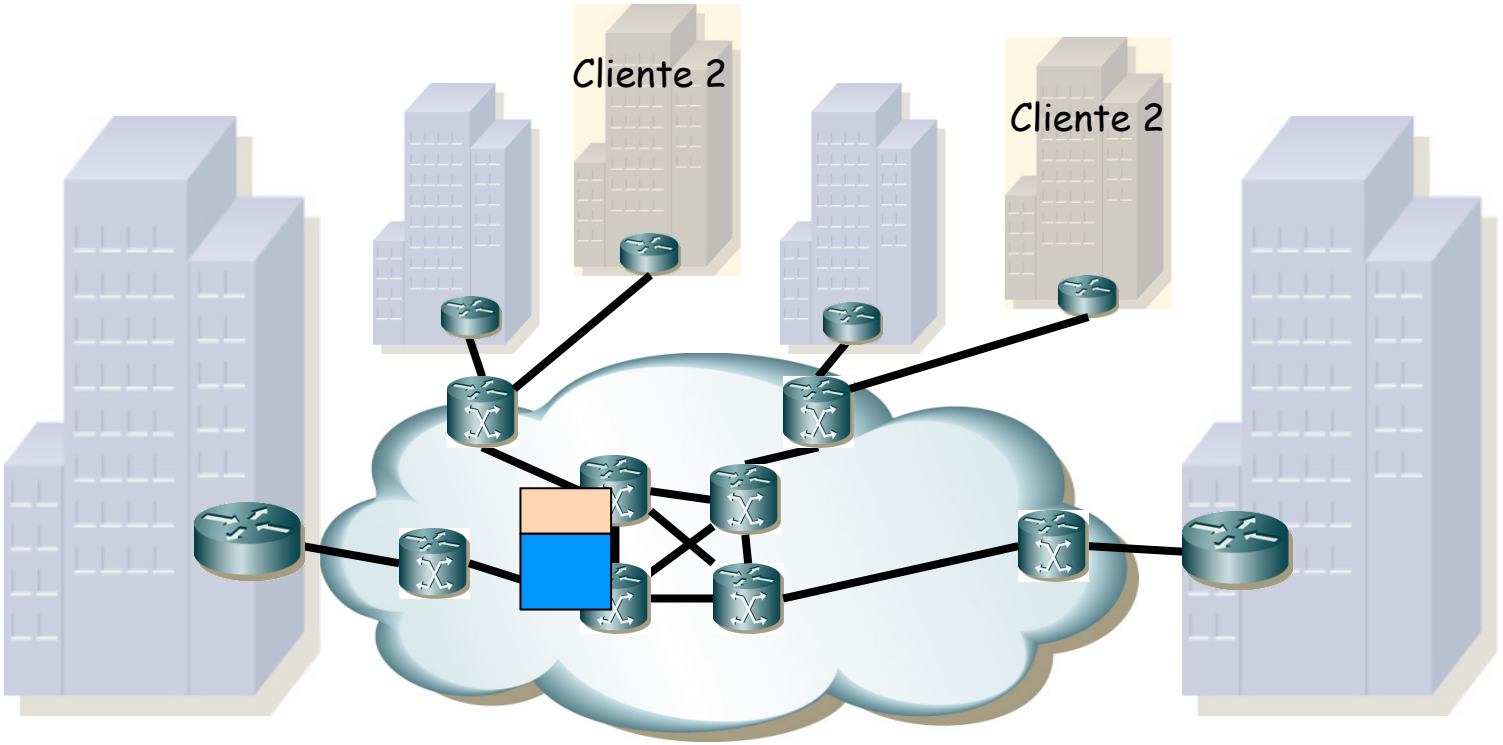
L3VPN: MPLS

- ¿Para qué esa etiqueta?
- Para que el PE de salida sepa a qué VRF pertenece el paquete
- No puede basarse en la dirección IP destino pues pueden estar duplicadas



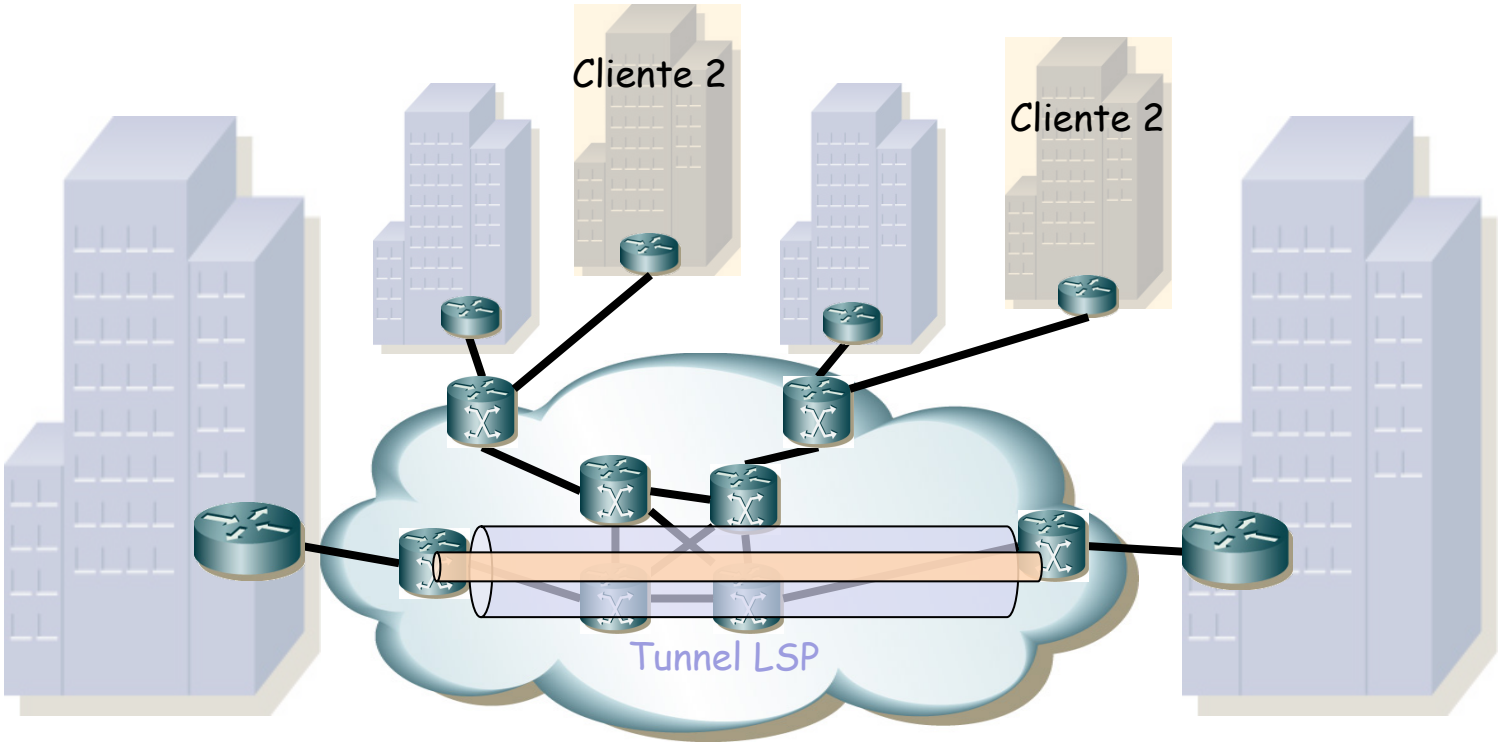
L3VPN: MPLS

- ¿Y en los P routers? ¿Reenvían en función de esa etiqueta?
- Tendríamos en ellos una gran cantidad de LSPs, para todas las VPNs
- Mala escalabilidad
- (…)



L3VPN: MPLS

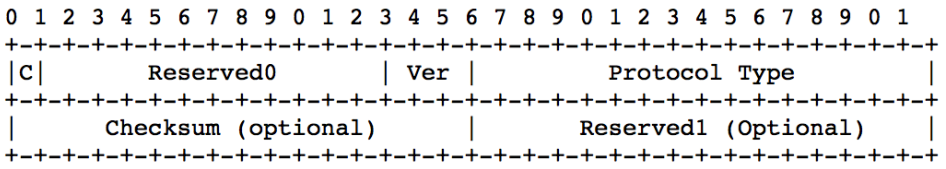
- ¿Y en los P routers? ¿Reenvían en función de esa etiqueta?
- Se crean LSPs entre los PEs (“Tunnel LSPs”)
- Los P routers reenvían en función de esa etiqueta externa
- Un full-mesh entre los PEs que compartan VRF
- Podrían ser otro tipo de túneles (GRE o IP en IP, RFC 4797), lo cual elimina el requerimiento de una red de transporte MPLS

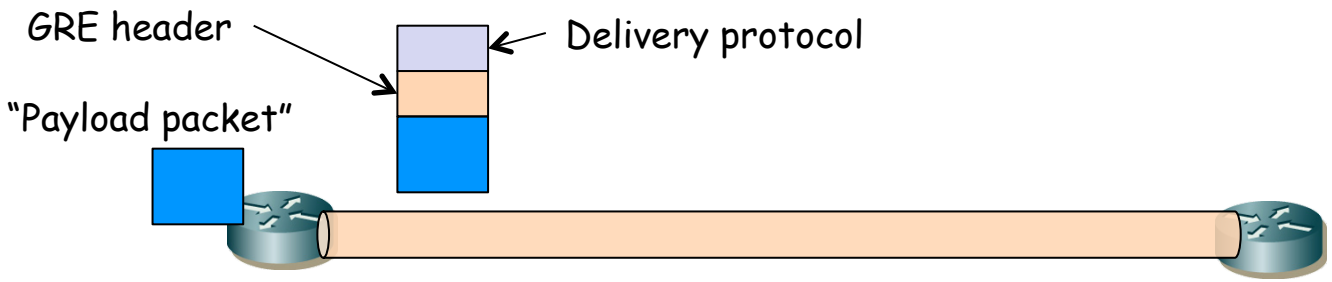


GRE

GRE

- RFC 2784 “Generic Routing Encapsulation (GRE)”
- Encapsular un nivel de red en otro nivel de red
- PPTP (Point-to-Point tunneling Protocol) usa algo similar a GRE
- La cabecera básica GRE ocupa 8 bytes
- Uno de los campos es un Ethertype (*Protocol Type*)
- La versión anterior (RFC 1701) tenía más campos que desaparecen en esta
- Aunque algunos se recuperan en la RFC 2890 “Key and Sequence Number Extensions to GRE” (…)

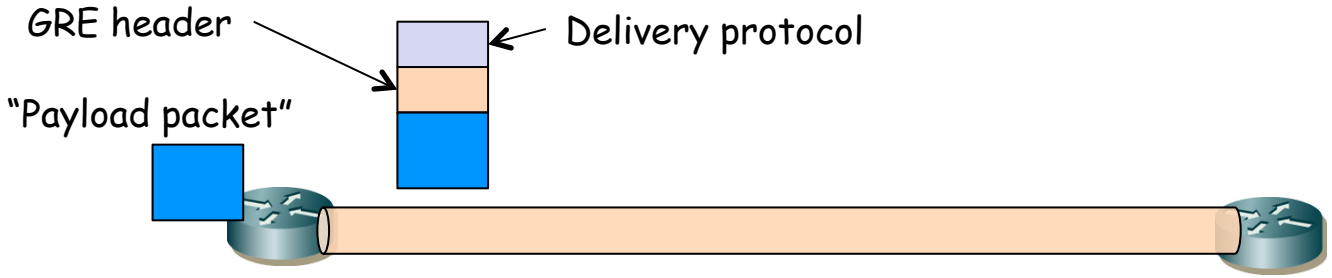




GRE

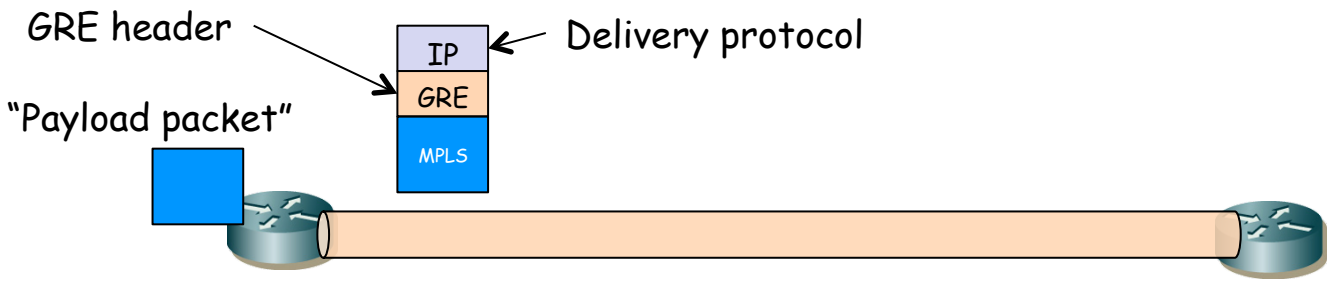
- RFC 2890 “Key and Sequence Number Extensions to GRE”
- “Key” sirve para distinguir flujos dentro del túnel
- “Sequence Number”
  - Si hay “key” entonces el número de secuencia es por “key”
  - Permite dar entrega en orden (aunque no fiable)
  - Si llega uno “anterior” lo descarta
  - Si llega uno que deja un hueco puede guardarlo intentando reconstruir la secuencia
  - Pasado cierto tiempo sin lograr reconstruir los reenvía

