
Content Distribution Networks (CDN)

Servicios en la web y distribución de contenidos
Doct: Tecnologías para la gestión distribuida de la información



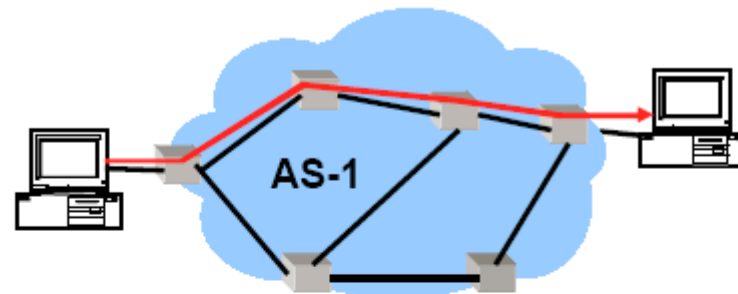


Índice

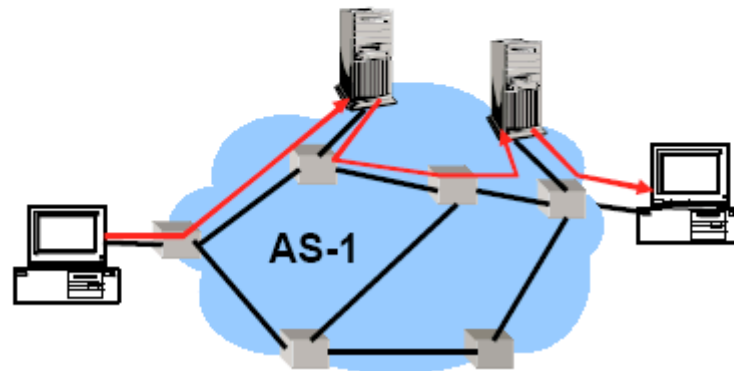
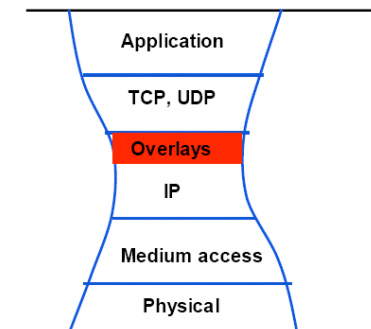
1. Introducción
2. Content Distribution Networks (CDN)
 - 2.1. CDN localización
 - 2.2. CDN redirección
 - 2.3. CDN replicación
 - 2.4. Estructura CDN
 - 2.5. Proveedores CDN
3. Proxy/cache
5. HTTP
6. Componentes del retardo
7. Ejemplo de investigación en CDNs+P2P

1. Introducción

- **Overlay Network:** red virtual de nodos y enlaces lógicos que se construye por encima de una red con el propósito de implementar un servicio de red que no está disponible en la infraestructura existente.



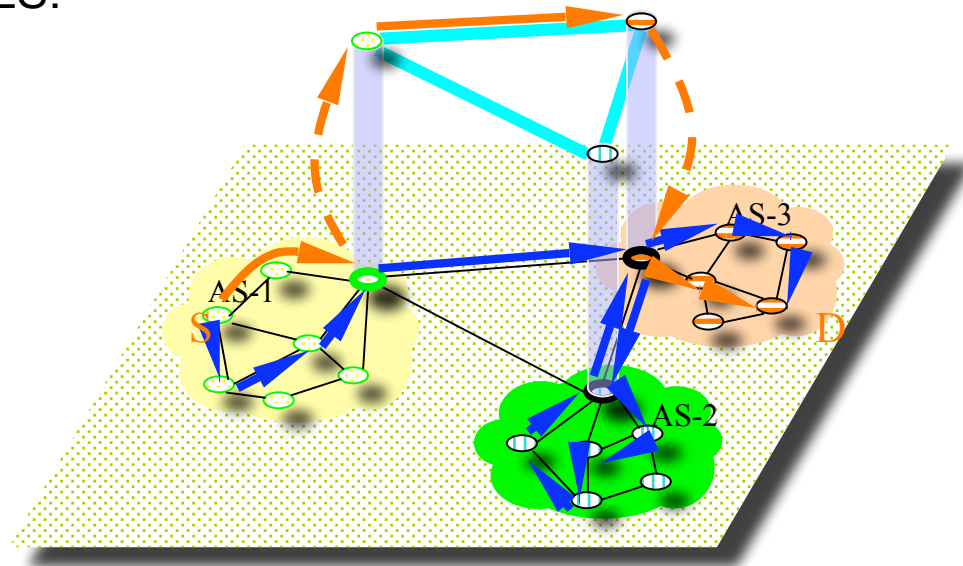
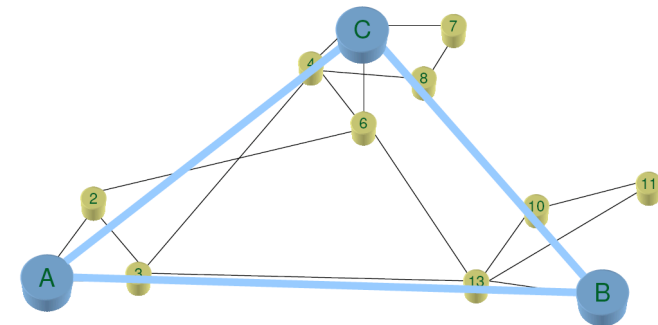
IP

Overlay Network
(over IP)

Introducción

■ Ejemplos de Overlay Networks:

- Redes P2P.
- Redes multicast.
- Redes de proxies.
- Redes de distribución de contenidos.**
- Rutas provenientes de una tabla de rutas.
- Red RSVP/Diffserv/MPLS.
- Mobile IP.



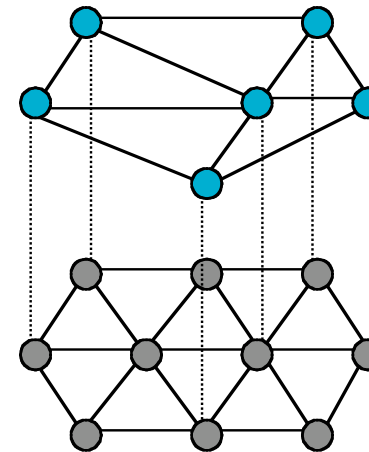
Introducción

■ Ventajas

- No es necesario nuevo equipamiento o modificar el software/protocolo de la red existente.
 - Se podrá desarrollar software/protocolos por encima de los existentes.
- No tiene por qué implantarse en todos los nodos.
 - Dependerá de las funcionalidades deseadas, la sobrecarga que introduce, problemas de seguridad y escalabilidad.

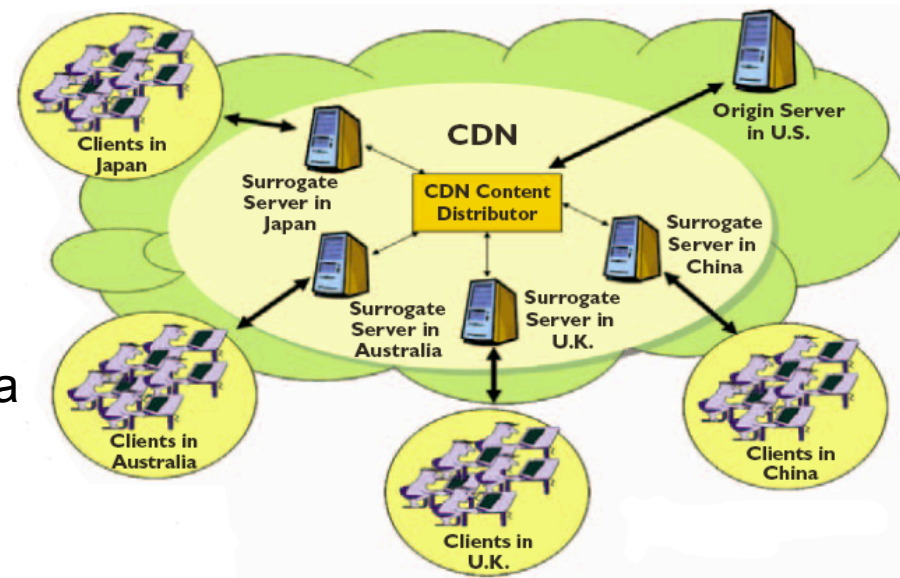
■ Inconvenientes

- Añade sobrecarga
 - Nuevas capas de pilas de protocolos.
- Añade complejidad.



