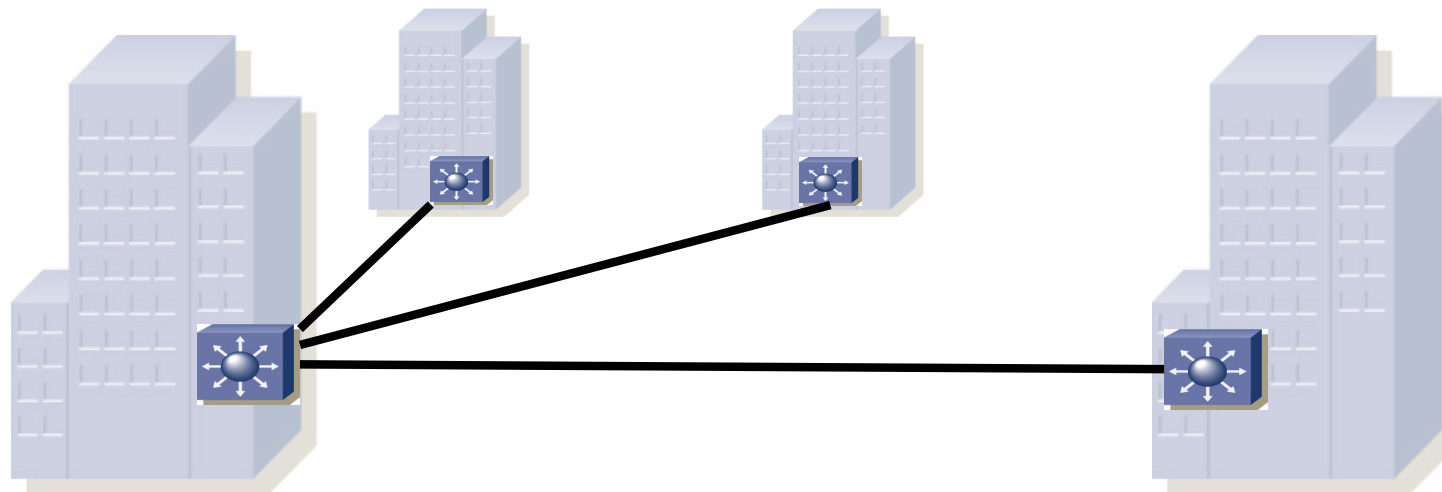


WAN optimization

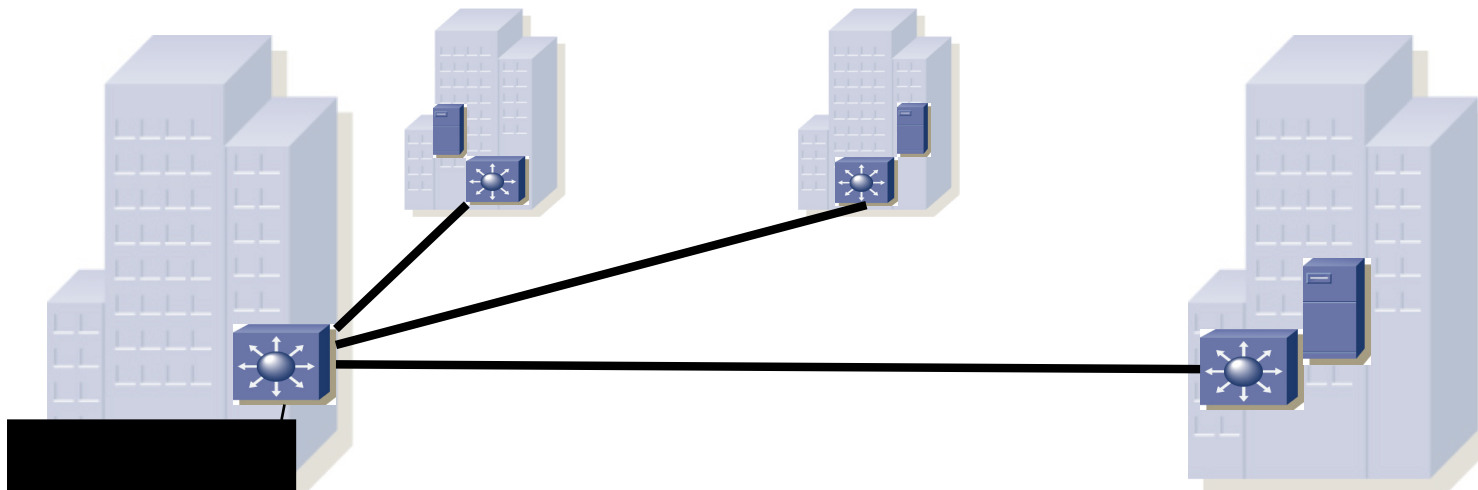
Rendimiento en la WAN

- 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