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|C| |K|S| Reserved0       | Ver |         Protocol Type         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Checksum (optional)      |       Reserved1 (Optional)    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Key (optional)                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Sequence Number (Optional)                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```



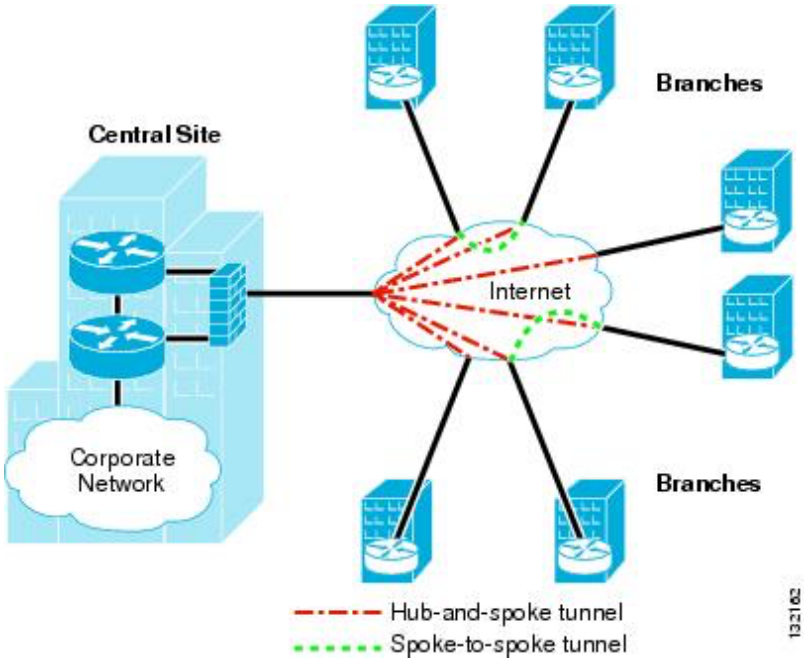
MPLS in GRE in IP

- RFC 4023 “Encapsulating MPLS in IP or Generic Ruting Encapsulation (GRE)”
- El “*delivery protocol*” podría ser IP (protocol = 47 = GRE)
- El “*payload packet*” podría ser MPLS (Ethertype 0x8847 para unicast y ese mismo ó 0x8848 para multicast, RFC 5332)
- EoMPLSoGRE = Ethernet over MPLS over GRE
- Al transportarse sobre IP puede emplear IPSec
- RFC 4023 contempla también que MPLS se transporte directamente sobre IP, lo cual es más eficiente (sin GRE)
- Puede haber motivos para tener GRE (exista el túnel con anterioridad, la implementación del equipo lo requiera en su fastpath, etc)



mGRE y DMVPN

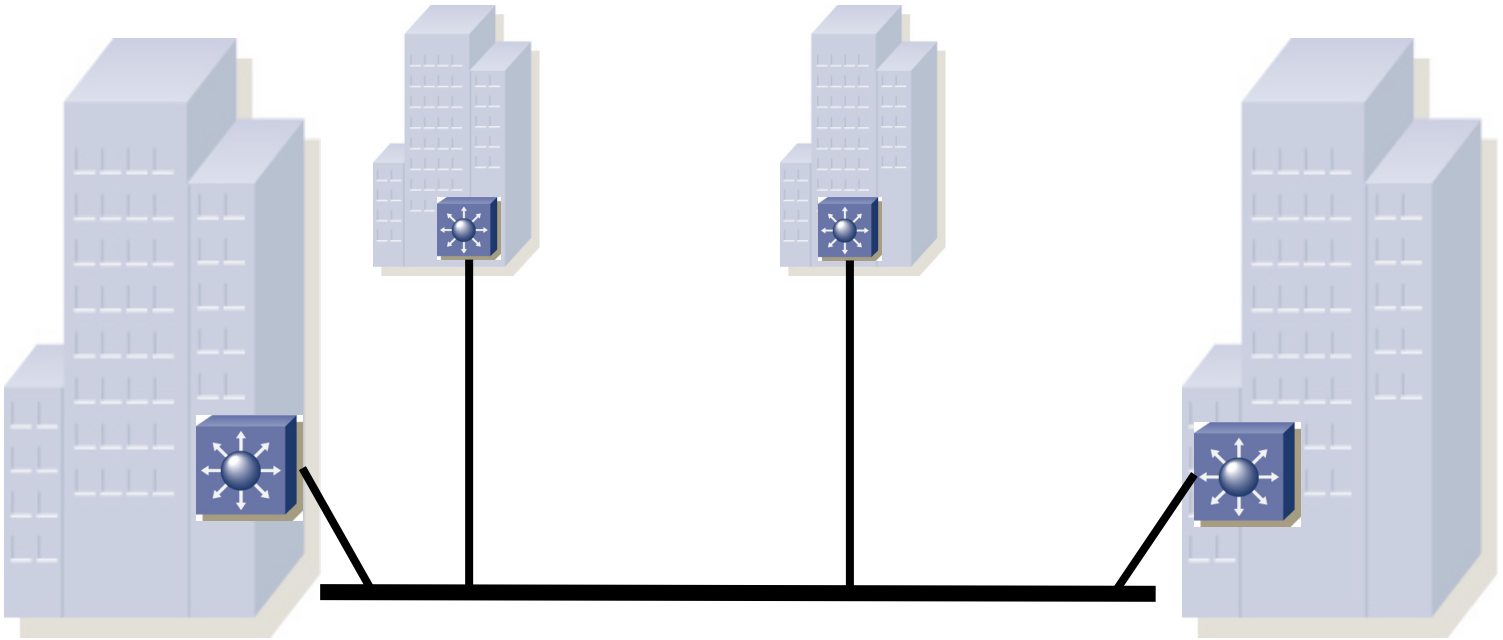
- mGRE = Multipoint GRE
- DMVPN = Dynamic Multipoint VPN
- Solución propietaria de Cisco



VPLS

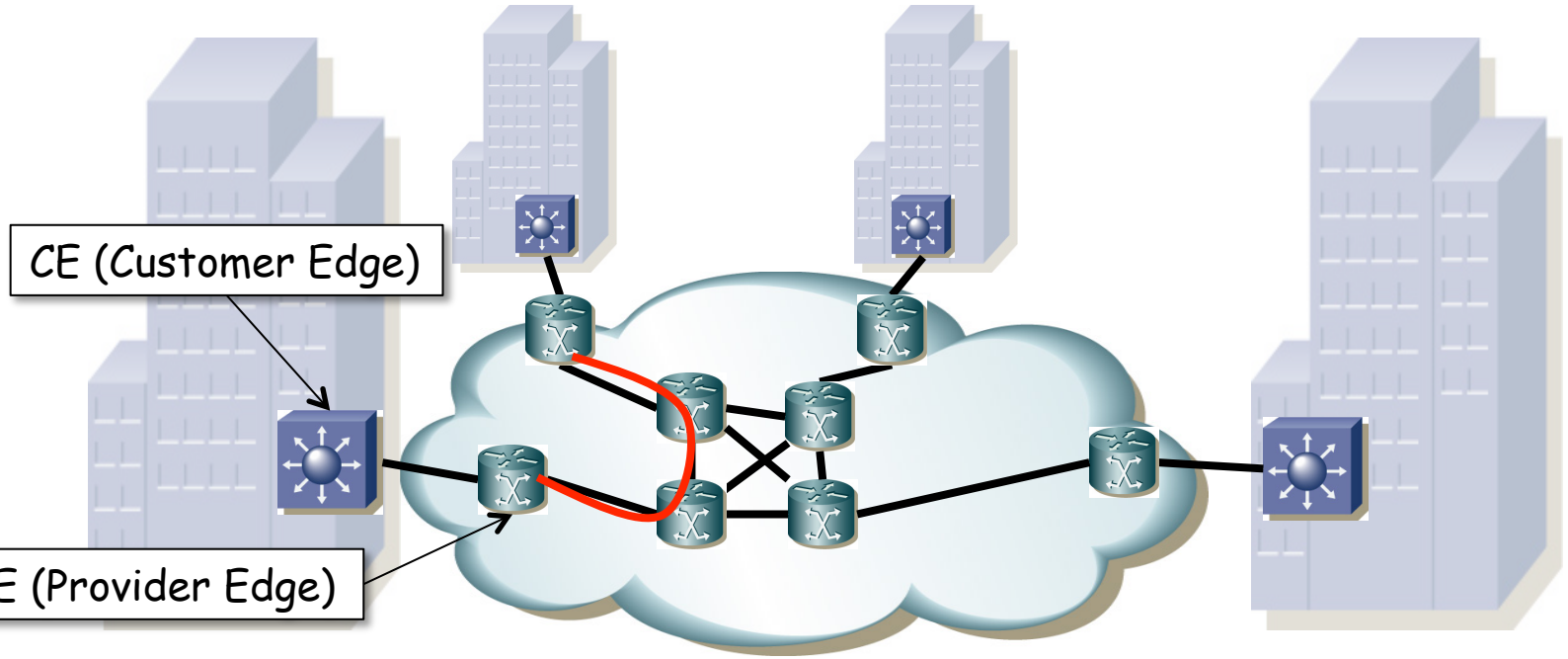
MPLS y VPLS

- *"Virtual Private LAN Service",* una VPN layer 2
- Interconecta múltiples *sites* en un solo dominio puenteado
- Todos los extremos se comportan como si estuvieran en una LAN
- Transporta Ethernet así que sobre ella el cliente puede usar IP o cualquier otro protocolo
- Los equipos de usuario (Customer Edge) pueden ser switches o routers
- Transporte MPLS u otra solución de túneles (GRE, L2TP, IPsec)



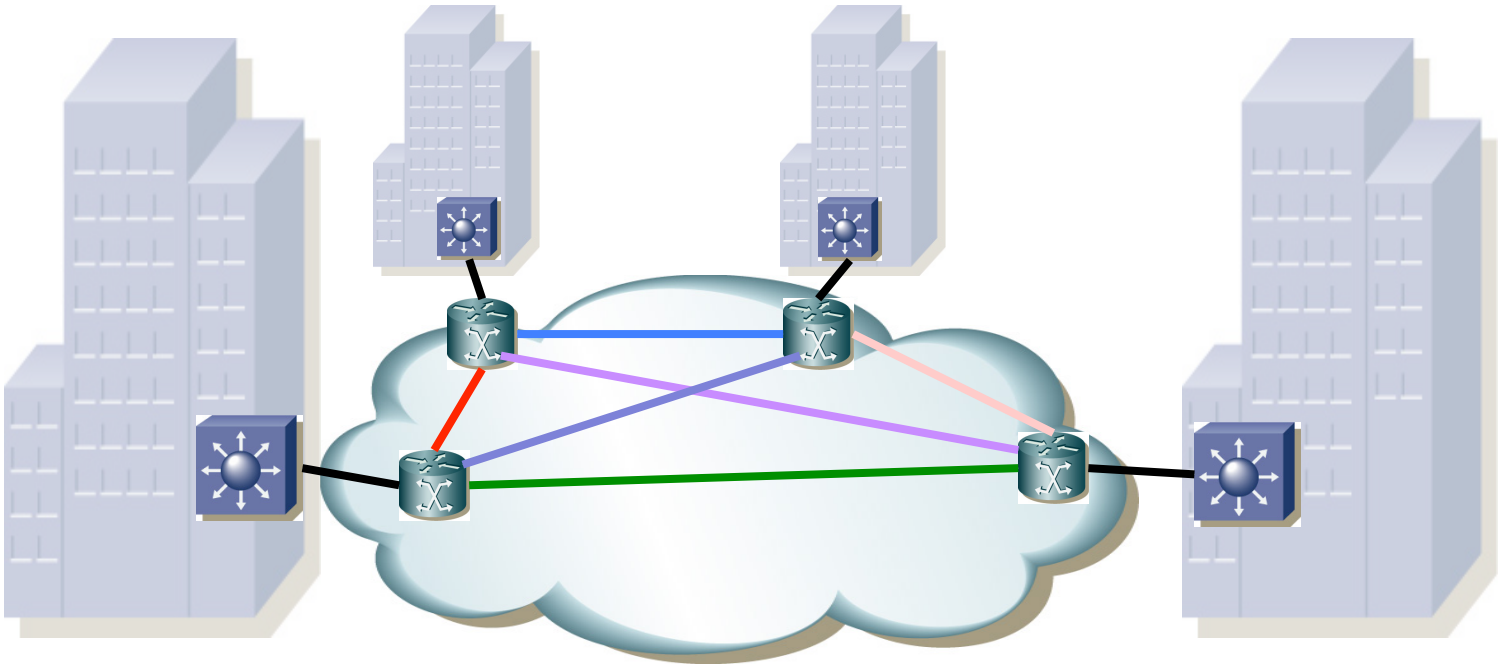
MPLS y VPLS

- El dominio MPLS puede transportar las tramas MPLS sobre IP o sobre otra tecnología
- La red puede dar servicio VPLS a más de un cliente
- El PE hace aprendizaje de direcciones MAC y replicación de tramas de forma independiente para cada cliente
- No interfiere el servicio de un usuario al otro (pueden por ejemplo emplear el mismo direccionamiento IP)
- Los equipos frontera establecen entre ellos los LSPs necesarios para el servicio multiacceso