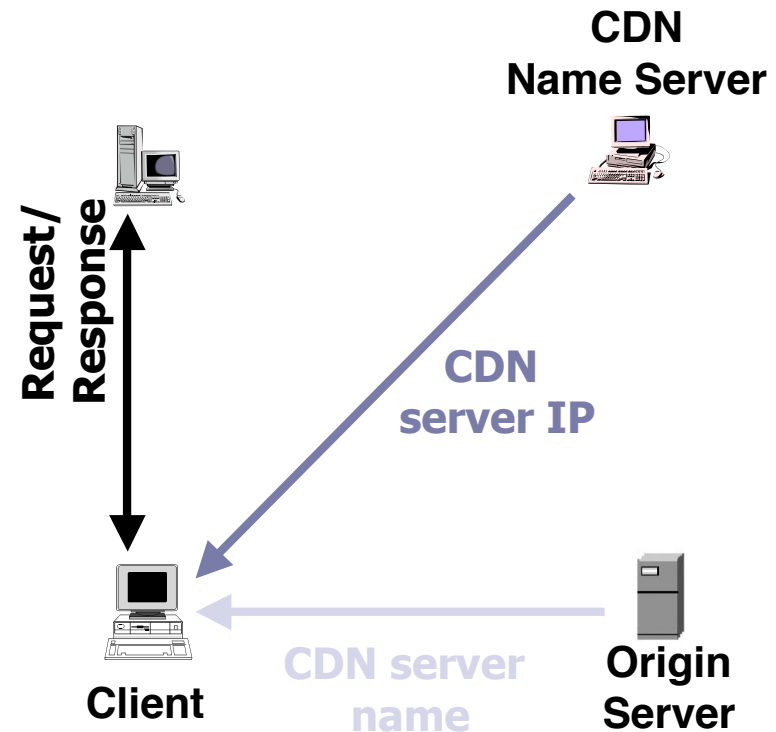
2. Content Distribution Networks (CDN)

- Servidores que poseen la misma información replicada que el original y atienden peticiones de usuarios por algún criterio que:
 - Equilibre la carga entre servidores.
 - Minimice la latencia en el volcado de contenidos al cliente.
- Facilitan las transferencias de ficheros grandes o rapidez en las consulta.
 - Los contenidos se descargan en los servidores del CDN.
 - Colocar los servidores cerca de los usuarios minimiza retardo y pérdidas.
 - El servidor CDN se coloca normalmente en el borde de la red de acceso.



Content Distribution Networks

- Características generales
 - El proveedor del servicio CDN puede ser diferente del proveedor de los contenidos (Ej: Akamai <-> CNN).
 - La actualización de los contenidos es automática y simultánea en todos los servidores CDN.
- CDN crea un mapa sobre cercanía entre sus servidores e ISPs (usuarios finales).
- Normalmente sirven contenidos estáticos
 - La dinamicidad del servidor original es difícil de replicar.
 - Habitualmente se encargan de imágenes, vídeos, publicidad, etc.





2.1. CDN localización

- Un CDN **localiza** su servidor sustituto (surrogate server) más “cercano” al cliente según criterios diversos:
 - Zona geográfica.
 - Normalmente colocados en POPs (Points of Presence) de diferentes ISPs (Internet Service Providers).
 - Topología.
 - Número de saltos (routers) entre cliente y servidor.
 - Retardo.
 - Medido de manera activa o pasiva.
 - Carga de los servidores.
 - En CPU o ancho de banda consumido en su acceso.



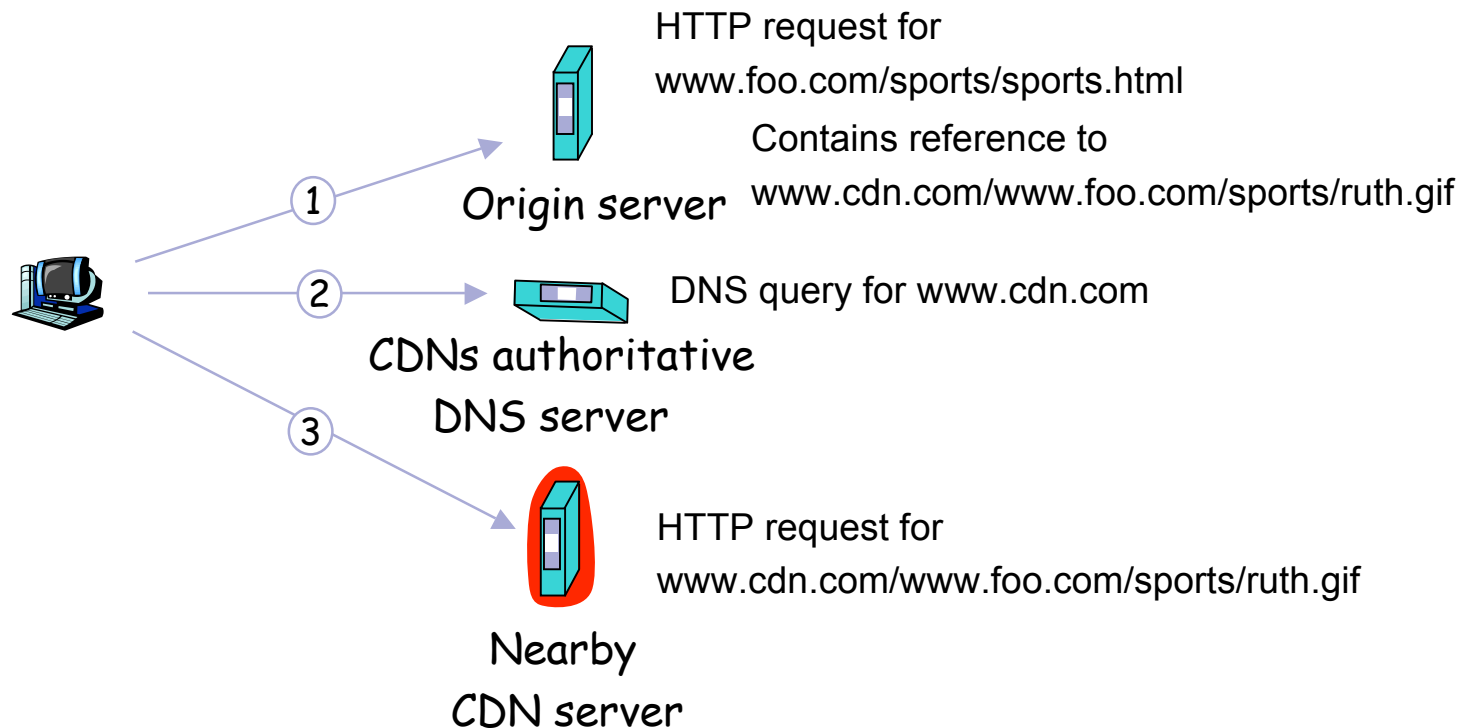
2.2. CDN redirección

- Un CDN **redirecciona** la petición del cliente al servidor CDN más cercano de manera transparente al usuario – balanceo de carga. Técnicas:
 - Redirección DNS
 - Reescritura de URL
 - Combinación de redirección y reescritura

- 1) Redirección DNS (Domain Name Service)
 - Los servidores DNS autoritativos son controlados por el CDN.
 - El nombre DNS del recurso se resuelve en una IP u otra del servidor CDN según interese (disponibilidad de recursos y condiciones de red).
 - Habitualmente se establecen tiempos de validez del mapeo (TTL) cortos, para que pueda hacer un balanceo mejor según las condiciones cambiantes de la red y servidores (10-30sg).
 - Tipos:
 - Sitio completo: se delega por completo la resolución del nombre del servidor original en los servidores DNS del CDN. El original queda oculto a los clientes.
 - Sitio parcial: el servidor original modifica las URL embebidas de objetos (principalmente imágenes) para que hagan referencia a nombres propios resolubles por los servidores DNS del CDN.

Ejemplo CDN redirección DNS parcial

- Proveedor de contenidos (www.foo.com)
 - Distribuye las páginas HTML.
- Proveedor CDN (www.cdn.com)
 - Distribuye las imágenes gif.
 - Transparente al usuario al utilizar redirecciones de DNS.





CDN redirección

- 2) Reescritura de URL (Uniform Resource Locator)
 - El servidor original reescribe dinámicamente las URLs embebidas en sus páginas según criterios (disponibilidad de recursos y condiciones de red) para redireccionar el cliente a diferentes servidores CDN.
 - A la hora de servir la página, el servidor original reescribe la URL colocando la dirección IP concreta del servidor CDN al que se quiere direccionar, evitando la necesidad de búsqueda de DNS.