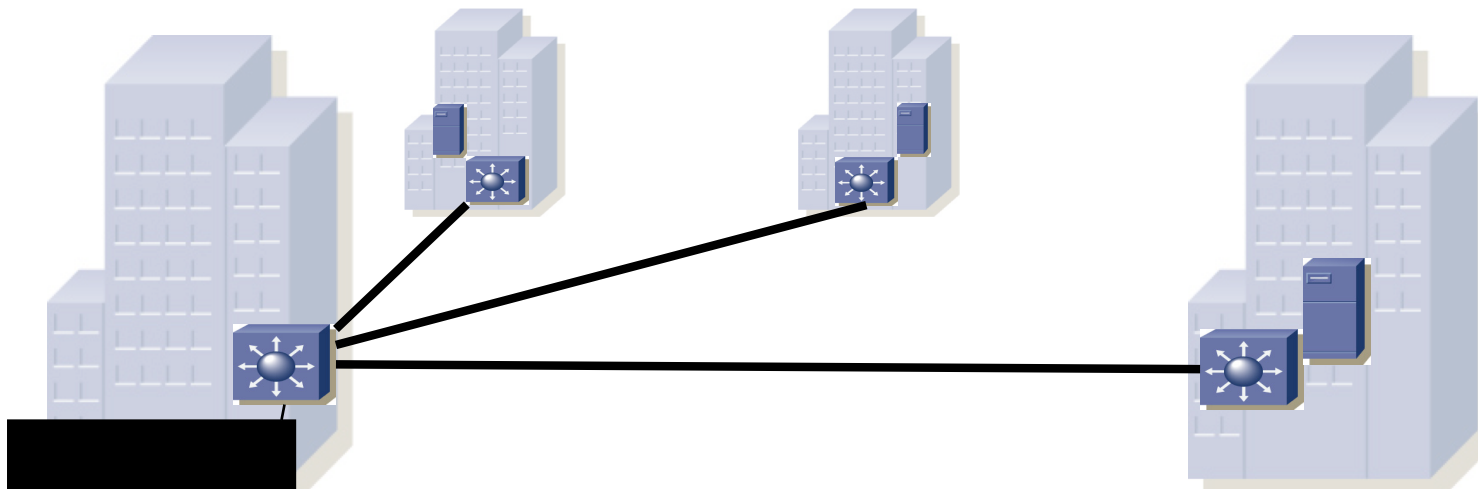
WAN vs LAN Bandwidth

- 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



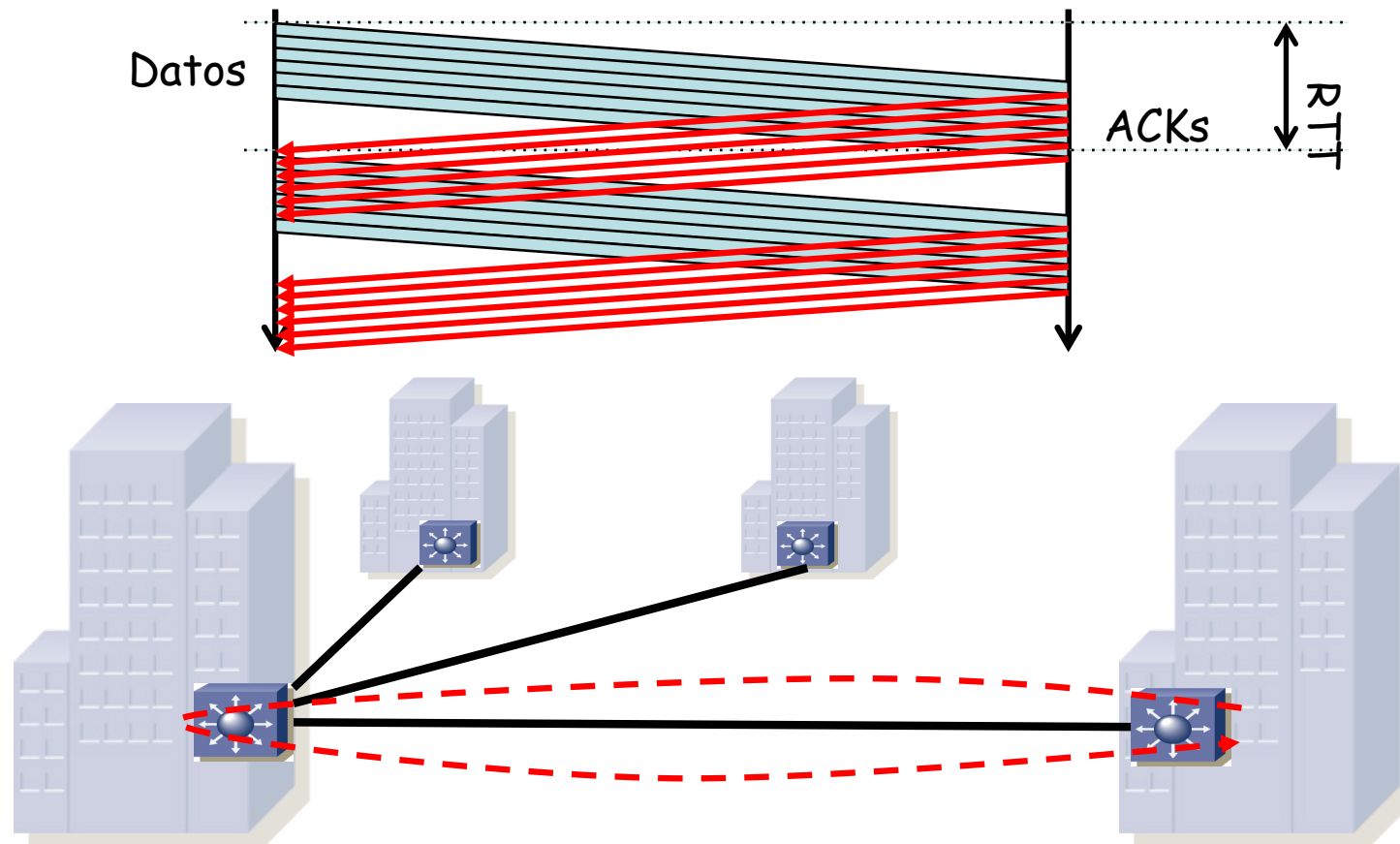
WAN vs LAN Bandwidth

- 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, autenticación, 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