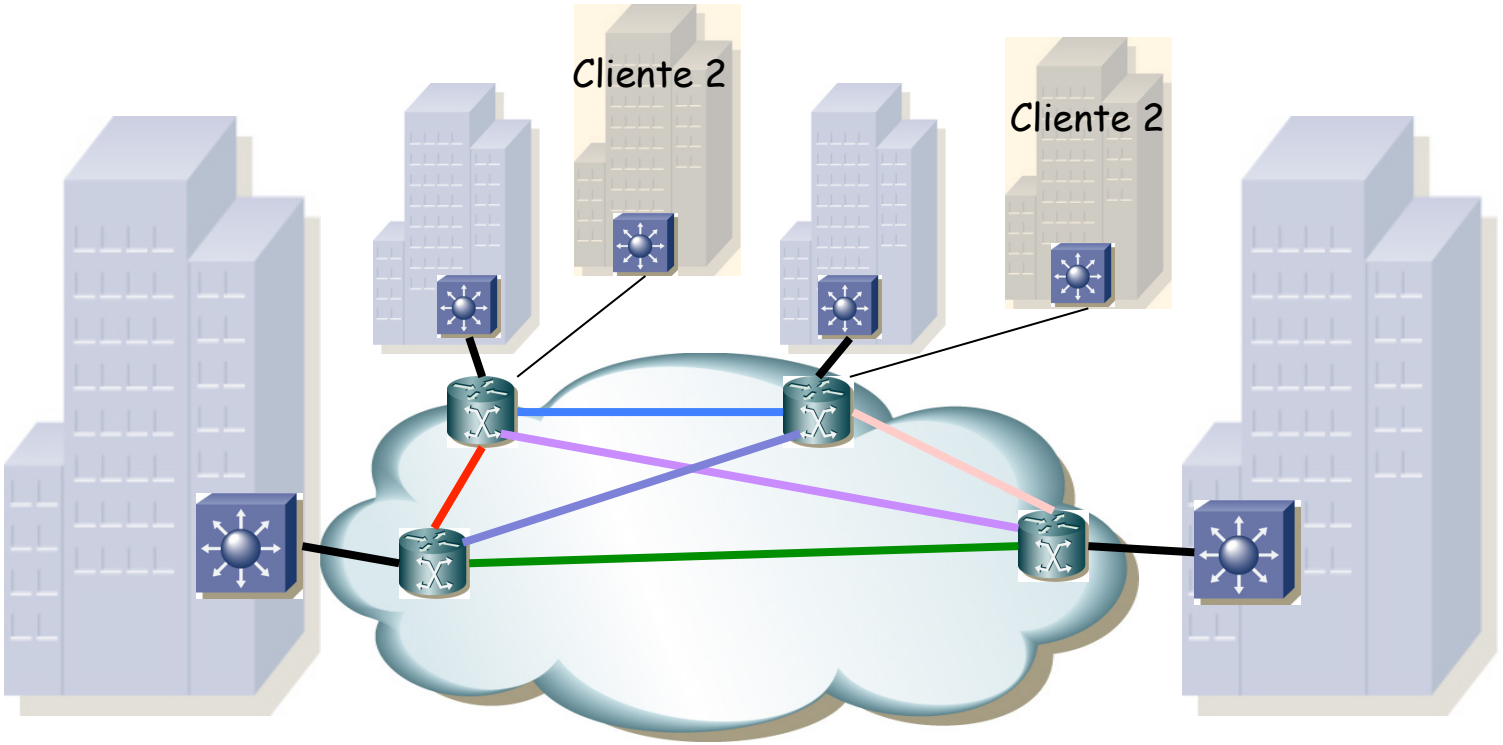
MPLS y VPLS

- Establecen un *full-mesh* entre los nodos frontera
- Para ello disponen de RSVP-TE (o LDP)
- Esos LSPs son globales al servicio VPLS, no particulares para cada cliente
- Es decir, puede haber otras LANs creadas con VPLS, para las sedes de otra empresa, y emplearán los mismos LSPs (…)



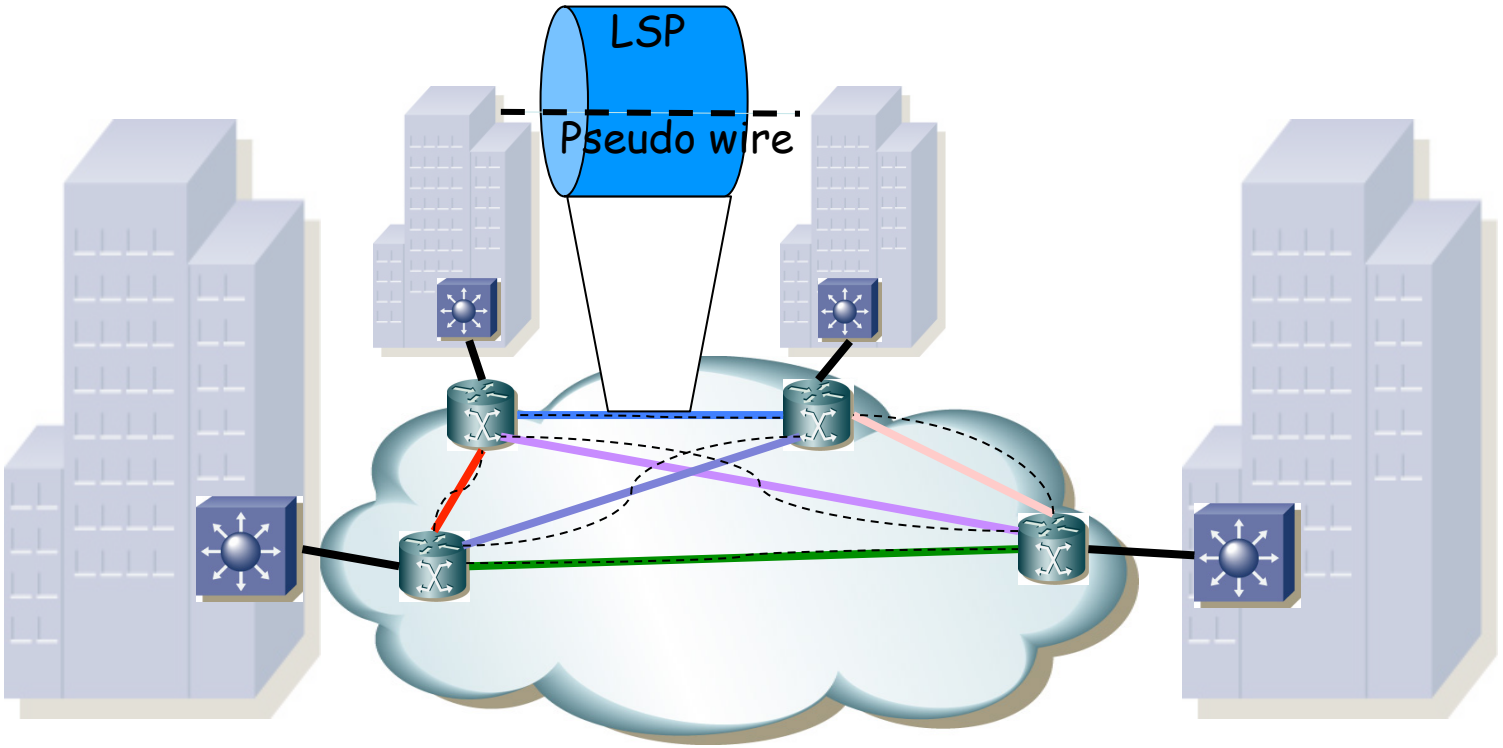
MPLS y VPLS

- Establecen un *full-mesh* entre los nodos frontera
- Para ello disponen de RSVP-TE (o LDP)
- Esos LSPs son globales al servicio VPLS, no particulares para cada cliente
- Es decir, puede haber otras LANs creadas con VPLS, para las sedes de otra empresa, y emplearán los mismos LSPs
- ¿Y para diferenciar a los clientes?



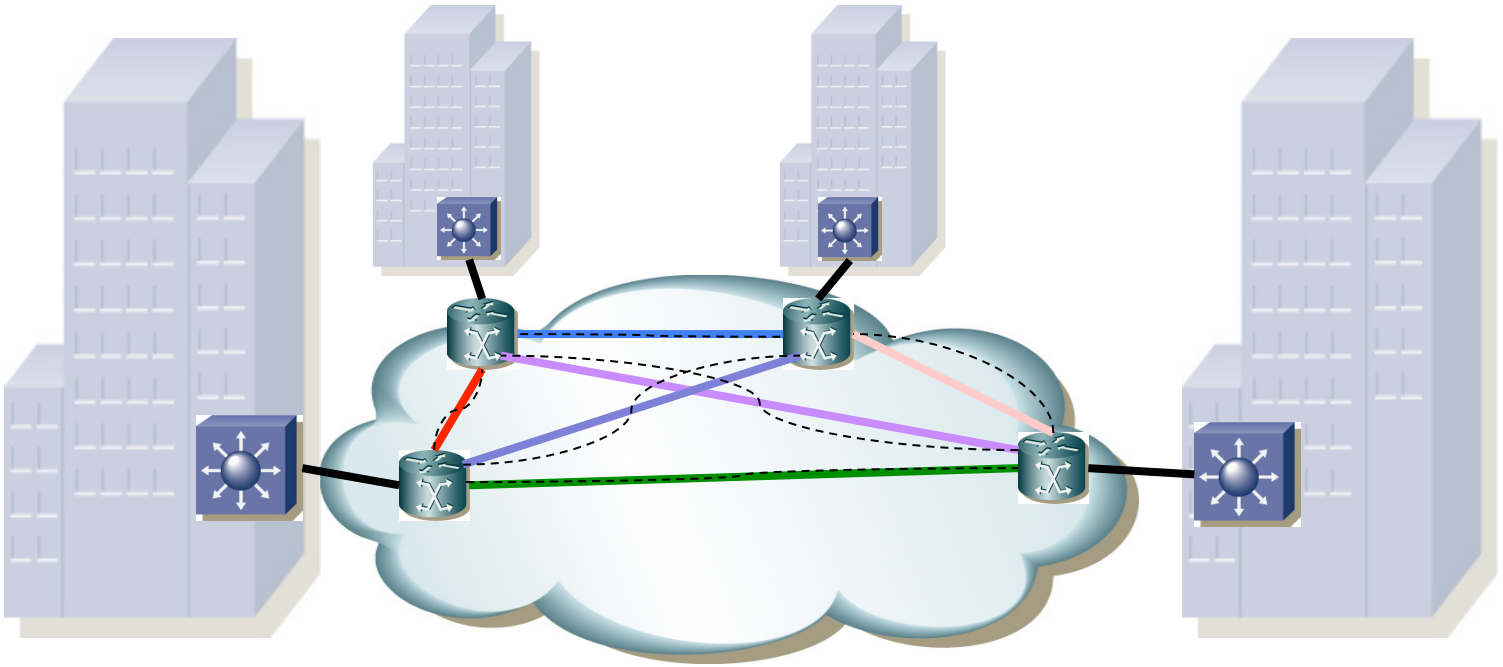
VPLS y PWE

- Por cada instancia VPLS (cada cliente) se establece un full mesh de *pseudo-wires* (PWs) entre los PEs
- RFC 3985 "Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture"
- Un PW emula un circuito, por ejemplo para transportar un E1 o un PVC ATM
- También puede transportar Ethernet, AAL5, SDH, etc