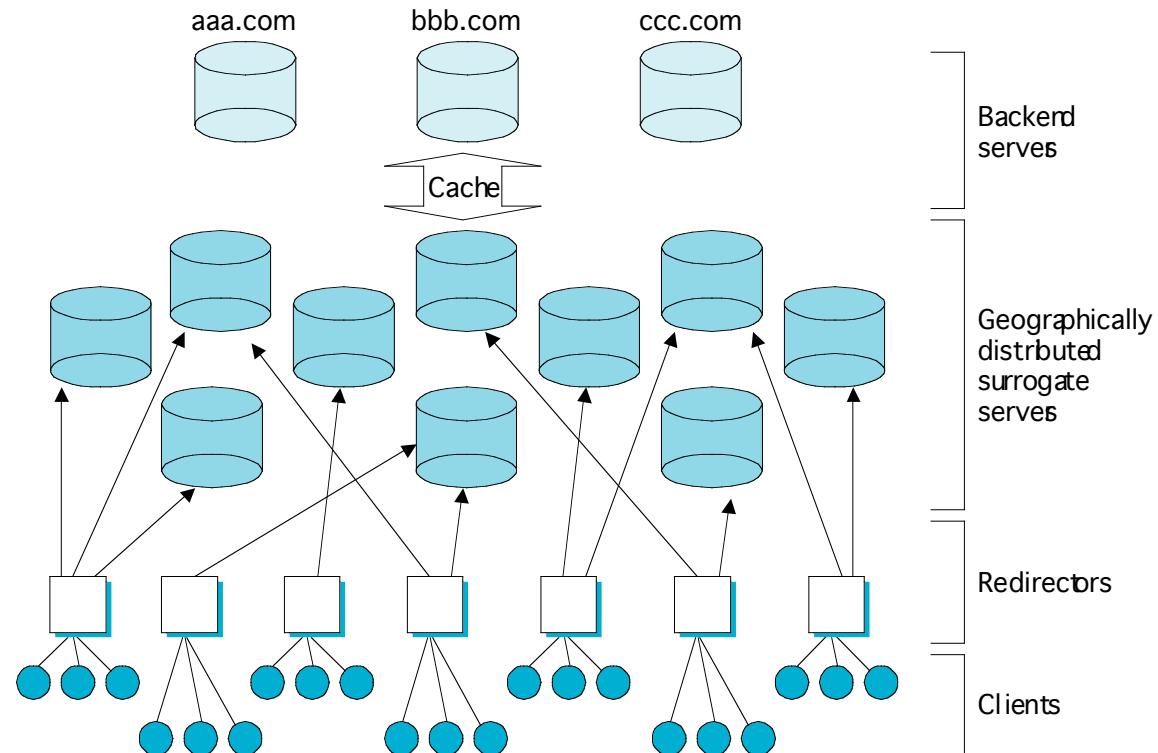
- 3) Combinación de redirección y reescritura
 - La reescritura de URL puede colocar el nombre DNS del servicio CDN en lugar de su IP de manera que quede en manos de los servidores DNS del CDN la redirección final.



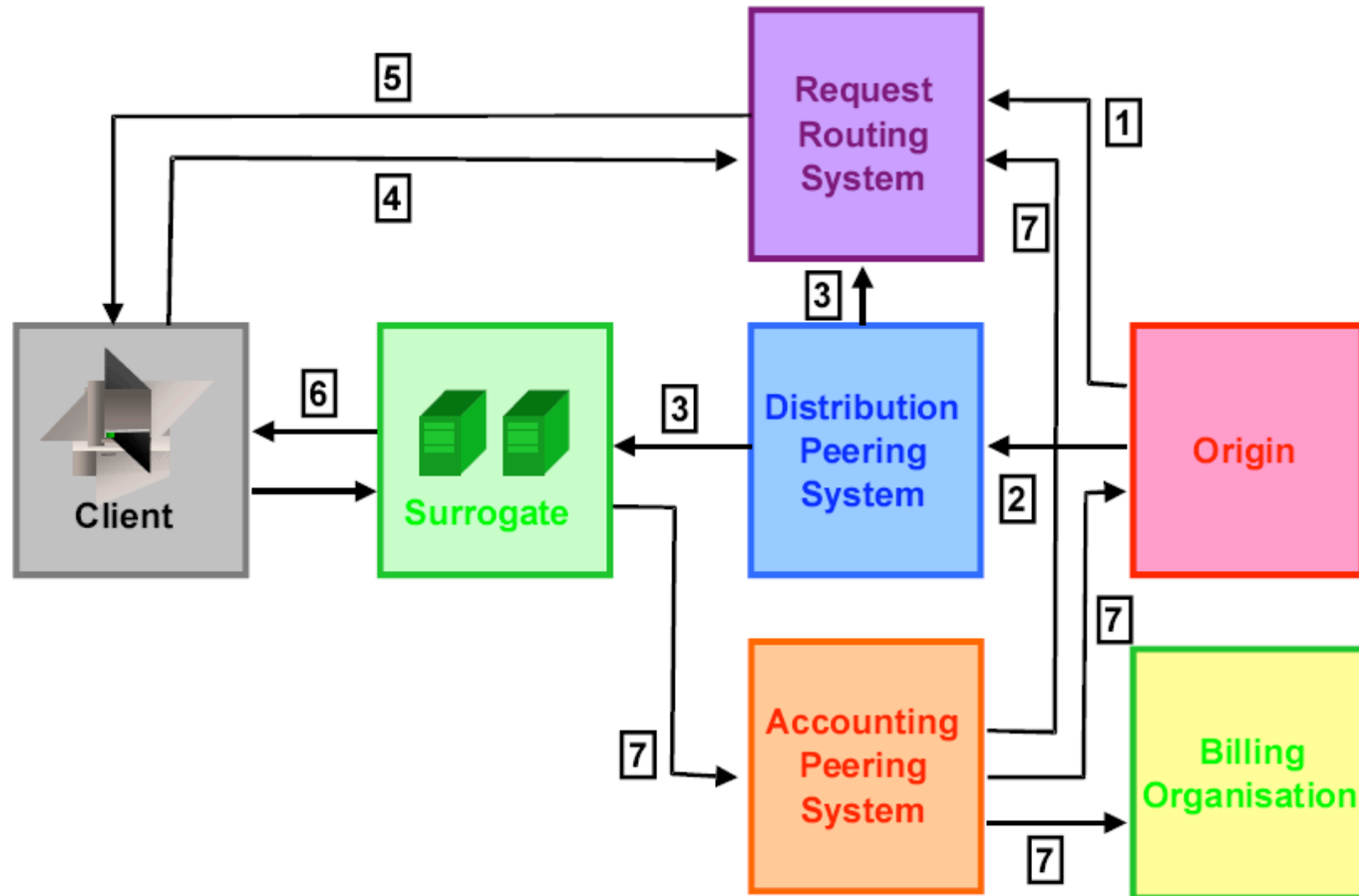
2.3. CDN replicación

- Un CDN **replica** los contenidos del servidor original de distintas formas:
 - Push-based
 - El servidor original descarga los contenidos a replicar en un sistema de distribución encargado de hacer las réplicas en los servidores surrogate del CDN.
 - Pull-based
 - El cliente es redireccionado al servidor surrogate del CDN que le corresponda y en caso de no tener los contenidos solicitados por éste los solicita al servidor original o a otros servidores surrogate del CDN, cacheándolos para futuras referencias. Trabaja a modo de la caché de un proxy.

2.4. Estructura CDN



Estructura CDN



Fuente: "CDN Peering Architectural Overview" IETF Internet draft



Estructura CDN

- 1. El servidor original delega su espacio de nombres URI para ser distribuido por el CDN a través del sistema de enrutado de peticiones.
- 2. El servidor original publica el contenido en un sistema de distribución.
- 3. El sistema de distribución replica el contenido en todos los servidores surrogate del CDN.
- 4. El cliente solicita un contenido con su URI como si se refiriera al servidor original.
- 5. El sistema de enrutado de peticiones redirecciona la petición al servidor CDN (surrogate) que convenga.
- 6. El servidor CDN atiende la petición del cliente.
- 7. La transacción es controlada por un sistema de contabilidad del CDN encargado del tema estadístico de utilización y facturación.