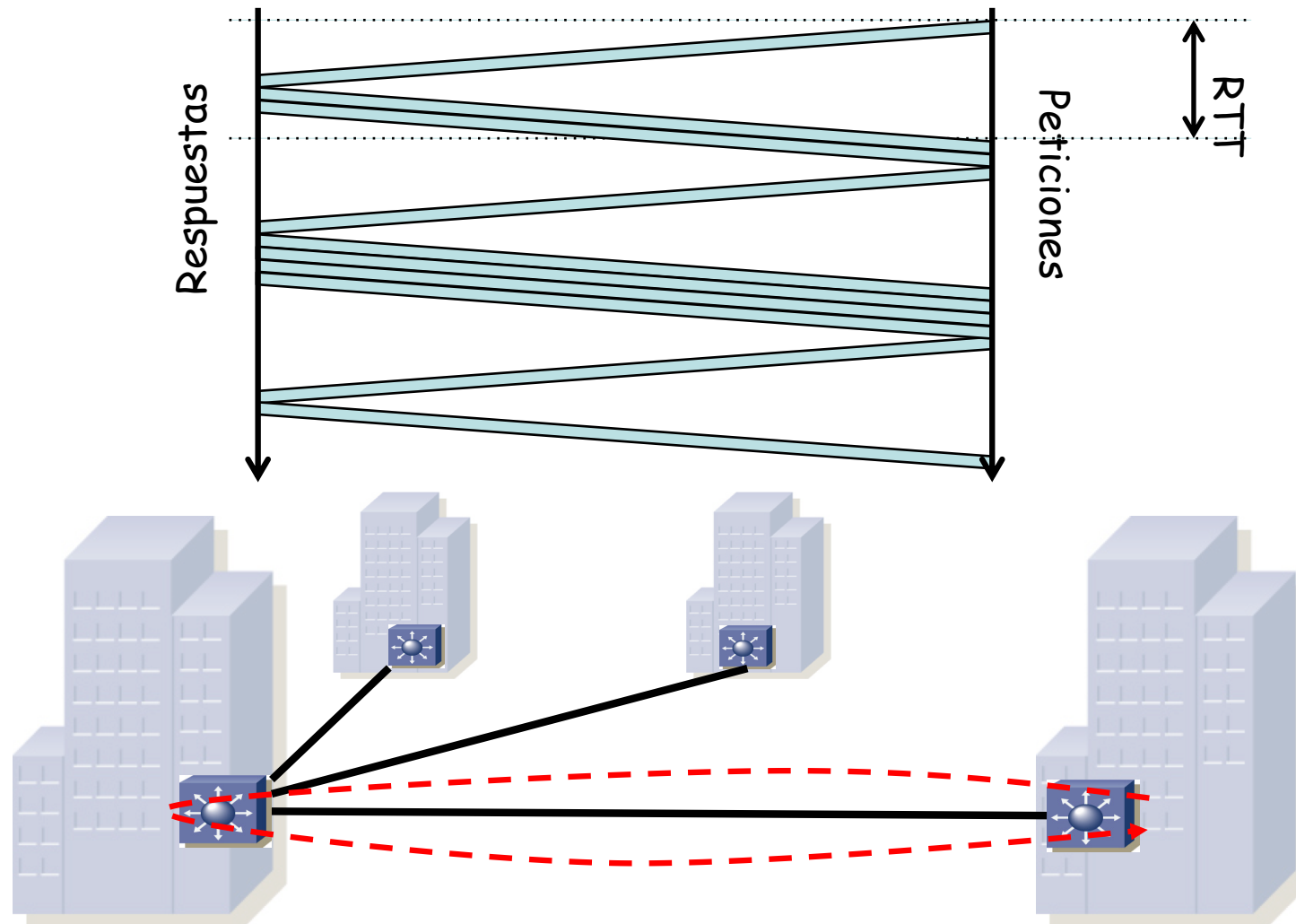
Latencia (retardo)

- 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)