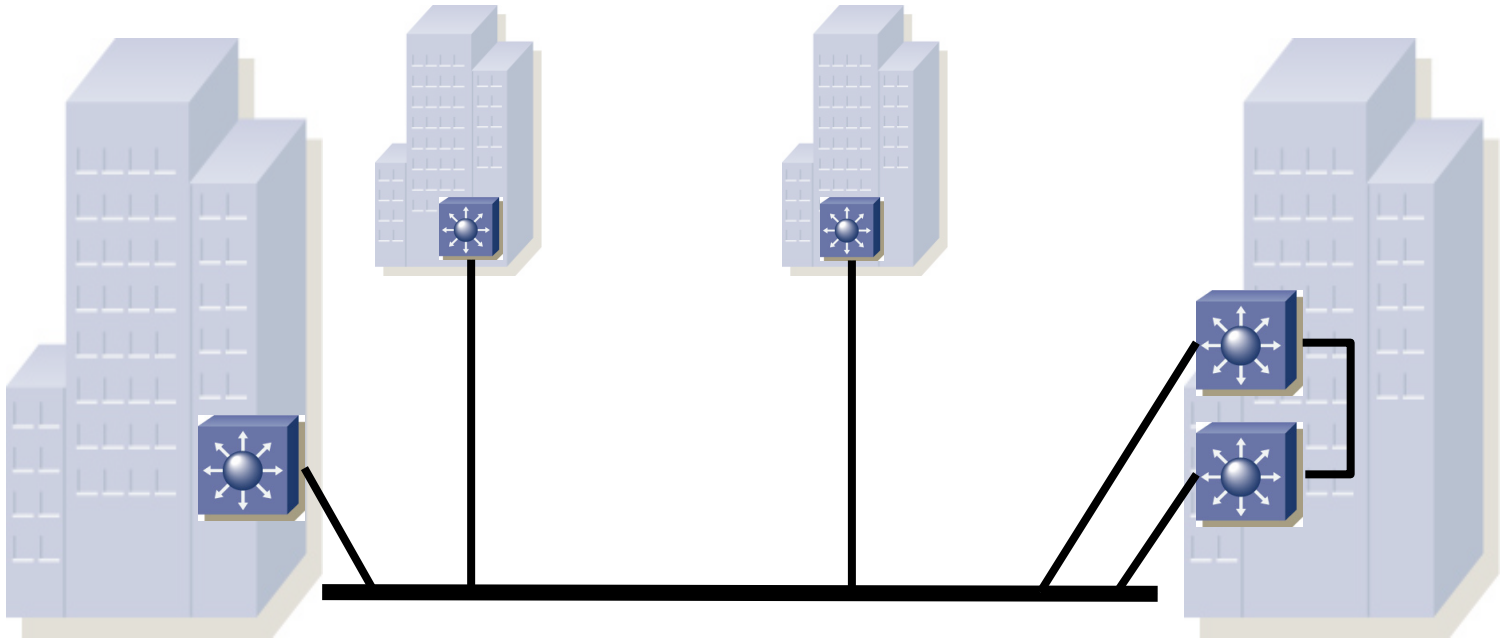
VPLS y PWE

- El full-mesh de PWs hace que los PE puedan enviarse directamente los unos a los otros
- Es decir, no necesiten hacer reenvío
- Así no hace falta resolver posibles bucles
- Simplemente se implementa una solución que se llama de "*split horizon*":
  - Un PE no debe reenviar tráfico de un PW a otro en el mismo mesh VPLS



VPLS y PWE

- Sí puede haber ciclos, pero creados por el usuario para obtener redundancia
- En ese caso podrá emplear STP
- Las BPDUs se transportarían normalmente por el mesh VPLS



VPLS Control Plane

- Dos alternativas para el establecimiento de los pseudo-wires:
  - RFC 4761 “Virtual Private LAN Service (VPLS) Using BGP or Auto-Discovery and Signaling”
  - RFC 4762 “Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling”

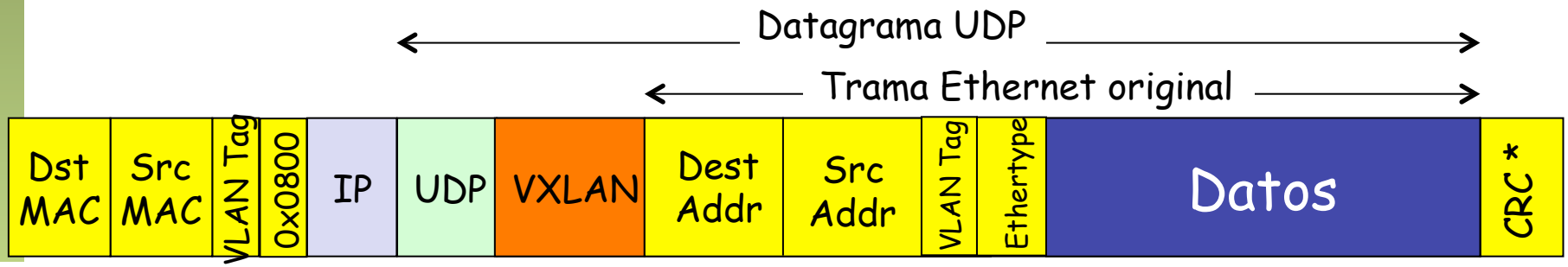
Problemas en VPLS

- Se deben establecer Nx(N-1)/2 pseudo-wires
- Problema de escalabilidad (cantidad de tráfico de control)
- Replicación de paquetes que sufren inundación:
  - Se lleva a cabo en el PE de entrada
  - Se dirigen punto-a-punto a cada otro PE del servicio
  - Mayor trabajo en el PE
  - Más uso de capacidad
  - Mayor retardo (si hay que enviar N veces la trama por N PWs que se implementan sobre el mismo LSP irán en serie)
- Si se añade un acceso del cliente, a un PE diferente, se deben crear los PWs, lo cual implica reconfigurar los demás PEs
- Para despliegues pequeños
- Mejoras:
  - H-VPLS (Hierarchical VPLS)
  - Hierarchical BGP VPLS

VXLAN y NVGRE

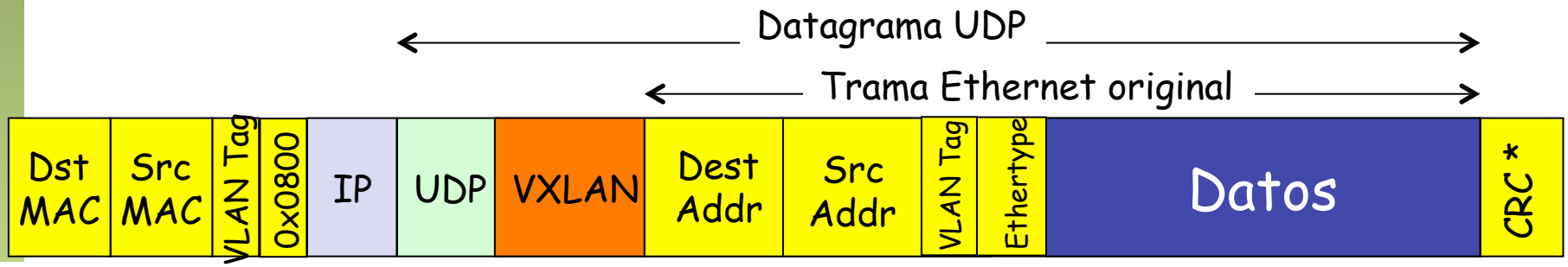
VXLAN

- RFC 7348 “Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks”
- 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
- 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