2.5. Proveedores CDN

Adero
(www.adero.com)



Akamai
(www.akamai.com)



Cidera
(www.cidera.com)



Pushcache
(www.pushcache.com)



EpicRealm
(www.epicrealm.com)



CacheWare
(www.cacheware.com)



Clearway
(www.clearway.com)



IBeam
(www.ibeam.com)



Mirror Image
(www.mirror-image.com)



Digital Island
(www.digitalisland.com),
includes the former Sandpiper
Networks





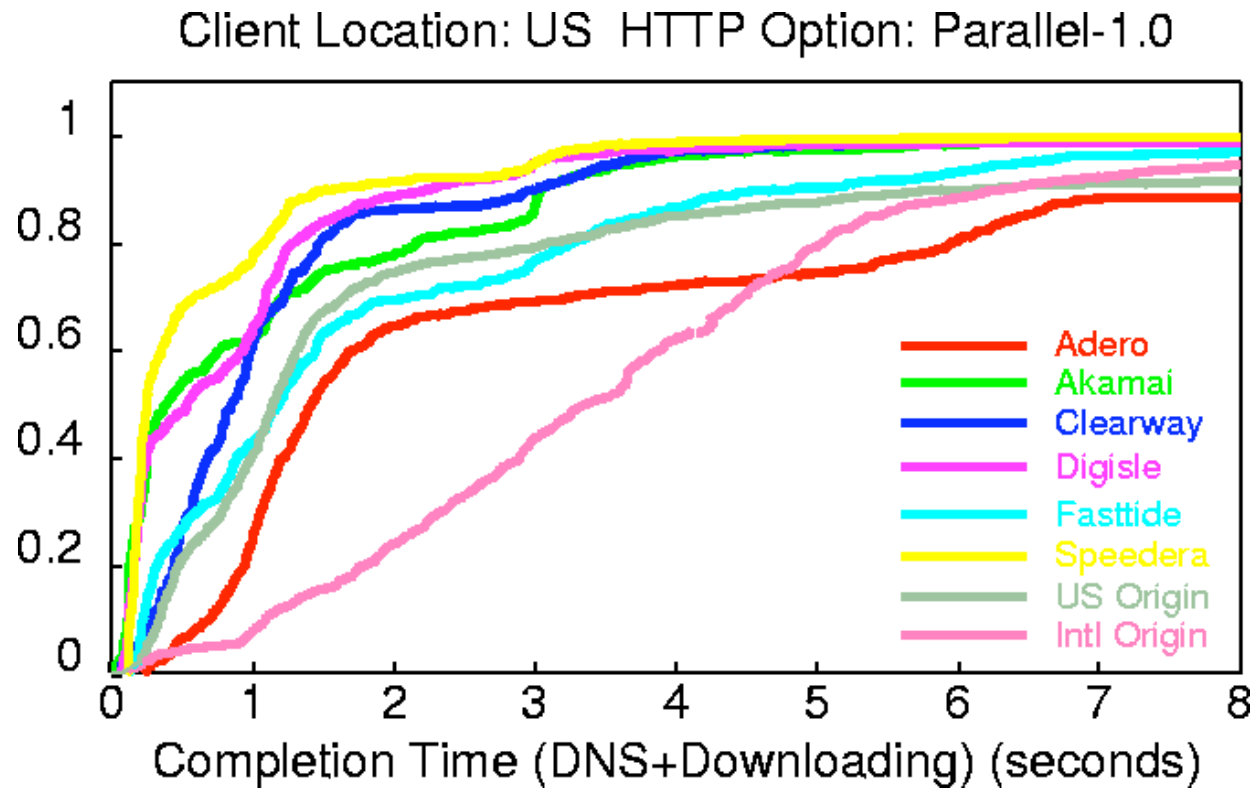
Proveedores CDN

Table 1. Content delivery network provider characteristics.

CDN service provider	Service type	Content distribution	Fees	Customers
Akamai www.akamai.com	Multi-ISP, partial-site request servicing, peering	More than 12,000 surrogate servers spanning 1,000 networks in 62 countries	US\$1,995 per month for each Mbps of delivered content	Covers 70 percent of the market, with more than 3,600 customers including Apple, CNN, MSNBC, Reuters, and Yahoo
Adero www.webvisions.com/ adero/	Multi-ISP, full-site request servicing, peering	Surrogate servers in more than 30 countries	Depends on resellers (CDNs that buy Adero services)	Serves 30 customers, including resellers Exodus and UUNET
Digital Island www.sandpiper.net	Multi-ISP, partial-site request servicing, peering	2,500 surrogate servers spanning 327 networks in 35 countries	Starts at US\$1,500 per month	More than 900 customers including AOL, Canon, Cisco Systems, Microsoft, and Hewlett Packard
Mirror Image www.mirror-image.com	Multi-ISP, partial-site request servicing, peering	22 surrogate servers in North America, Europe, and Asia	US\$2,100 per month for each Mbps of delivered content	More than 200 customers including Creative, Open Systems, and SiteRock
Inktomi www.inktomi.com	Single-ISP, full-site request servicing, peering	10 surrogate servers across China	Starts at US\$4,000 per month	13 CDNs including Adero and Digital Island and more than 200 Web sites



Proveedores CDN



Akamai

- En 2002, 10.000 servidores en 53 países.
- Servidores en POPs de proveedores más importantes.
- Dispone de su propia red de servidores de DNS.
- Basado en combinación redirección DNS y reescritura
 - URL embebida: ARL (Akamai Resource Locator)

http://www.foo.com/a.gif



http : //a ^{Serial #} *836* ^{Akamai Domain} *.g.akamaitech.net* ^{Type} */ 7* ^{Serial #} */ 836* ^{Provider Code} */ 123* ^{Object Data} */ e358f5db0045* ^{absoluteURL} */ www.foo.com/a.gif*

- Serial: identifica a un conjunto de servidores CDN.
- Provider code: identificador del cliente (servidor original).
- Object data: hash del objeto, para detectar versiones del mismo.