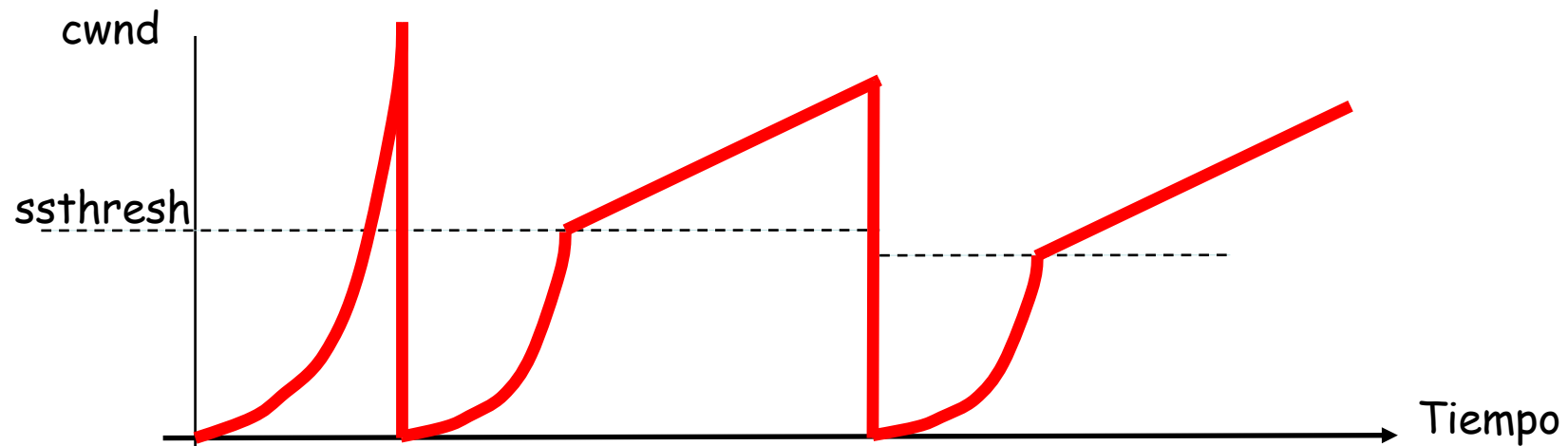
Latencia (retardo)

- 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)



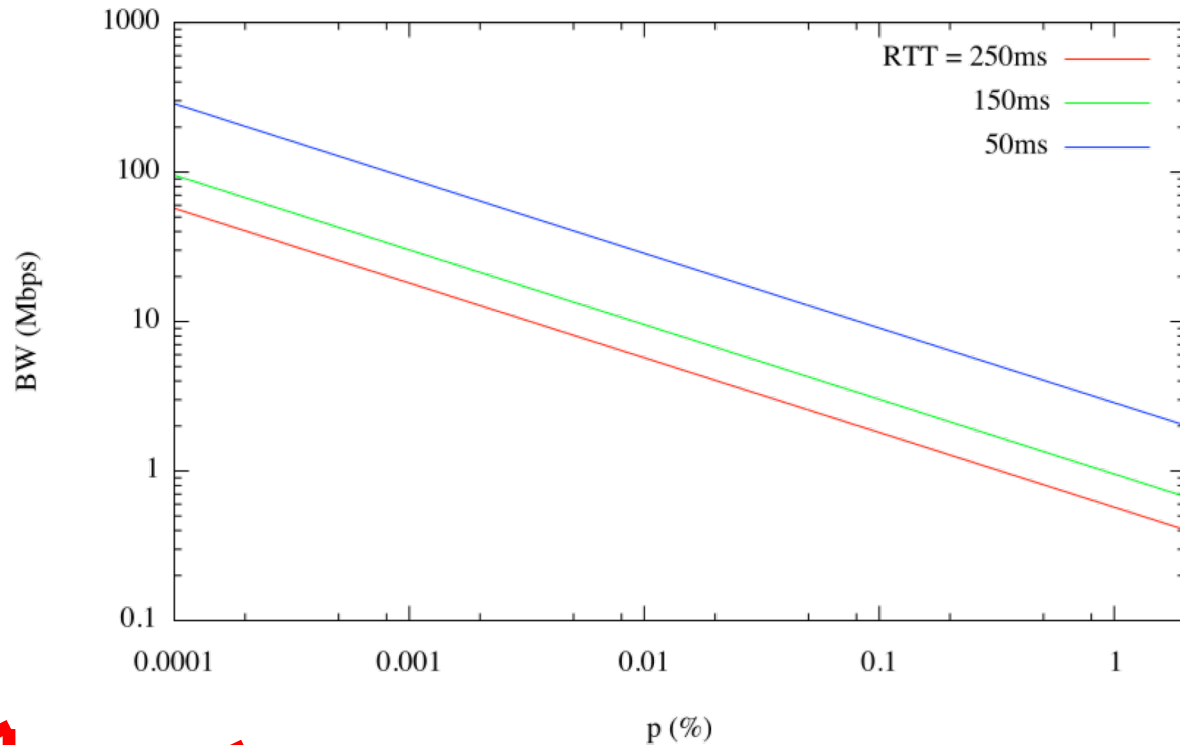
Pérdidas (TCP)