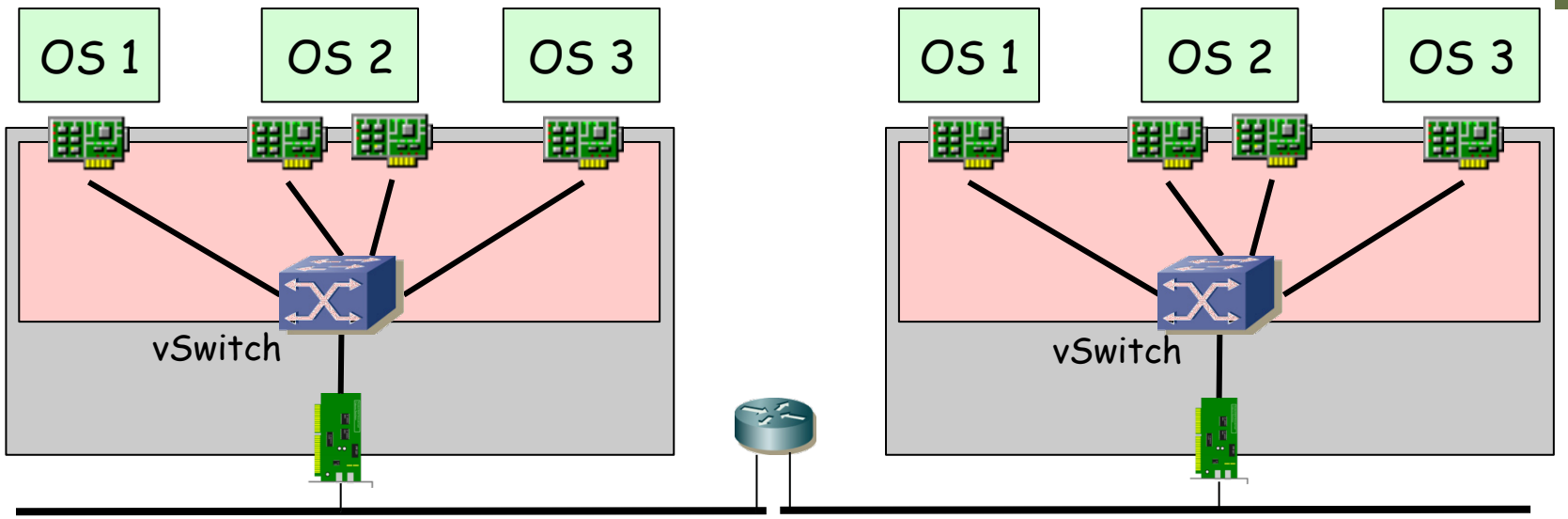
VXLAN

- 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



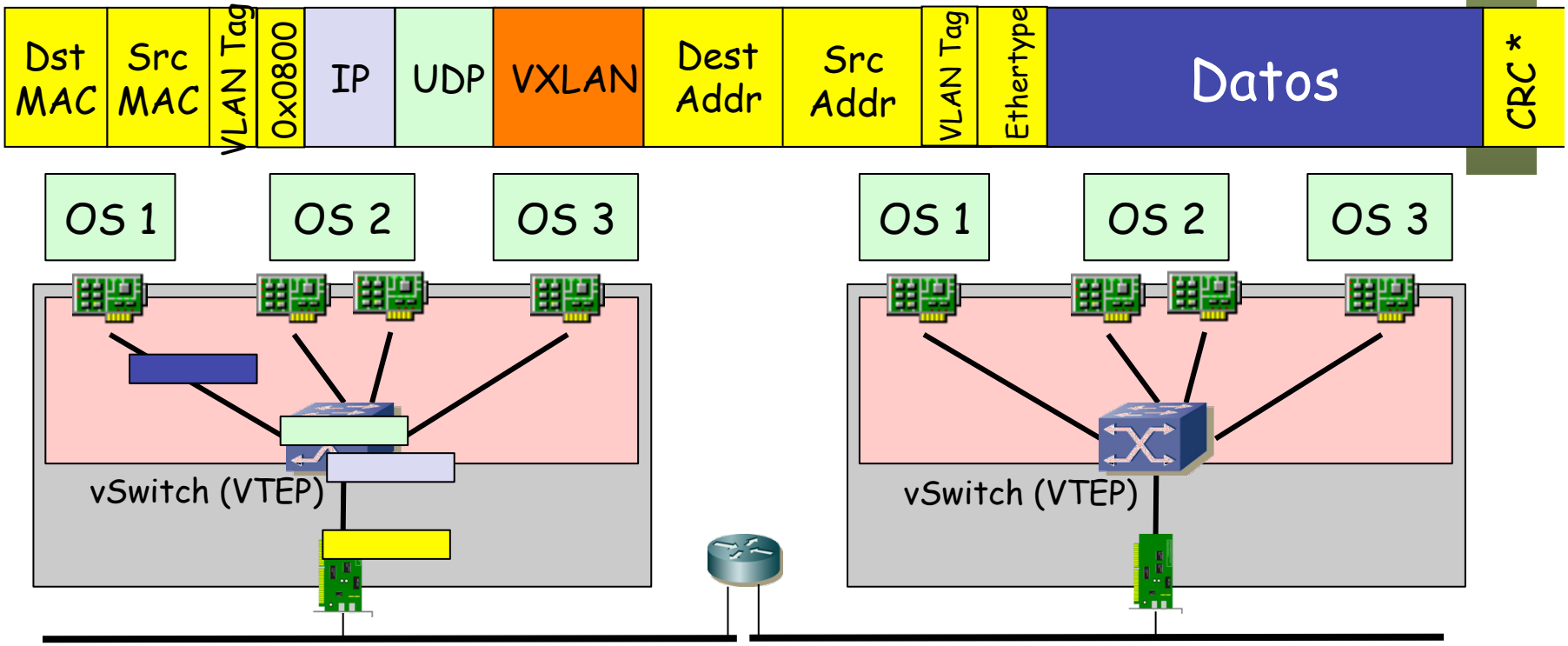
VXLAN: *Data plane*

- 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



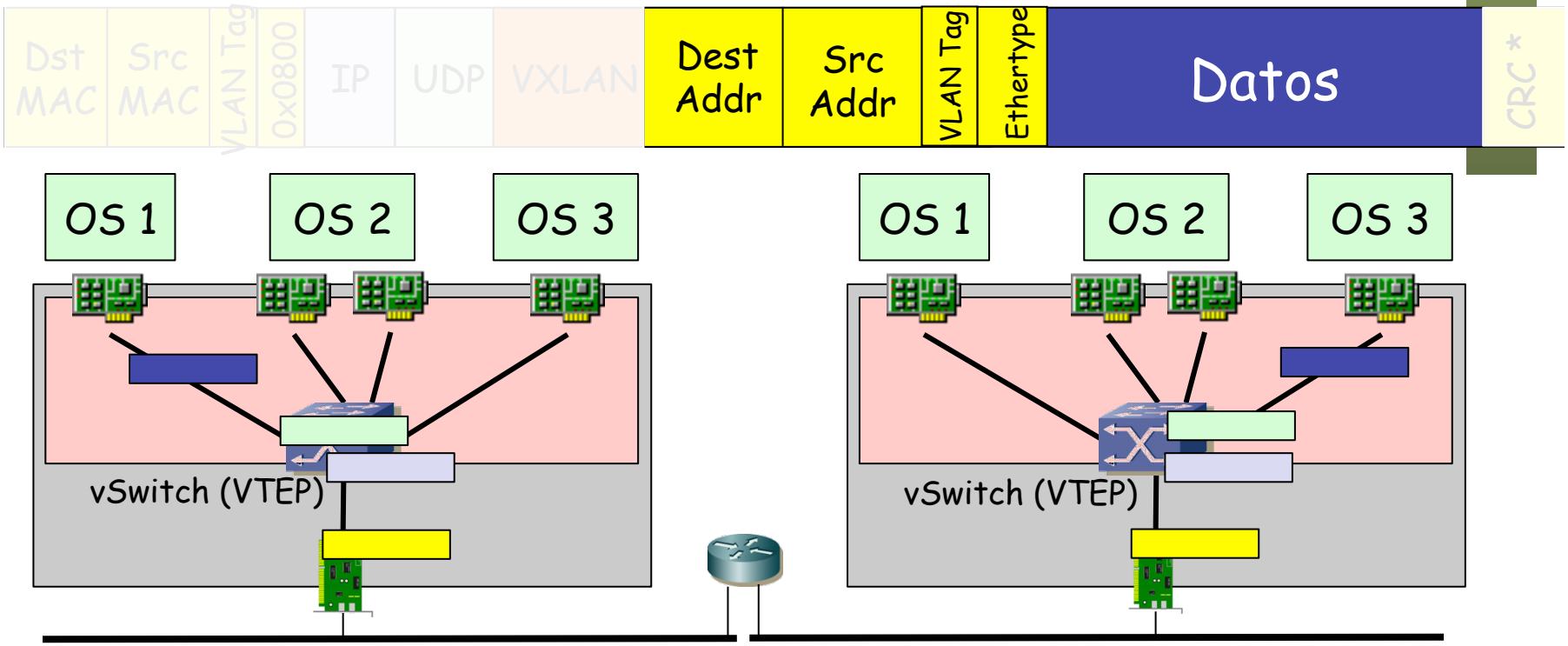
VXLAN: *Data plane*

- 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



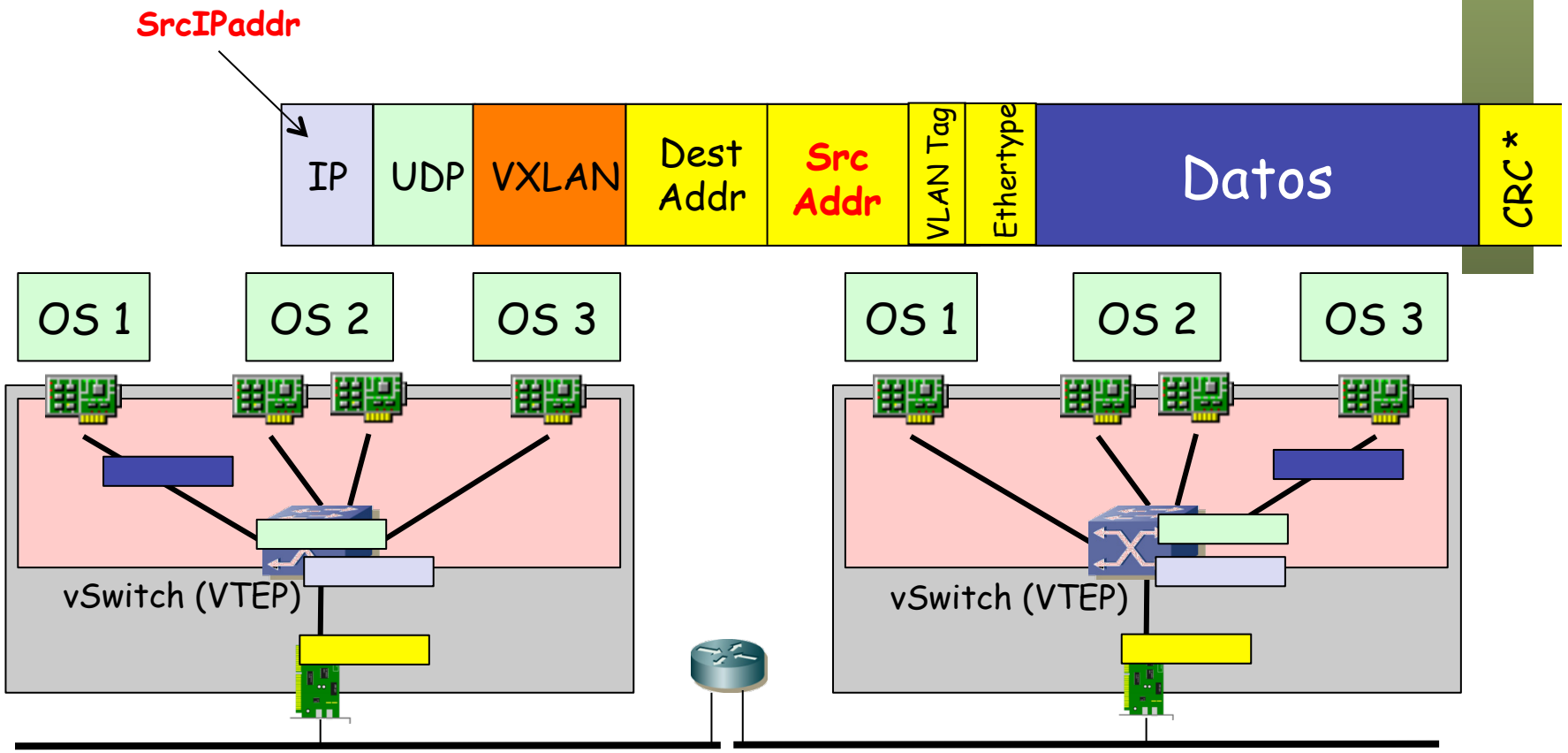
VXLAN: *Data plane*

- 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



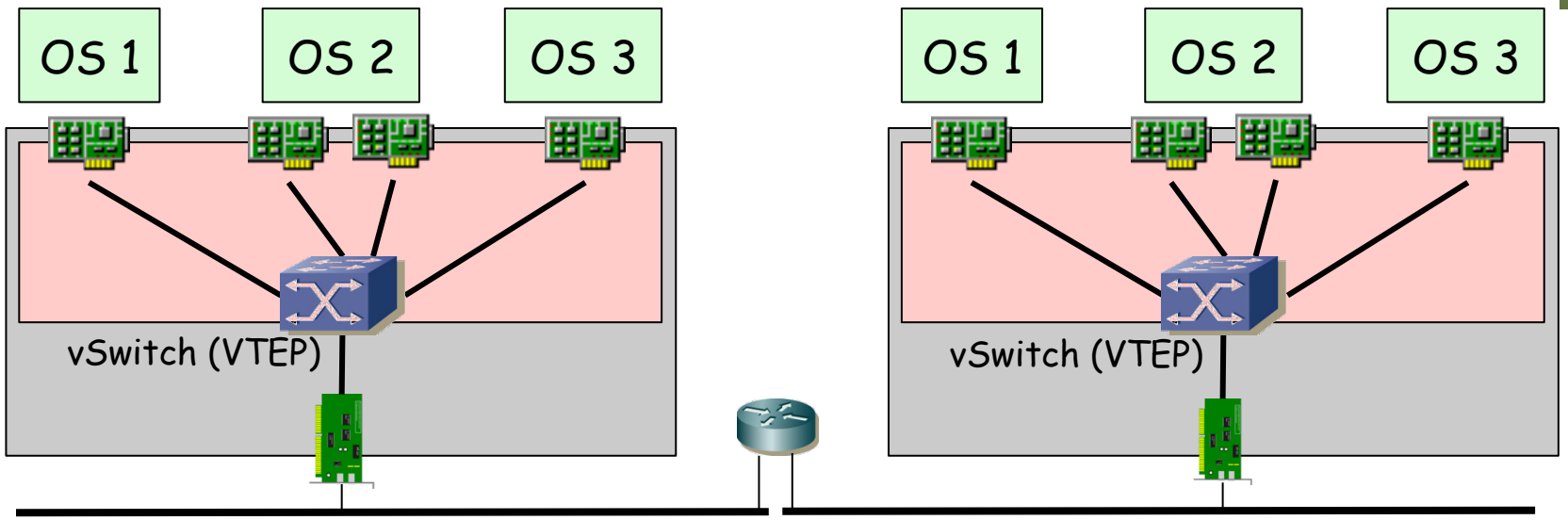
VXLAN: *Control plane*

- 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



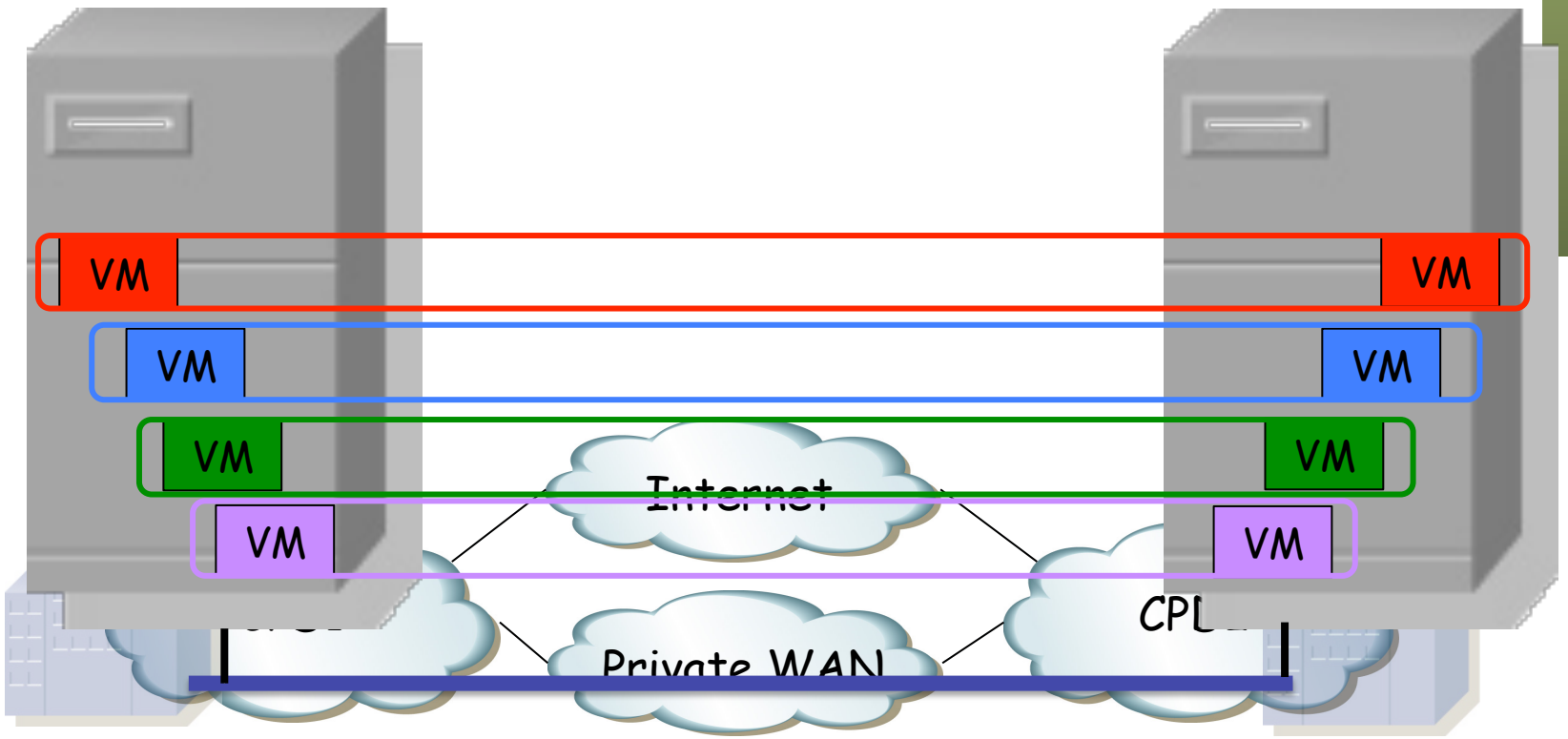
VXLAN: *Control plane*

- ¿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



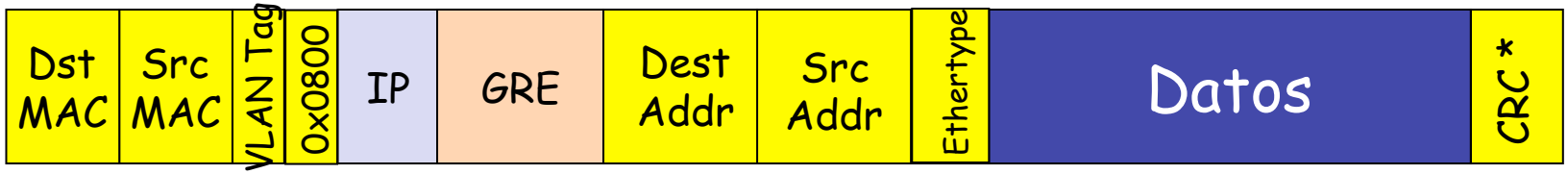
VXLAN

- Los hosts pueden estar en el mismo CPD o en diferente
- Los hosts pueden ser máquinas virtuales
- El transporte entre esas VMs es de las tramas Ethernet
- Eso quiere decir que se comportan como si estuvieran en la misma VLAN
- ¿O en varias VLANs? A fin de cuentas transporta el C-Tag
- La RFC no lo deja claro y parece más inclinada a retirar esa etiqueta (sección 6.1)