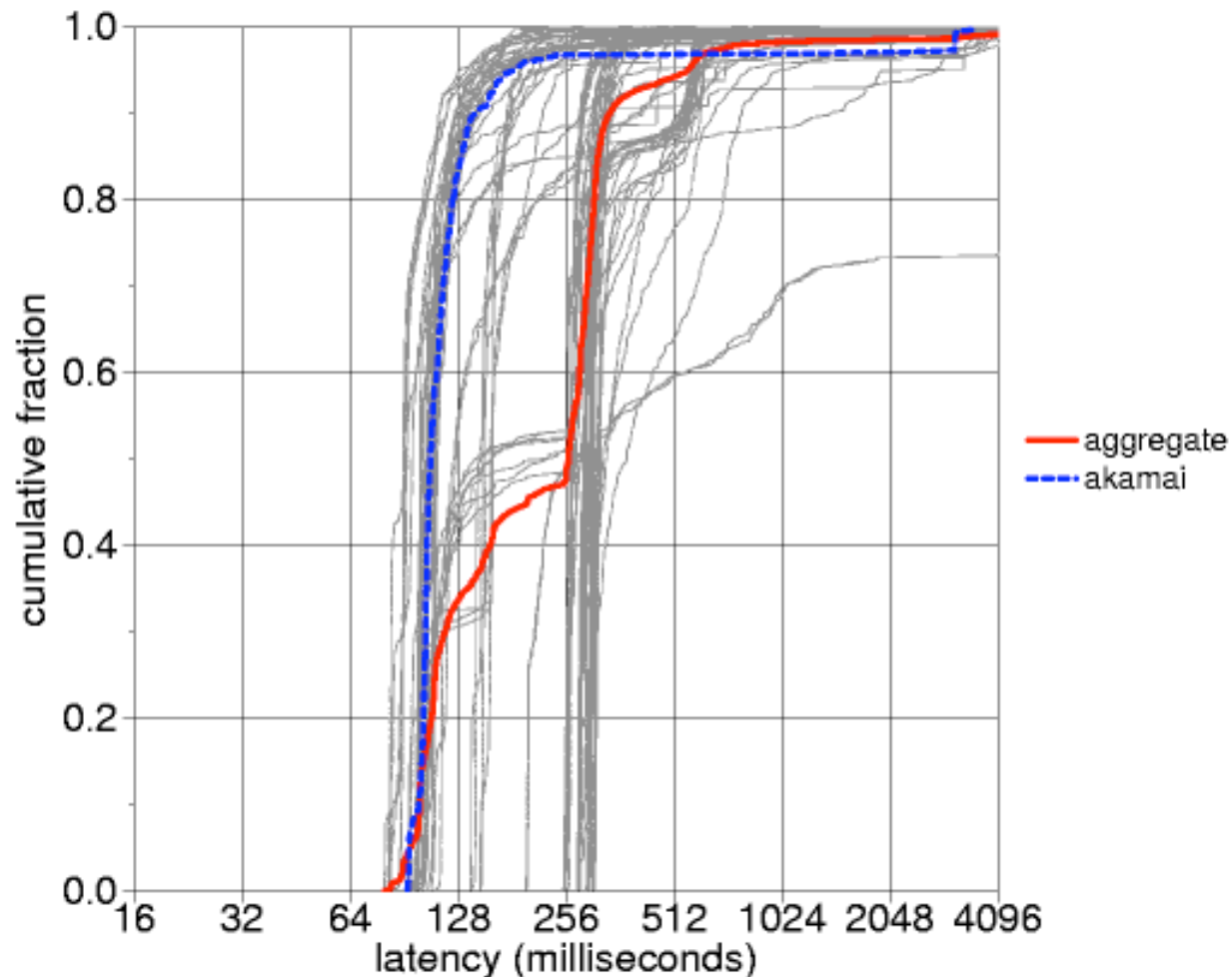


Akamai

□ Redirección DNS

- Tiene en cuenta la IP origen que realiza la petición (localización).
- Utilizar una red de servidores DNS en 2 niveles:
 - high-level.akamai.net servers (HLDNS)
 - low-level.g.akamai.net servers (LLDNS)
- Los servidores raíz de Internet (*.net) se encargan de devolver al cliente las IPs de HLDNS de las que selecciona una,
 - DNS raíz, TTLs de alrededor de 2 días.
- Los servidores de alto nivel son los que reciben las peticiones de resolución (*.akamai.net) y la redireccionan a servidores de segundo nivel (*.g.akamai.net) que estén lo más cercano posible al cliente.
 - HLDNS, TTLs de alrededor de 20 minutos.
- El servidor de bajo nivel es el que hace la resolución final devolviendo la IP del servidor CDN “más cercano” para ese cliente. Basado en mapas de la red que incorporan el estado de la red y de los servidores, actualizados cada 2-10 segundos.
 - LLDNS, TTLs de alrededor de 20 segundos.
- TTLs compromiso entre disminuir el tráfico de peticiones DNS y ahorrarse el tiempo de resolución, con el de tener la información lo más actualizada posible.

Akamai, eficiencia en la localización



[Johnson00]

Distribución acumulada del retardo.

- Gris: para cada servidor. Gran variabilidad.
- Rojo: media de todos los servidores.
- Azul: Akamai, no el mejor.



3. Proxy/cache

- Un servidor proxy es una entidad de red en la que los clientes delegan la realización de solicitudes HTTP.
- Habitualmente usado en contenidos web pero también para contenidos de streaming.
- Ventajas del uso de proxys:
 - Reducción del tiempo de respuesta.
 - Reducción del tráfico web a través de la red de acceso a Internet, y por tanto reducción del coste de conexión.
- Inconvenientes:
 - Contenidos estáticos.
 - Útil para ficheros medianos y grandes.
 - Número limitado de clientes conectados a cada proxy y el aumento continuo de webs hace que el índice de peticiones que se encuentran almacenadas en la caché sea pequeña (del 30-50% en los mejores casos).



Proxy/cache

Secuencia de uso de un proxy:

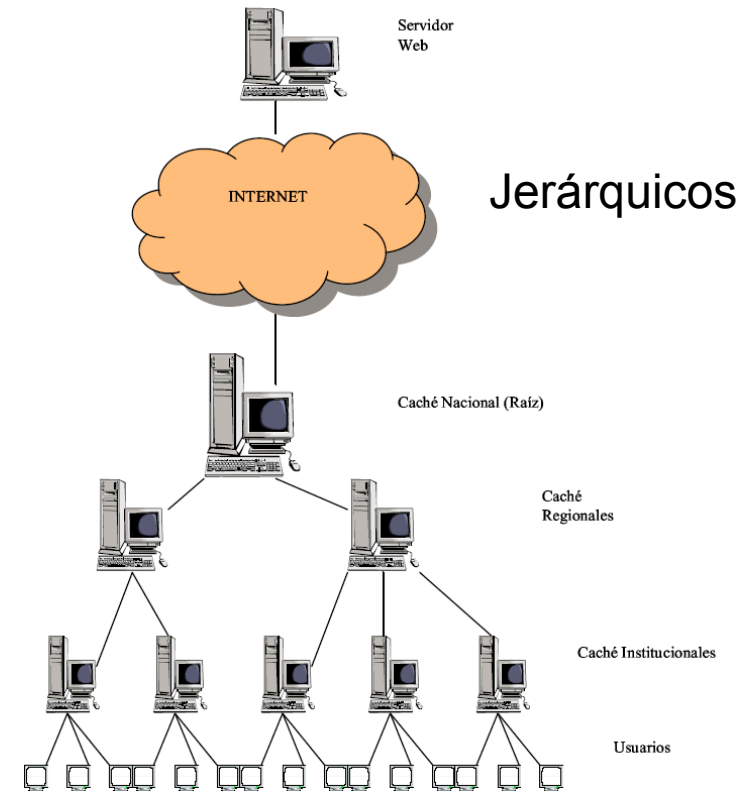
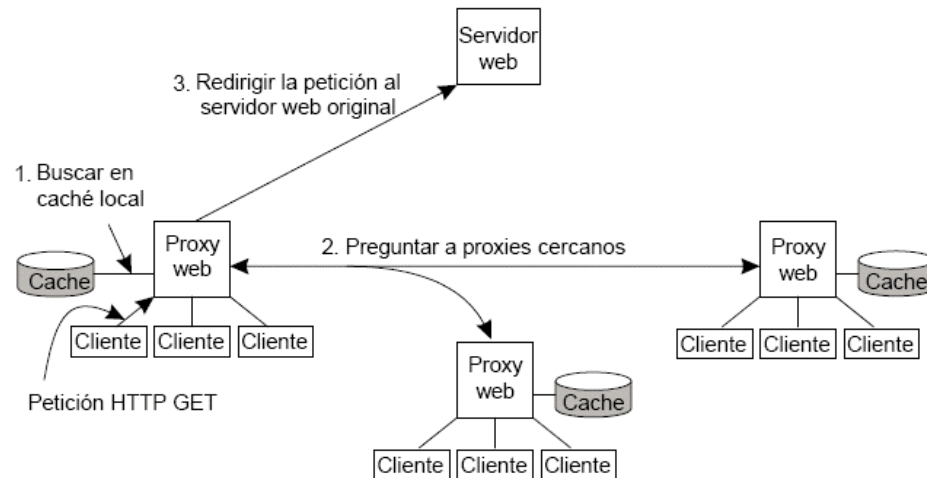
- 1. El navegador establece una conexión con el proxy y le envía la solicitud HTTP. La cabecera Host indica el servidor original.
- 2. El proxy comprueba si tiene una copia del objeto almacenada localmente. En tal caso, le envía el objeto al navegador en un mensaje HTTP de respuesta.
- 3. Si el proxy no tiene el objeto, abre una conexión con el servidor original y le envía una petición HTTP del objeto requerido por el navegador. El servidor original envía la respuesta al servidor proxy.
- 4. El proxy almacena una copia en su caché y redirige otra copia en una respuesta HTTP al navegador, por la conexión establecida en el primer paso.