- 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

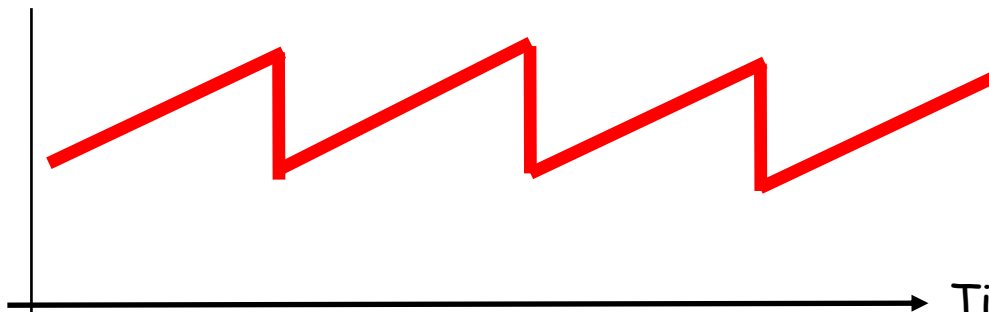
MSS = 1460 bytes

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

*M. Mathis
 (ACM SIGCOMM'97)*



cwnd



TCP Reno en
congestion avoidance

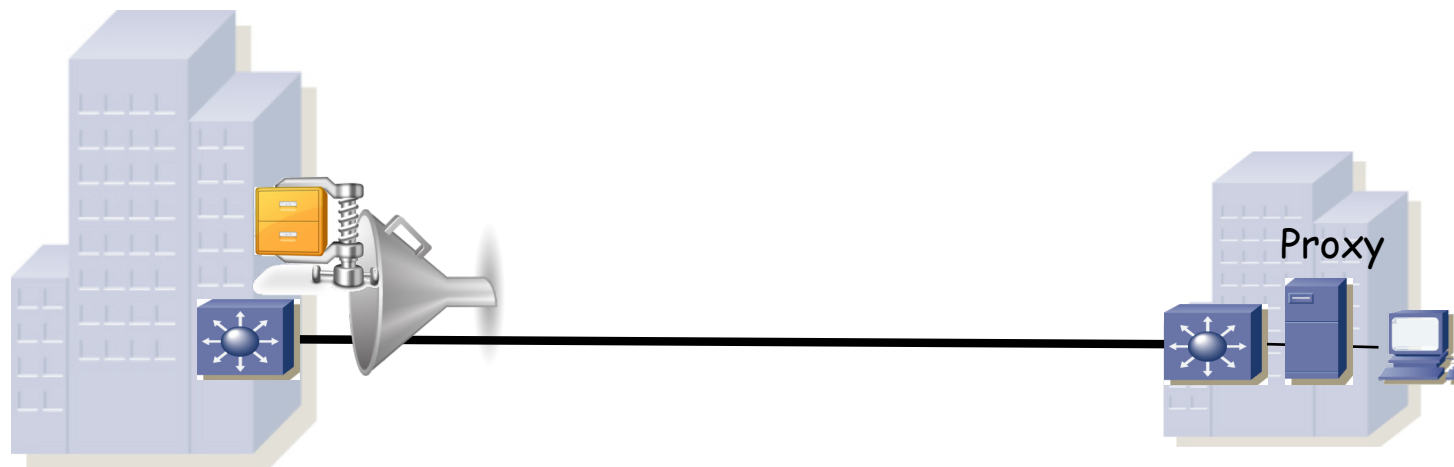
WAN optimization