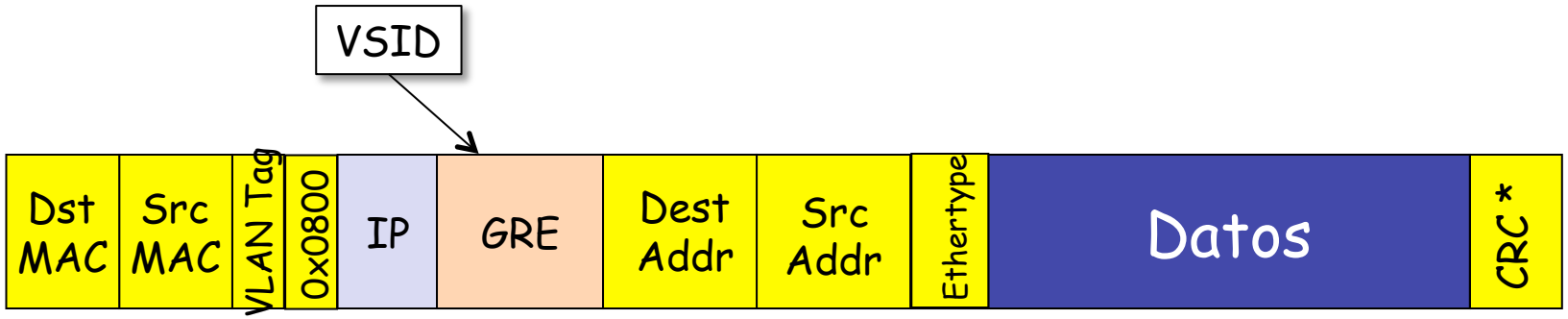
NVGRE

- *Network Virtualization using GRE*
- Crea una topología capa 2 virtual sobre una red capa 3
- RFC draft, última versión es de noviembre de 2014: http://ietfreport.isoc.org/idref/draft-sridharan-virtualization-nvgre/
- 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

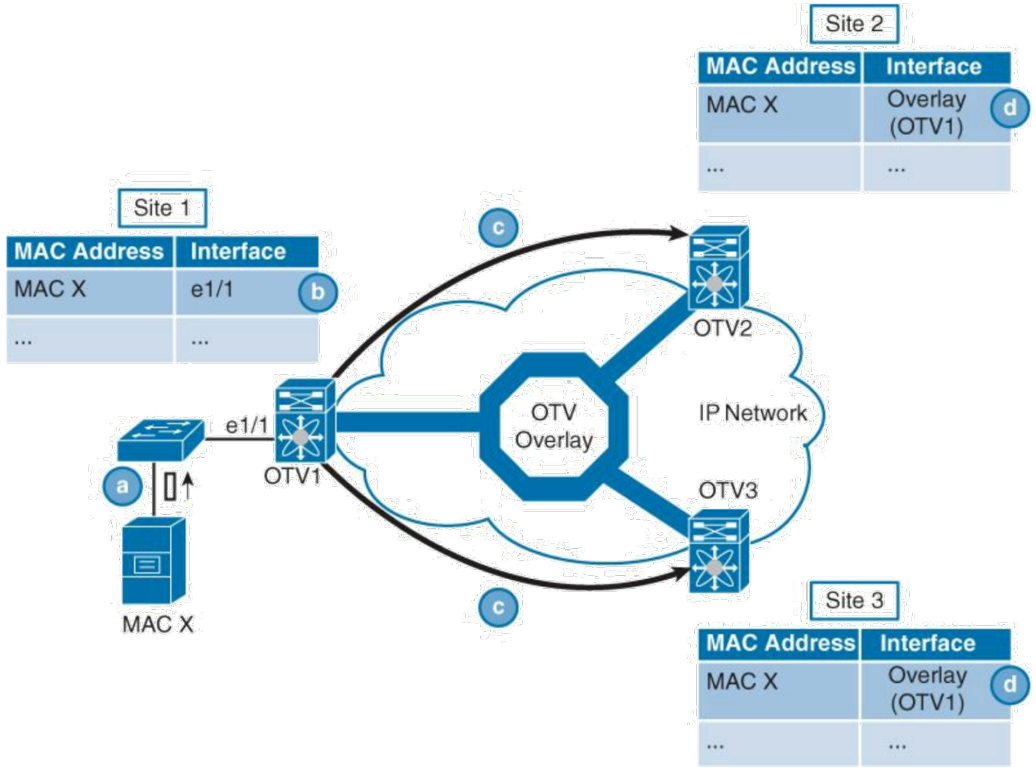
- La RFC no detalla cómo el Endpoint conoce la dirección del destino al que mandar el paquete IP
- Broadcast y multicast
  - Se puede emplean encaminamiento multicast IP con una o más direcciones multicast por VSID
  - Se puede implementar con N-way unicast
- Lo soporta Hyper-V (draft propuesto por Microsoft)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

Microsoft

OTV

OTV

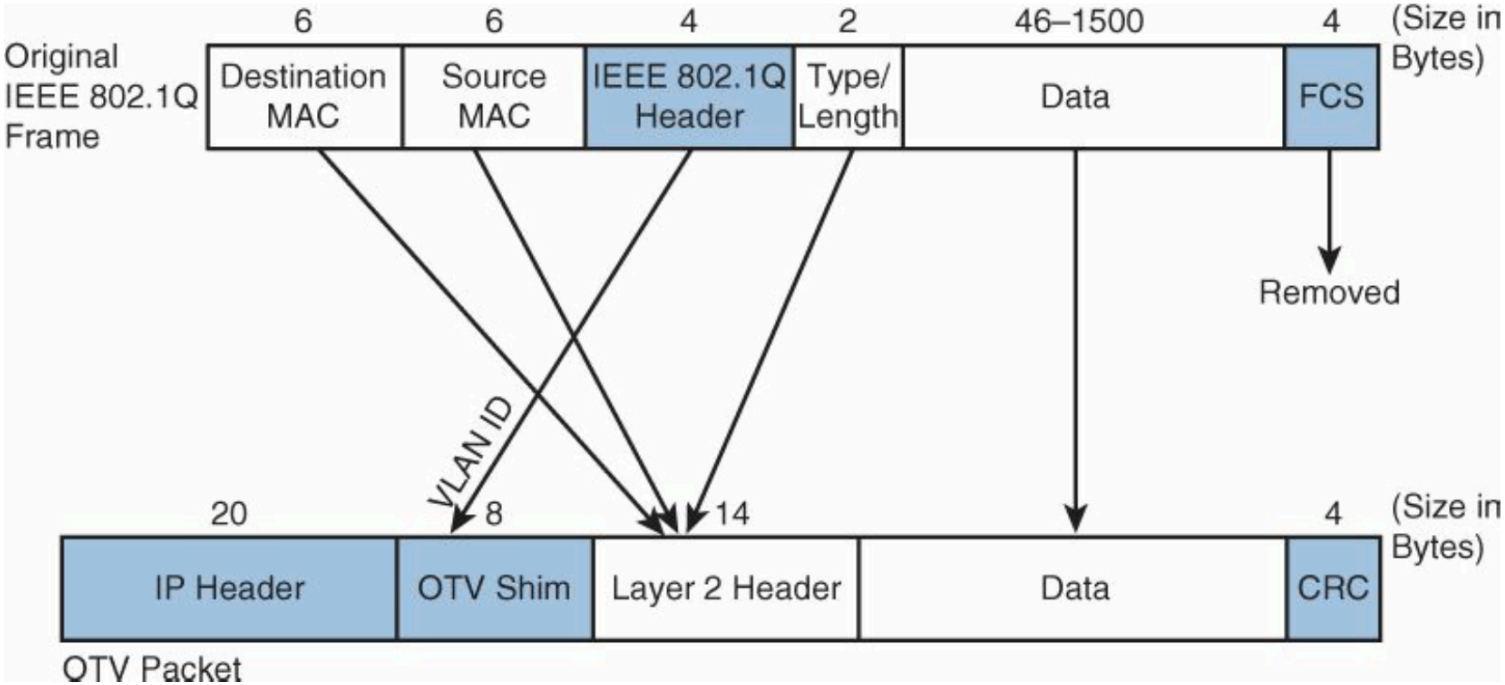
- *Overlay Transport Virtualization*
- Solución propietaria de Cisco
- Conectividad Ethernet a través de una red IP
- No hace aprendizaje de direcciones MAC en el plano de datos
- Emplea IS-IS para intercambiar esa información