Proxy/cache

■ Tipos de coordinación

- Jerárquicos
- Cooperativos

Cooperativos



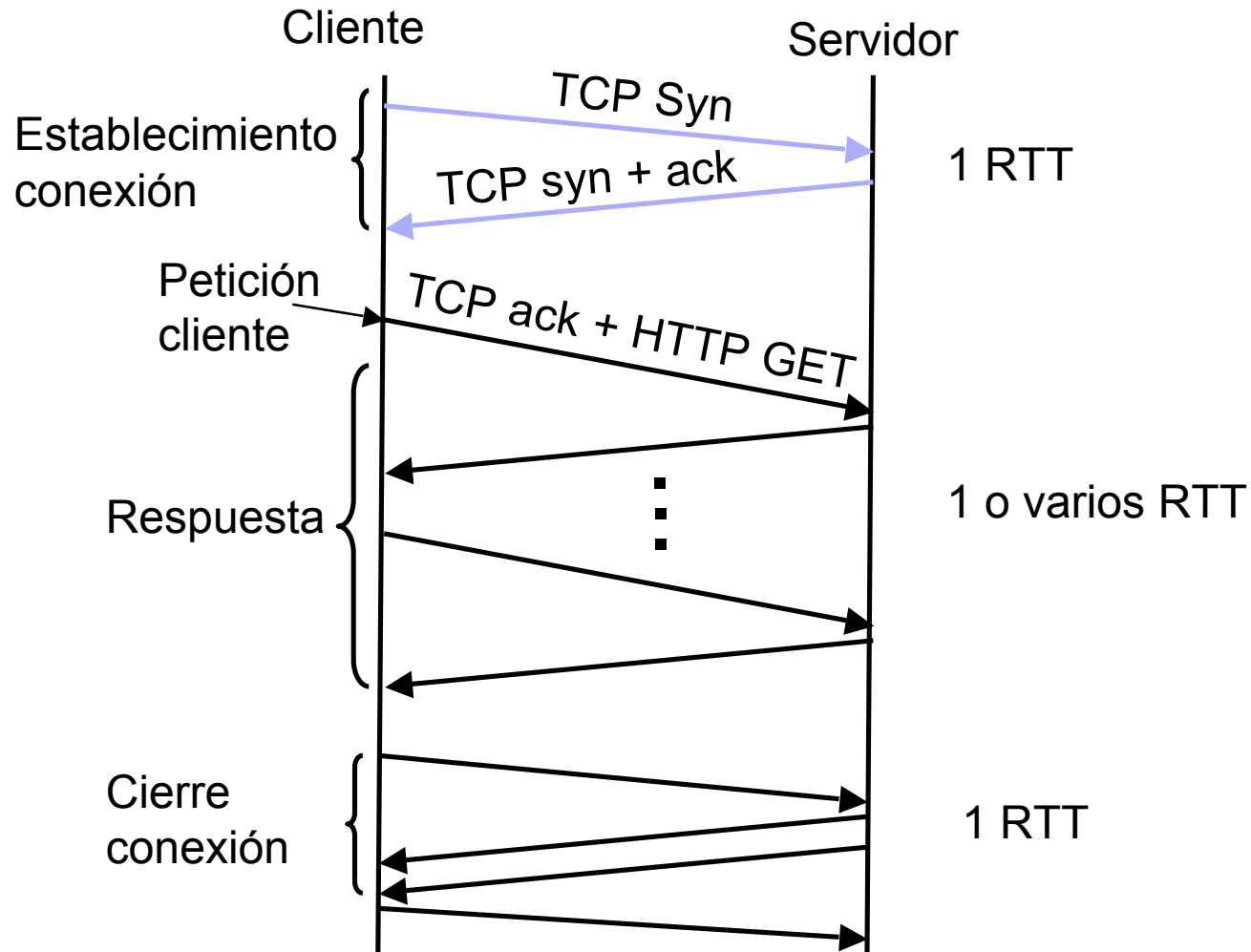


5. HTTP

- World Wide Web (WWW): una base de datos distribuida de páginas enlazadas a través de [Hypertext Transport Protocol \(HTTP\)](#)
- Emplea TCP como protocolo de transporte
- Comunicación cliente-servidor:
 - El cliente envía un mensaje al servidor solicitando un recurso
 - El servidor recibe el mensaje, lo procesa, y responde con un mensaje HTTP de respuesta, que contiene el recurso solicitado
- El protocolo HTTP es un protocolo sin estado y modo texto.
- Versiones:
 - HTTP/1.0 –1992, RFC 1945
 - Información cliente/servidor, cacheo simple
 - HTTP/1.1 – 1996, RFC 2616
 - HTTP/1.1 es compatible hacia atrás con HTTP/1.0
 - Conexiones persistentes.
 - Transferencia de partes de ficheros (byte-range header).

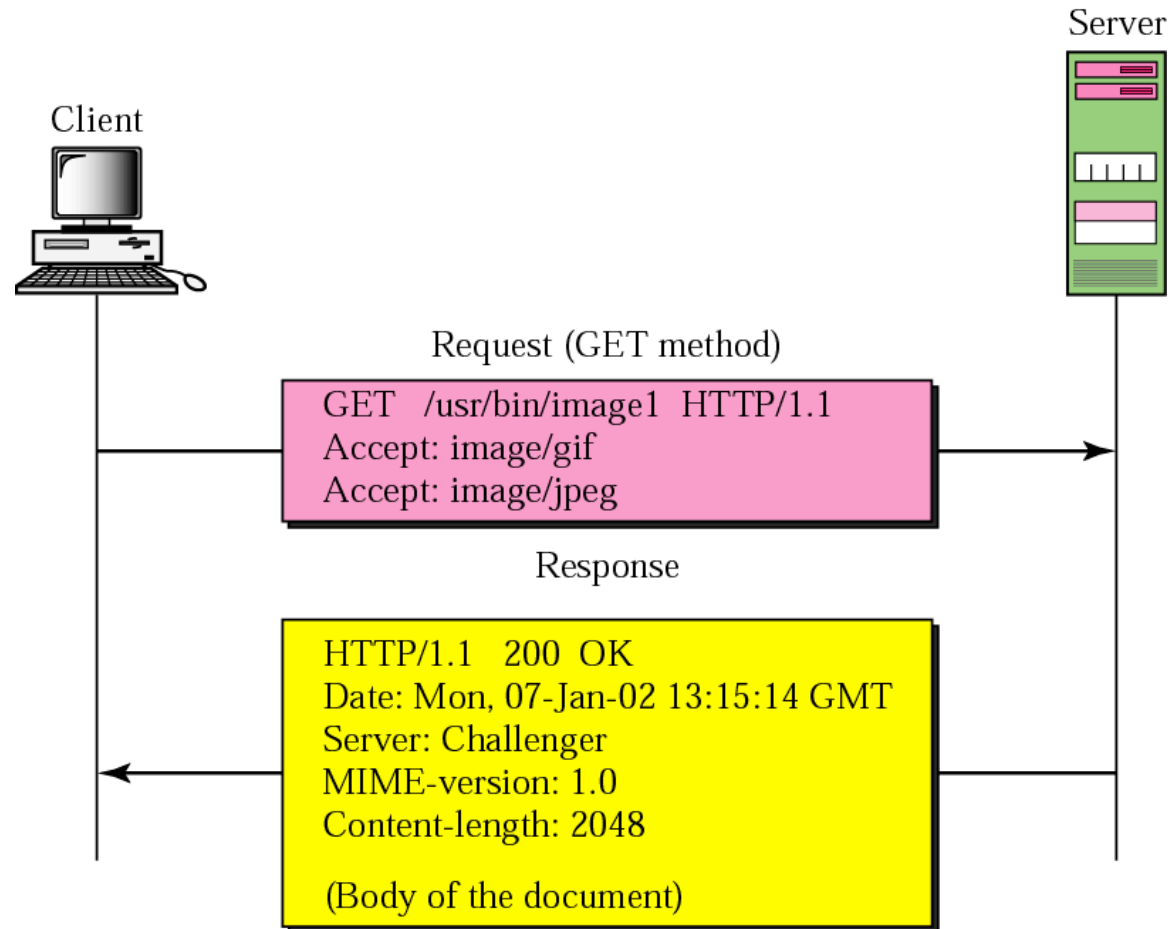


Conexión HTTP



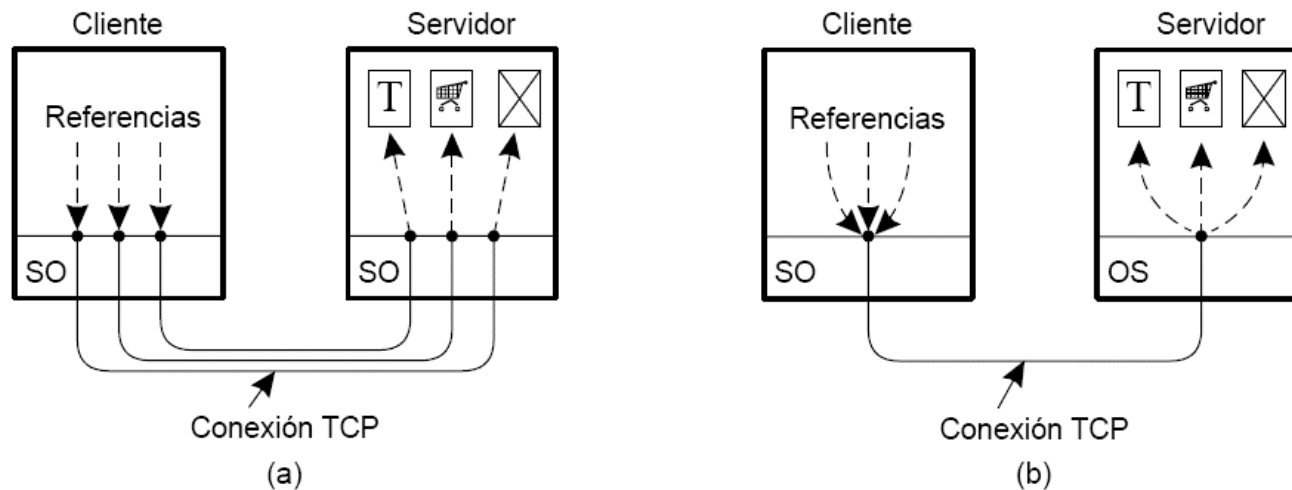


Conexión HTTP



HTTP

- Una conexión es no persistente si cada conexión TCP se cierra después de que el servidor envíe un objeto.
 - HTTP/1.0 usa conexiones no persistentes (a)
 - HTTP/1.1 usa conexiones persistentes por defecto (b) o no persistentes
- Los navegadores modernos pueden emplear cierto grado de paralelismo en las conexiones TCP.