- 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



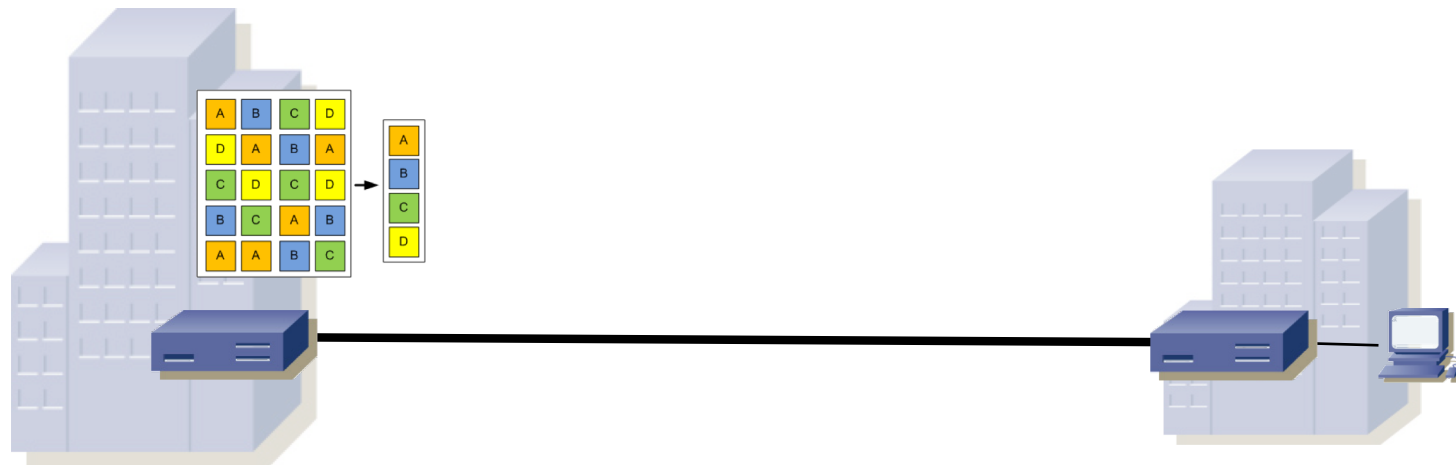
Compresión

- 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



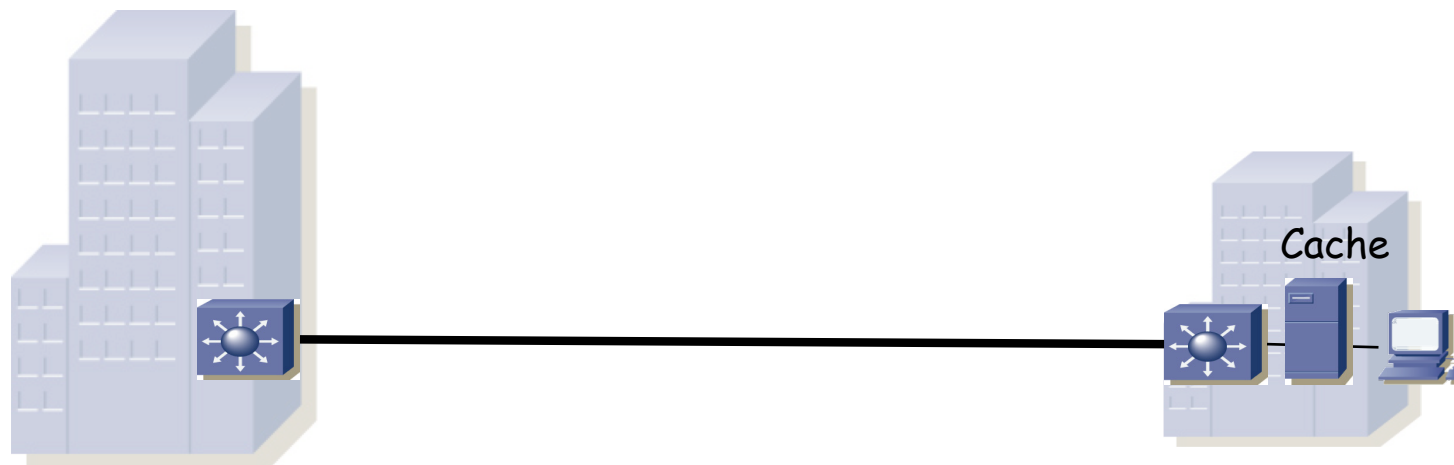
Data deduplicación

- 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