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*

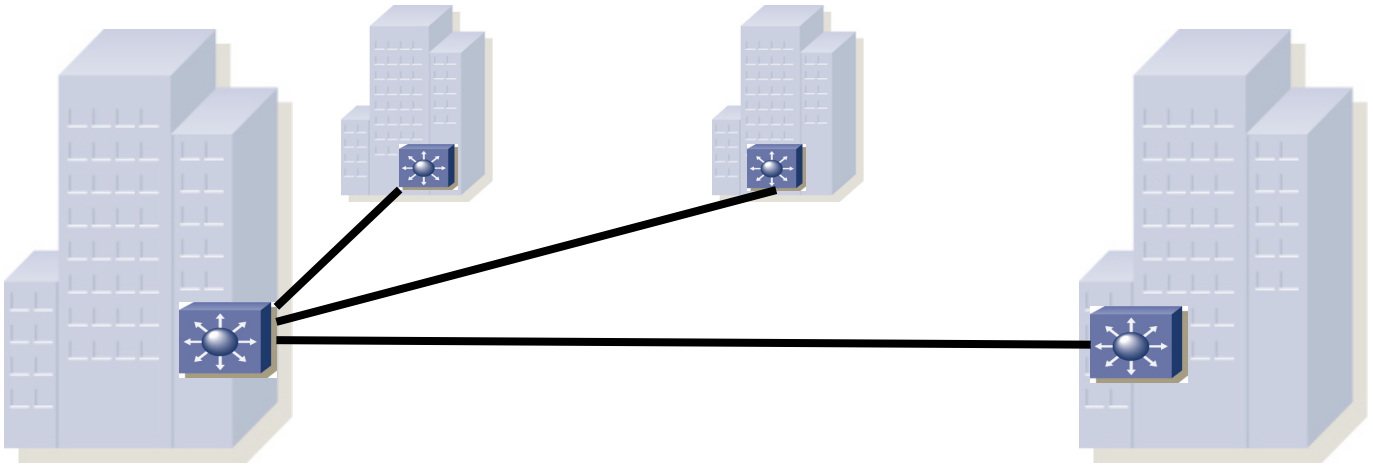
DNS y stickiness

- 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

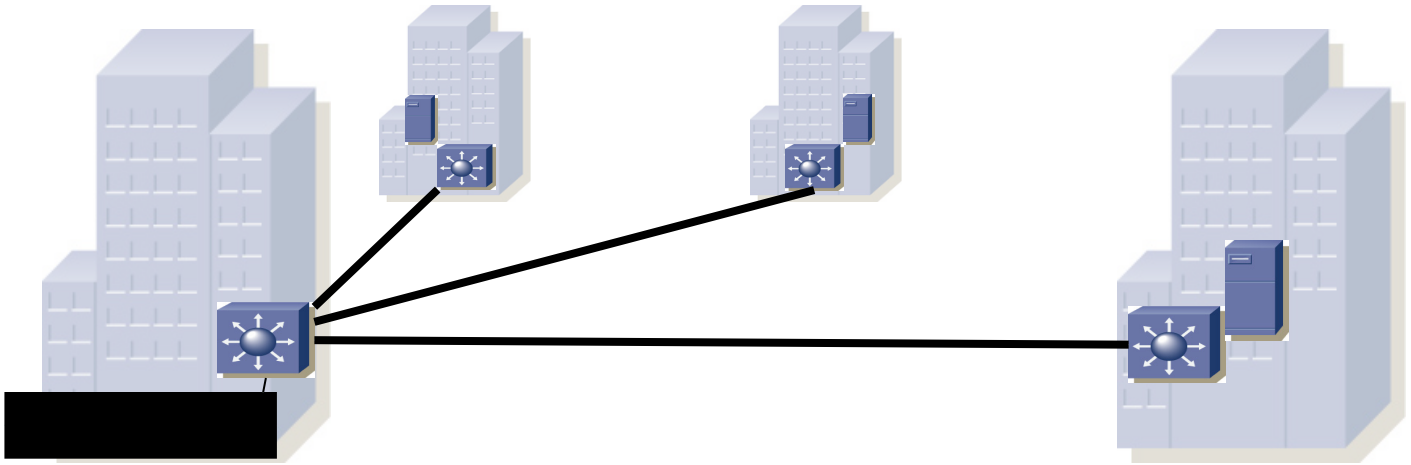
Rendimiento en la WAN

- 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



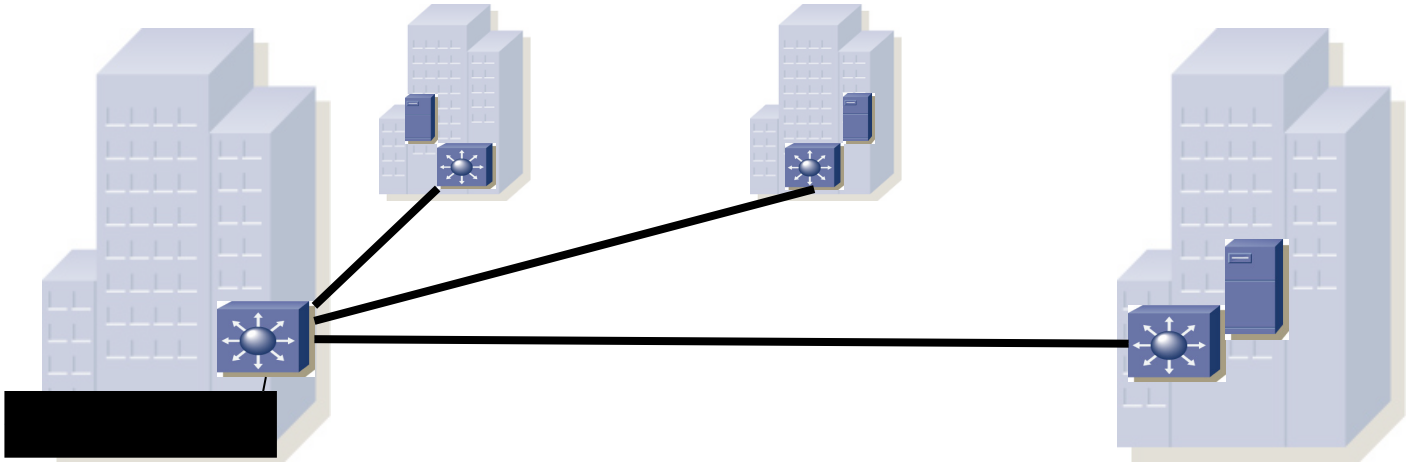
WAN vs LAN Bandwidth

- 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



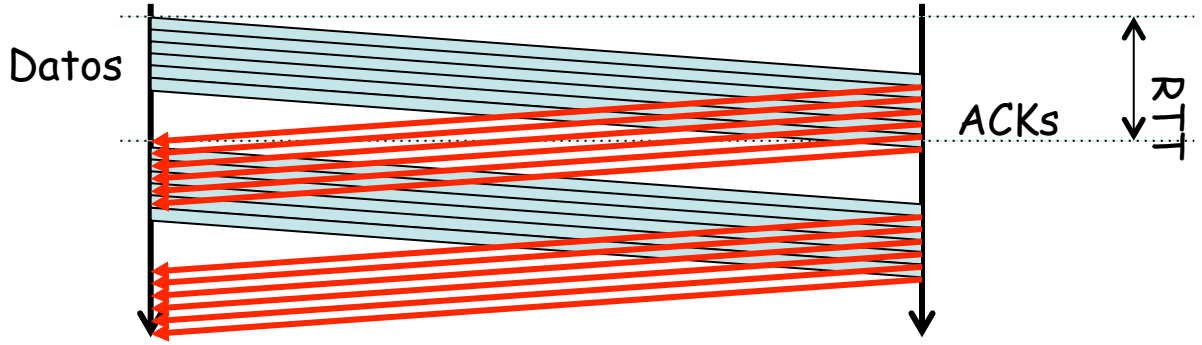
WAN vs LAN Bandwidth

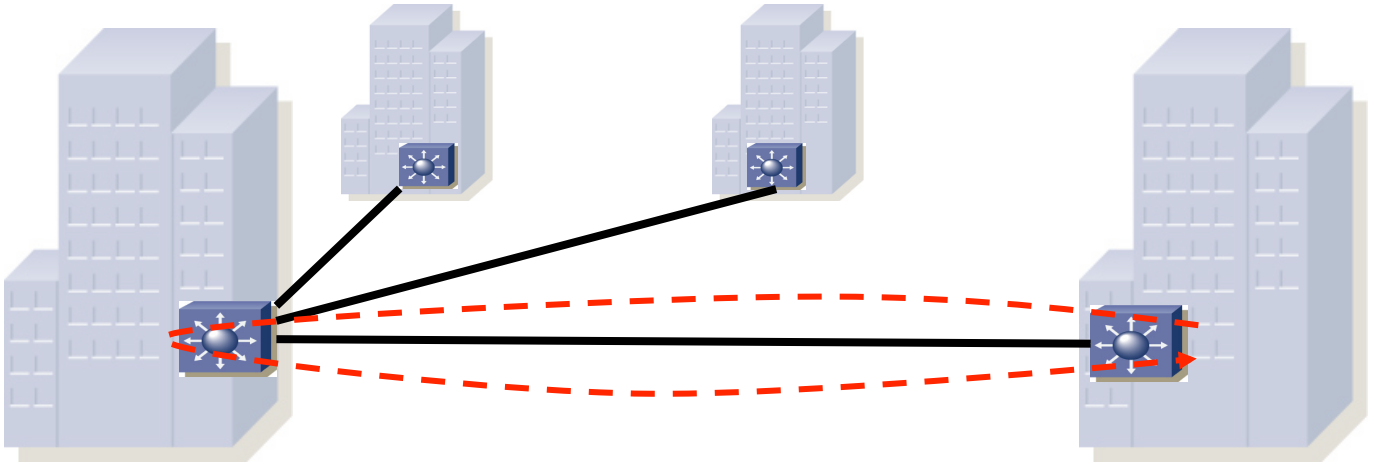
- Hoy en día accesos de baja o alta velocidad según la localización o las necesidades
- DSL, cable modem, fibra, metro Ethernet, UMTS, satélite, etc
- Por el enlace WAN: acceso a ficheros, backups, e-mail, acceso a Internet, streaming, impresión, aplicaciones de gestión, aplicaciones empresariales, autentificación, 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)

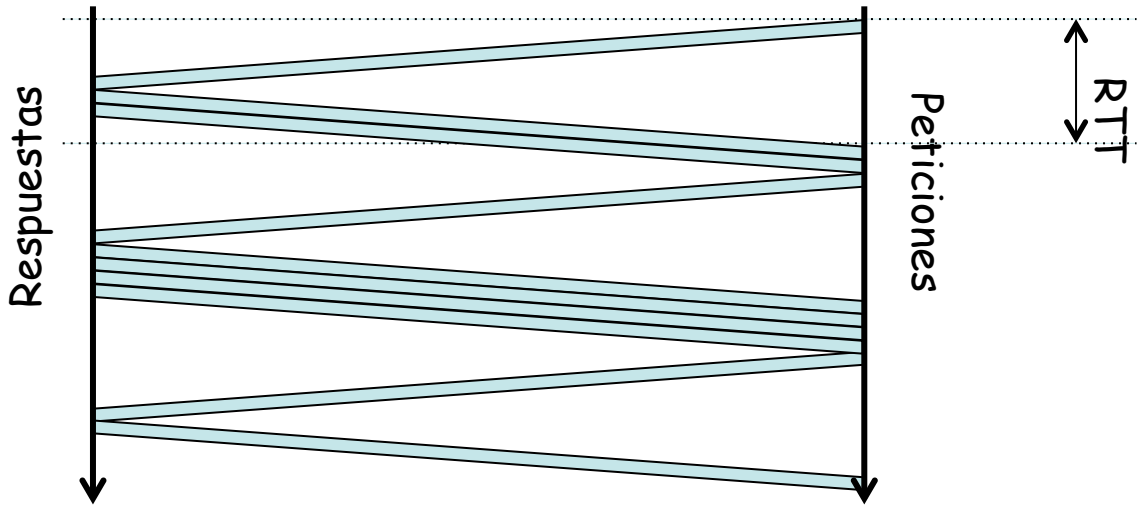
- El RTT tiene un impacto en la velocidad de transferencia y en la interactividad
- En protocolos de ventana deslizante estamos limitados por el tamaño de la ventana entre el RTT
- Esto no es solo TCP sino también protocolos de nivel de aplicación (por ejemplo SMB)
- Añadir capacidad a los enlaces no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

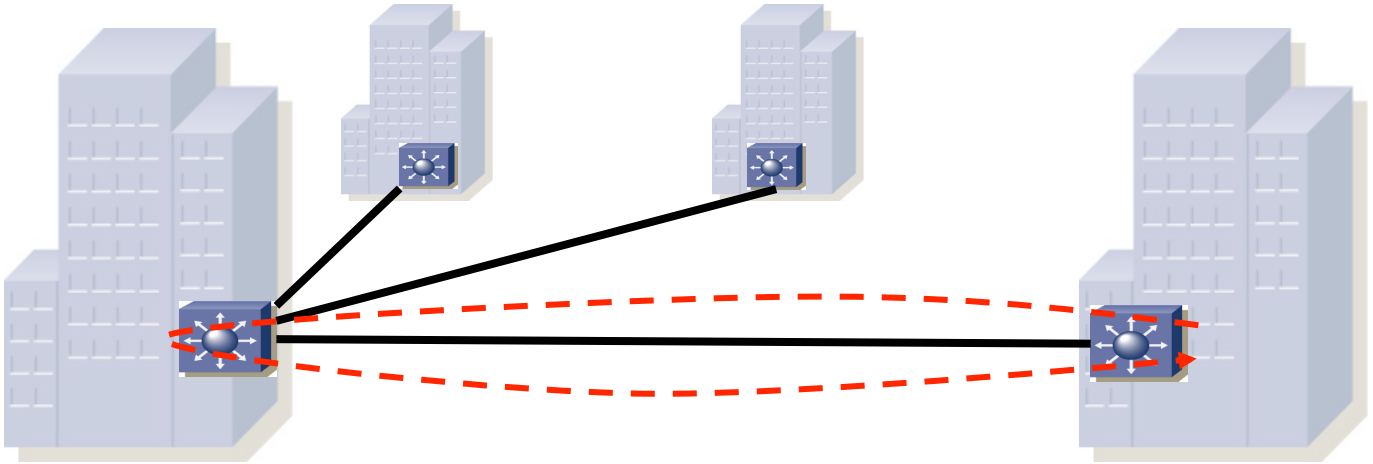




Latencia (retardo)

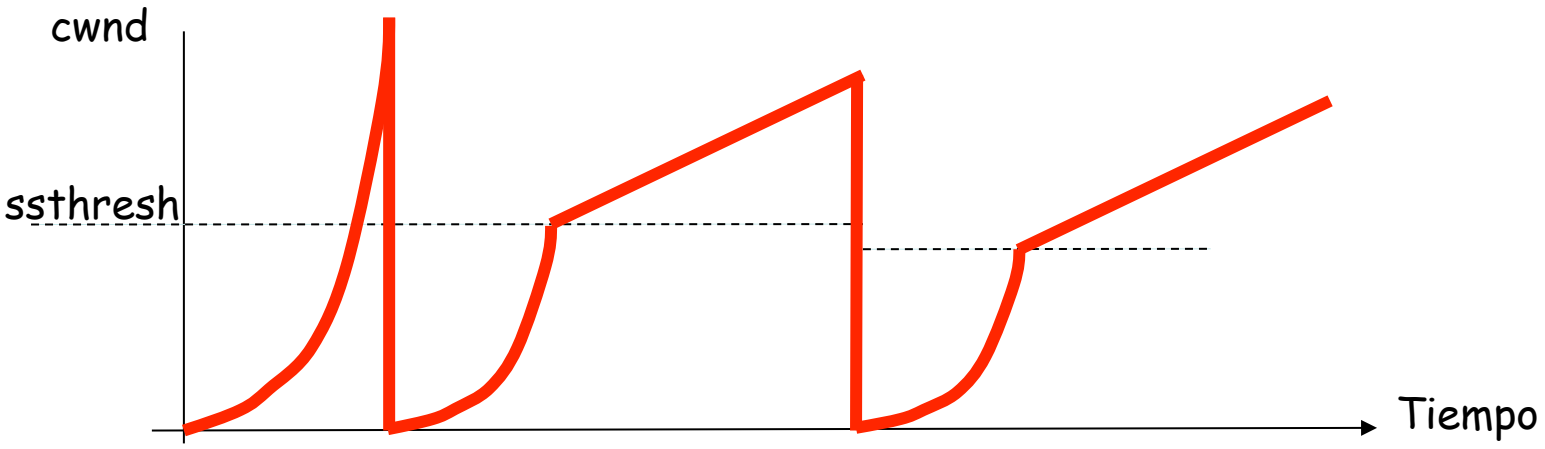
- Aplicaciones muy *chatty* consumen muchos RTTs con pocos datos
- No están limitadas por la ventana del protocolo sino porque necesita la respuesta a una petición antes de poder hacer la siguiente