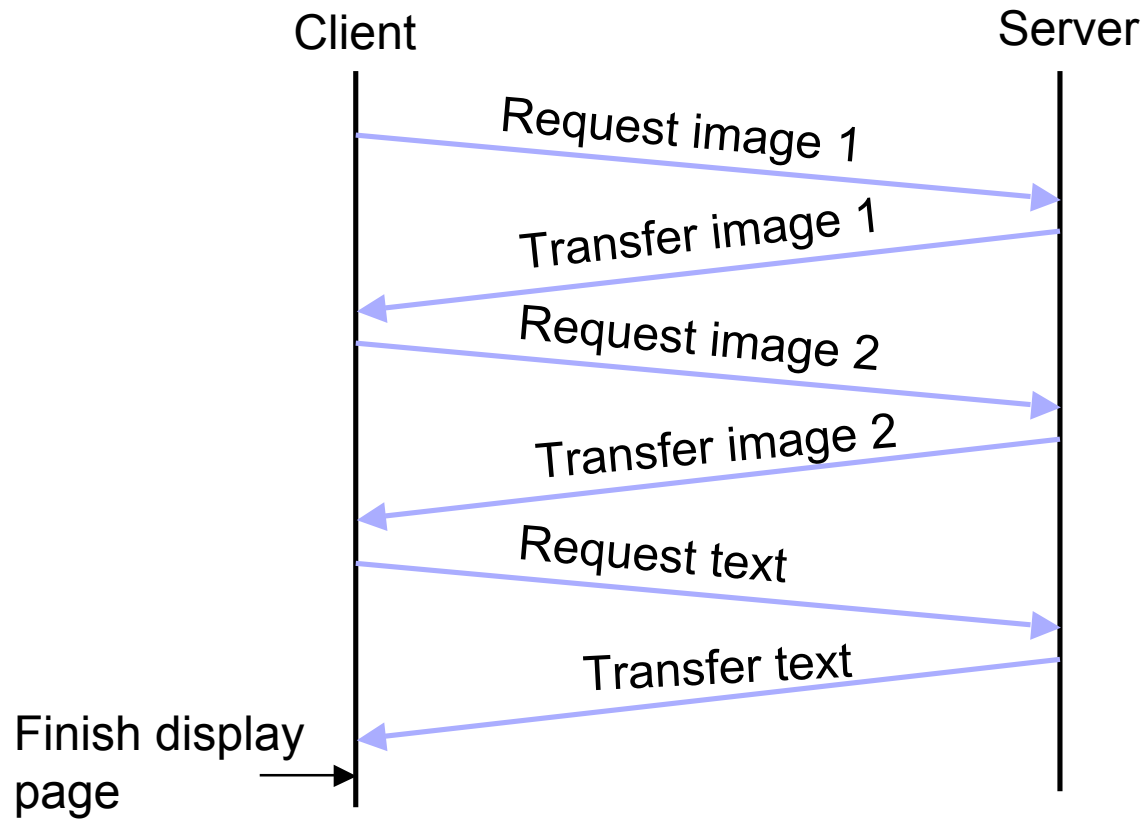


HTTP/1.0 Conexiones no persistentes

- Se crea una nueva conexión TCP por cada recurso
 - Alto número de objetos embebidos en una página web.
 - Muchas conexiones de duración corta.
- Transferencia TCP
 - Demasiado lenta para objetos pequeños.
 - Las conexiones TCP pueden no terminar nunca la fase slow-start.
- Varias conexiones se pueden realizar en paralelo (5 por defecto en la mayoría de navegadores)



HTTP/1.0 Conexiones no persistentes





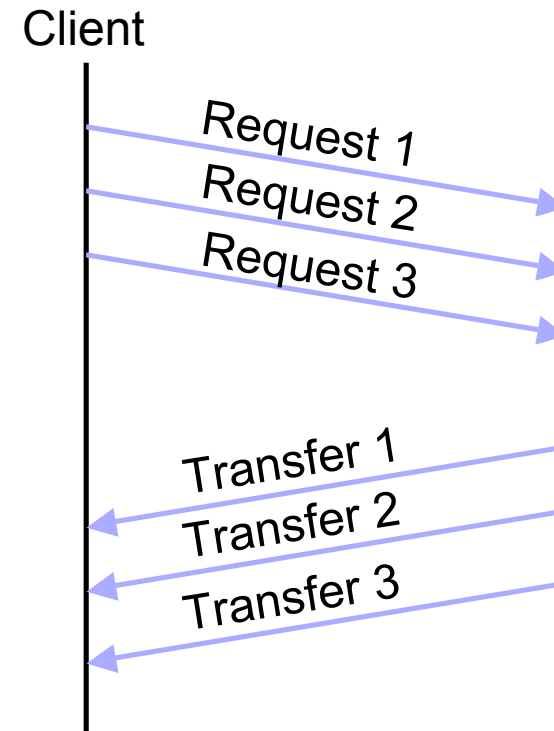
HTTP/1.1 Conexiones persistentes

- Un servidor HTTP que use conexiones persistentes cierra las conexiones cuando no se usan durante cierto tiempo configurable.
- Tipos de conexiones persistentes:
 - Sin pipelining: el cliente envía una nueva petición sólo cuando ha llegado la respuesta anterior.
 - Los objetos referenciados en una página sufren un RTT entre la petición y la respuesta
 - La conexión está inactiva durante cierto tiempo, consumiendo recursos en el servidor
 - Con pipelining: el cliente envía nuevas peticiones tan pronto encuentra referencias en los objetos previamente recibidos (antes de terminar de recibir respuestas previas).
 - Esto reduce los RTT entre petición y respuesta.



HTTP/1.1 Conexiones persistentes

- Pipelining
 - Acumula en buffers peticiones/respuestas para reducir el número de paquetes.
 - Múltiples peticiones se pueden mandar en el mismo segmento TCP.
 - El orden de las respuestas debe mantener el de las peticiones.





HTTP en CDNs

Mean Download Performance Range for Different Numbers of Images and Protocol Options (Jan. 2001)

Protocol Option	Site	Mean Download Time Range (sec.)			
		6 images	12 images	18 images	54 images
Parallel-1.0	CDN	0.26-0.76	0.40-1.23	0.58-1.53	1.49-3.31
	US Origin	1.63	2.45	3.40	8.42
Serial-1.1	CDN	0.27-0.53	0.42-0.81	0.61-1.13	1.46-2.52
	US Origin	1.06	1.46	1.96	4.87
Pipeline-1.1	CDN	0.26-0.50	0.37-0.67	0.47-0.88	1.09-2.04
	US Origin	<i>Partial Support</i>			

CDNs perform significantly better than origin sites, although reducing the number of images (e.g. due to caching) and using HTTP/1.1 options reduces the performance difference.

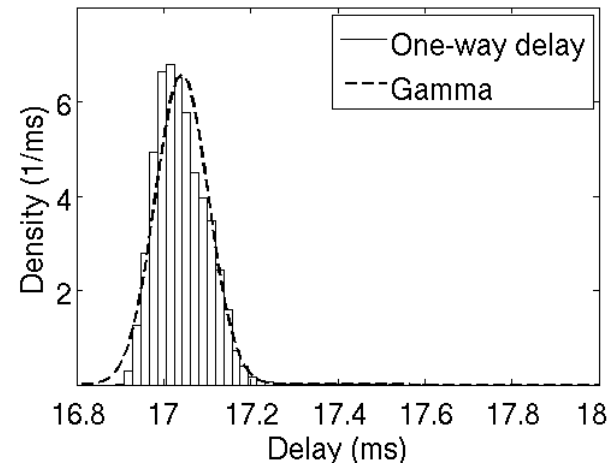


6. Componentes del retardo

- Retardo:
 - RTT (Round Trip Time)
 - Ping.
 - En un sólo sentido (OWD, One-way Delay)
 - $OWD \neq RTT/2$
- El retardo extremo a extremo se puede dividir en 4 componentes:
 - Retardo de transmisión: tiempo necesario para poner todos los bits de un paquete sobre la red.
 - Retardo de propagación: tiempo necesario para propagar un bit por la red de origen a destino.
 - Retardo de procesamiento: tiempo necesario para procesar un paquete en cada router.
 - Retardo en cola: tiempo que un paquete necesita esperar en la cola de un router antes de ser retransmitido.