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)

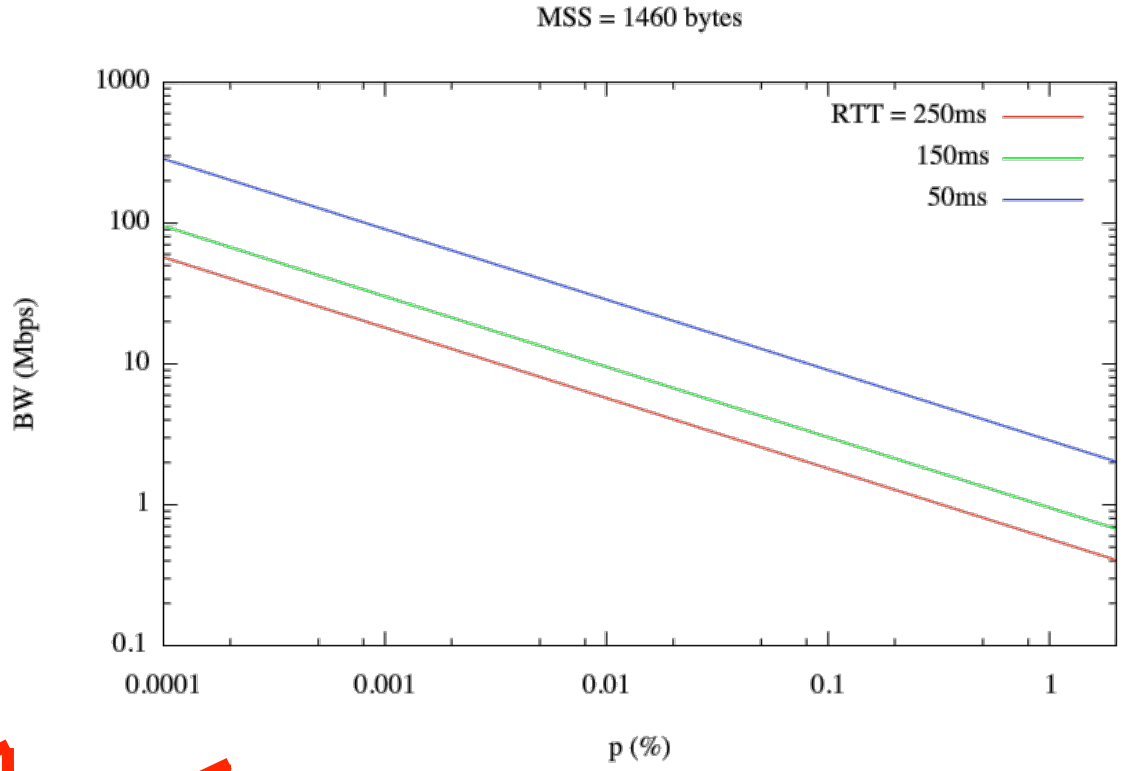


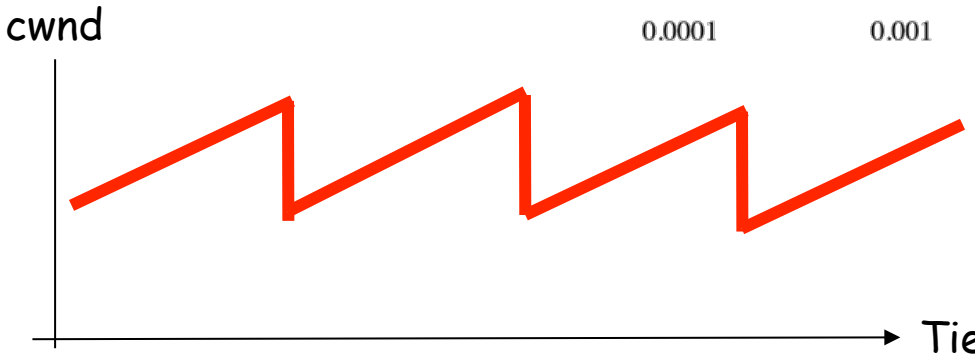
Pérdidas (TCP)

- 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)





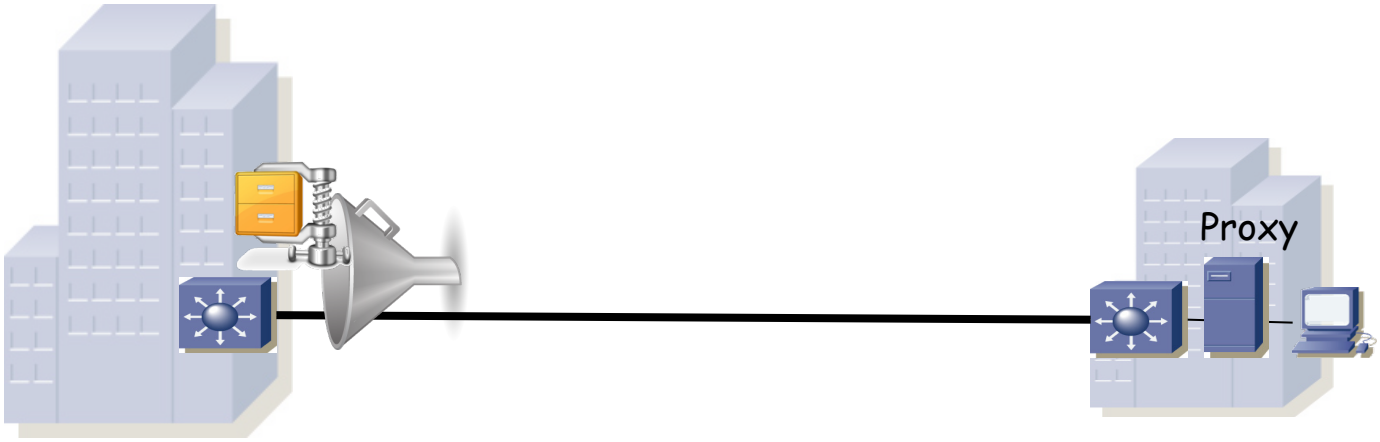
TCP Reno en
congestion avoidance

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

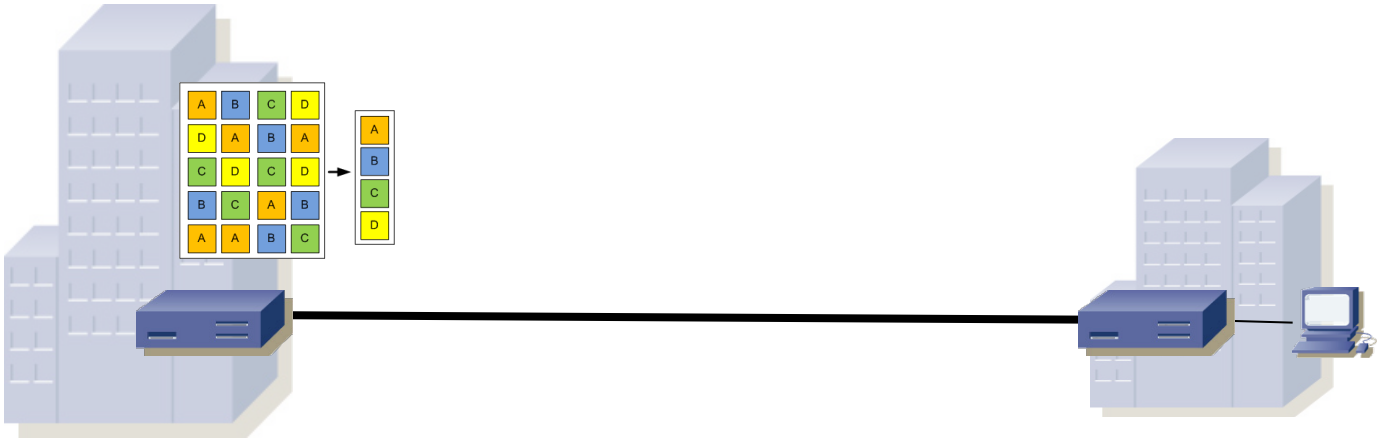
Compresión

- 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



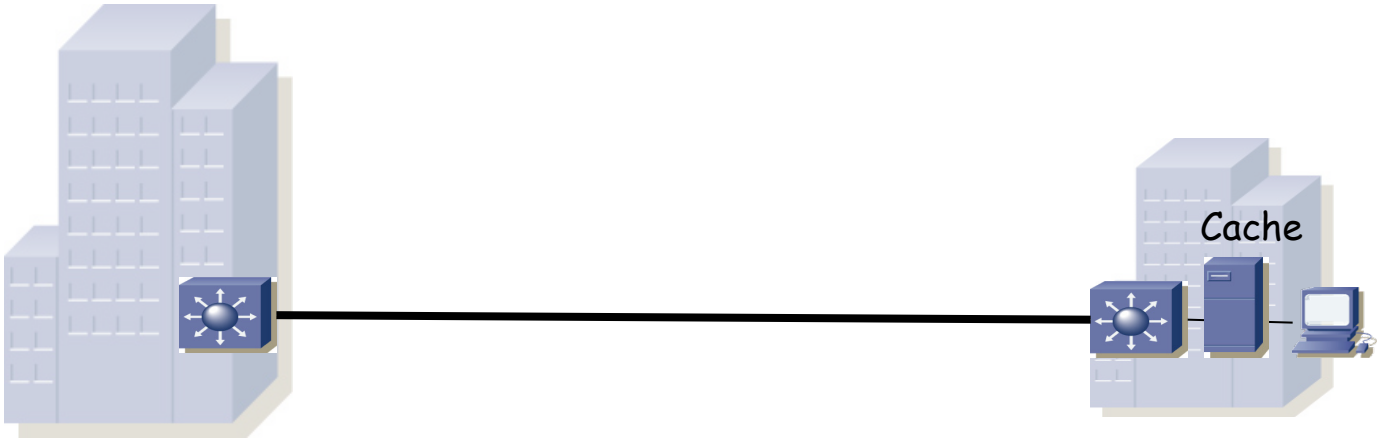
Data deduplicación

- 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 deduplicado



Latencia: Caches

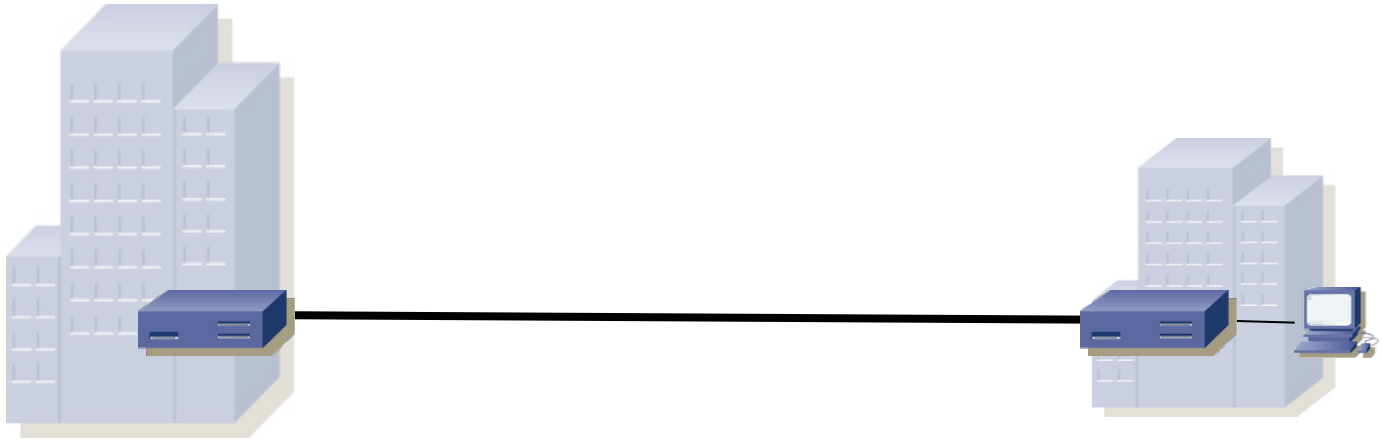
- Sencillas para protocolos como HTTP que están diseñados con ellas en mente
- Factibles pero más complejas para protocolos de compartición de ficheros (por ejemplo SMB)
- Caches para diferentes aplicaciones pueden acabar manteniendo los mismos objetos, ejemplo:
  - Usuario descarga un fichero de un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