Componentes del retardo

One-way delay entre
suecia y hungría



- Retardo de transmisión → constante
- Retardo de propagación → constante
- Retardo de procesamiento → constante en arquitecturas actuales, o casi constante.
- Retardo en cola → variable



7. Ejemplo: research in CDNs

- [AoJan06] Ao-Jan Su et al . "Drafting Behind Akamai (Travelocity-Based Detouring)". SIGCOMM'06, September 2006

Motivación:

- CDNs como Akamai hacen esfuerzos para decidir cual de sus servidores sustituto es el más cercano/mejor
 - Aunque sus sistemas de medida son privados
 - Mejoran el tiempo de servicio así que lo hacen bien
- En sistemas peer to peer el decidir si dos hosts están cerca o lejos ayuda a organizar la red overlay (ver siguiente tema...)
 - Es una pregunta interesante
 - Que Akamai está resolviendo para sus propios fines
- ¿Podemos extraer esta información de Akamai y utilizarla para nuestra medida?

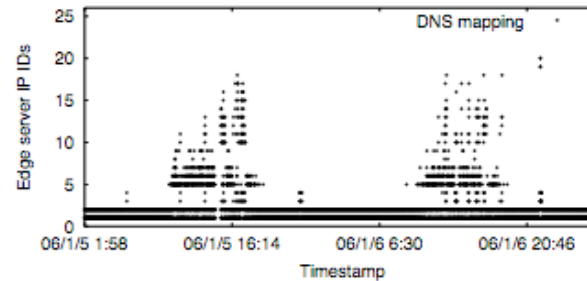


Ejemplo: research in CDNs

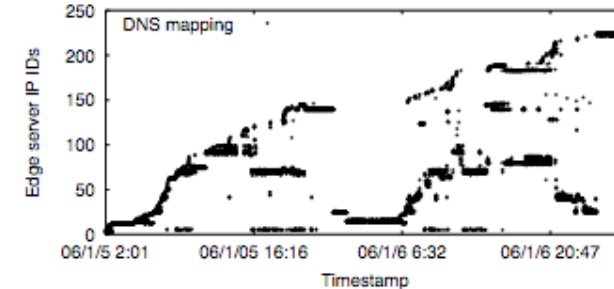
- Propuesta de mapeo de hosts a servidores de Akamai
- A groso modo... si A y B obtienen el mismo servidor de Akamai como más cercano es que están cerca entre sí
- Pero no es tan fácil
 - La CDN no da siempre el mismo servidor cercano
 - Varía con el tiempo
 - Varía con las condiciones de la red
 - Varía con la localización del usuario
 - Varía con el cliente para el que se replica el contenido
- Además el DNS de Akamai me da 2 servidores cercanos a elegir en lugar de 1

Ejemplo: research in CDNs

- Variabilidad del número de servidores cercanos observados
- 140 nodos de PlanetLab piden el servidor para un mismo contenido
 - En un día

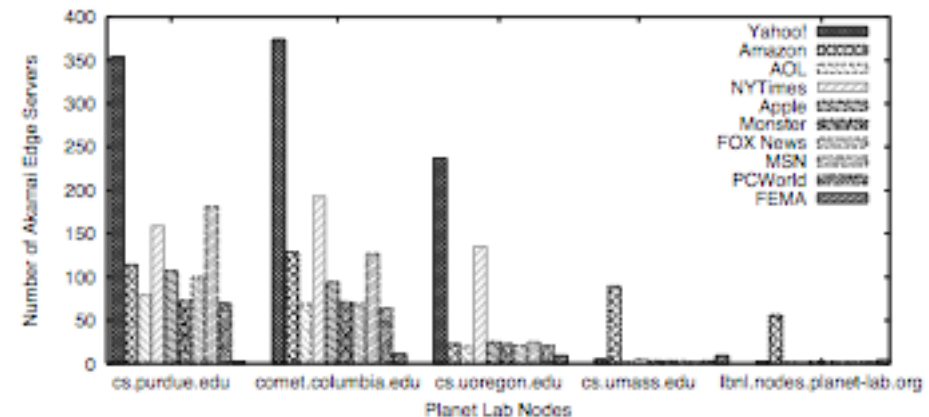


(a) From Berkeley



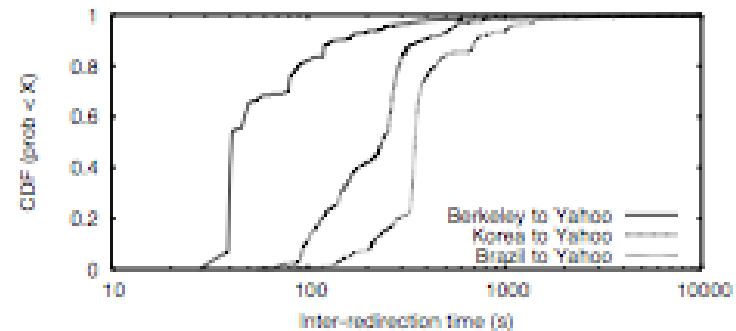
(b) From Purdue

- Pocos servidores significa que estamos cerca de esos pocos servidores
- Muchos servidores significa que al estar lejos según las condiciones de la red se eligen diferentes
- Pidiendo diferentes clientes



Ejemplo: research in CDNs

- Variación con el tiempo.
 - 20 segundos TTL de los LLDNSs
 - Pero como de rápido se evalúa cual es el mejor servidor cercano?
 - Menos de 100 segundos



- Comprueban que el servidor cercano elegido está correlado con la latencia del usuario a dicho servidor
 - Se esta eligiendo servidores cercanos en latencia
 - Esta encontrando caminos en Internet de calidad



Ejemplo: research in CDNs

- Propuesta para redes overlay
 - Akamai-driven on-hop source routing
 - Mapear algunos de los peers de la red overlay a servidores de Akamai
 - Cuando tengo que enviar algo de A a B
 - Elegir servidor cercano de Akamai -> elegir peer C mapeado a ese servidor
 - Enviar A -> C -> B
 - Si Akamai elige bien los servidores cercanos A-C-B puede ser más rápido que A-B
- Evalúan las prestaciones de esta propuesta sin implementarla
 - Nodos de planetlab
 - Pings a servidores más cercanos de Akamai
 - Pings entre parejas de nodos
 - Comparación de latencias

Ejemplo: research in CDNs

- Comparación
Camino directo menos caminos por cada servidor



- En media es mejor por el directo
- Pero si elegimos bien el servidor intermedio más del 50% de los caminos mejorarían



Ejemplo: research in CDNs

- Proponen técnicas para implementarlo
(FAS) First Akamai Server (BTAS) Better of the Two Akamai Servers...
- Reflexiones de [AoJan06]
 - Solo se usa la CDN para las medidas locales no se usa la red de distribución de Akamai
 - Aunque es posible que no siente bien a las CDNs pero no se aumentaría apreciablemente su carga con estas técnicas.
 - Podria llevar a sobrecarga en los caminos que Akamai ve como buenos?
- Al parecer hay quien dice haber implementado la propuesta <http://www.aqualab.cs.northwestern.edu/projects/Ono.html>






Referencias

- [Kurose]
 - Capítulo 7, sección 7.5
- [Peterson]
 - Capítulo 9, sección 9.4
- [Rodriguez01] P. Rodriguez, C. Spanner, E.W.Biersack ."Analysis of Web Caching Architectures: Hierarchical and Distributed Caching". IEEE/ACM Transactions on Networking'01 (TON), August 2001.
- [Krishnamurthy01] Balachander Krishnamurthy, Craig Wills, Yin Zhang. On the Use and Performance of Content Distribution Networks ACM SIGCOMM INTERNET MEASUREMENT WORKSHOP 2001.
- [Johnson00] Kirk L. Johnson, John F. Carr, Mark S. Day, M. Frans Kaashoek. The Measured Performance of Content Distribution Networks. Computer Communications, Volume 24, Number 2, pp. 202-206(5), 1 February 2000.
- [Vakali03] A. Vakali and G. Pallis. Content delivery networks: status and trends. IEEE Internet Computing, Volume 7, Issue 6, pp.68-74, Nov.-Dec. 2003.
- [AoJan06] Ao-Jan Su et al ."Drafting Behind Akamai (Travelocity-Based Detouring)". SIGCOMM'06, September 2006



Artículos a presentar

-  P. Rodriguez and E.W. Biersack. Dynamic parallel access to replicated content in the Internet. IEEE/ACM Transactions on Networking, Volume 10, Issue 4, pp.455–465, Aug. 2002.
-  S. Bakiras. Approximate Server Selection Algorithms in Content Distribution Networks. IEEE International Conference on Communications, ICC 2005, Vol. 3, pp. 1490- 1494, 16-20 May 2005.
-  Ganguly, S. Saxena, A. Bhatnagar, S. Izmailov, R. Banerjee, S. Fast replication in content distribution overlays. INFOCOM 2005, Volume: 4, pp. 2246- 2256, Aug 2005