

Tema 1:

Arquitectura de la WEB:

- La web
- Arquitectura y Componentes
- Arquitectura TCP-IP
- Paradigma Cliente/Servidor
- Proxies

La WEB

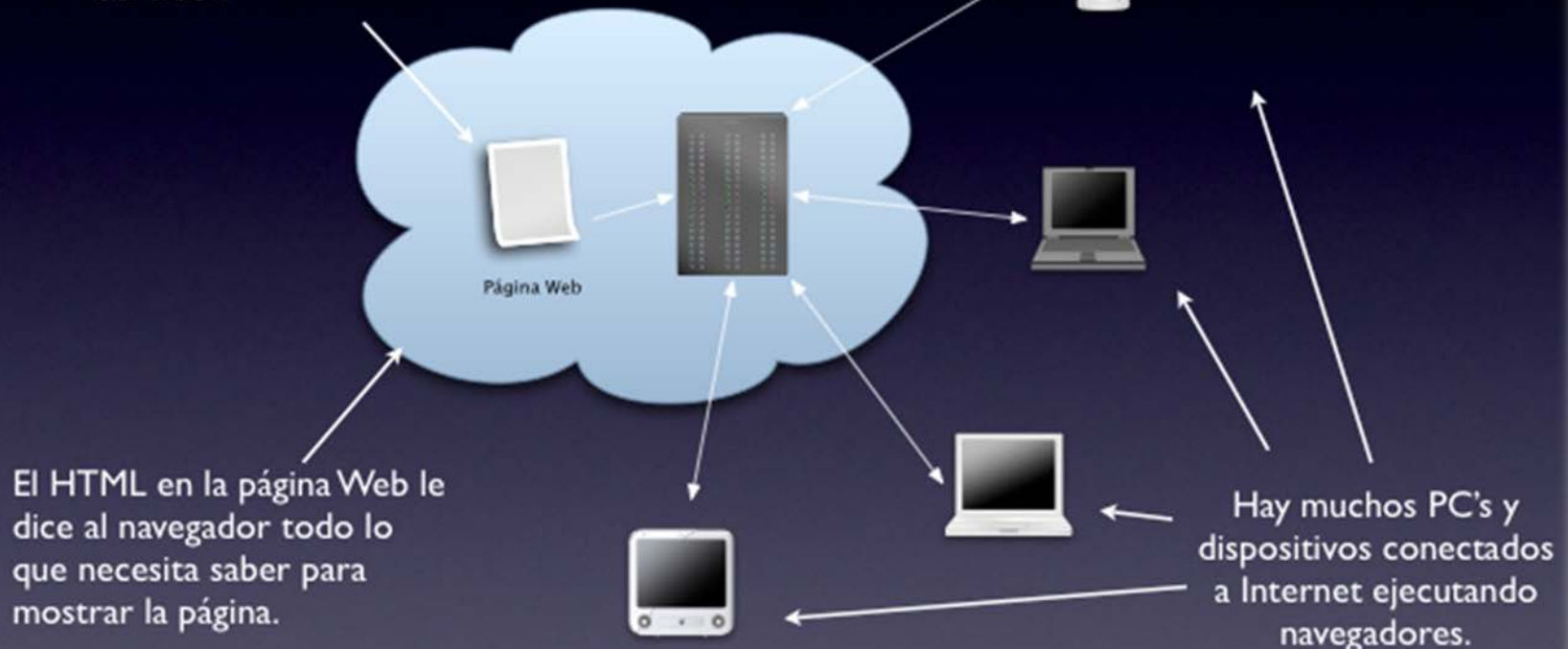
Sistema de distribución de documentos de hipertexto o hipermedios interconectados y accesibles vía Internet



La WEB. Contenidos

Para hacer páginas Web, se crean archivos escritos en HTML y se colocan en un servidor.

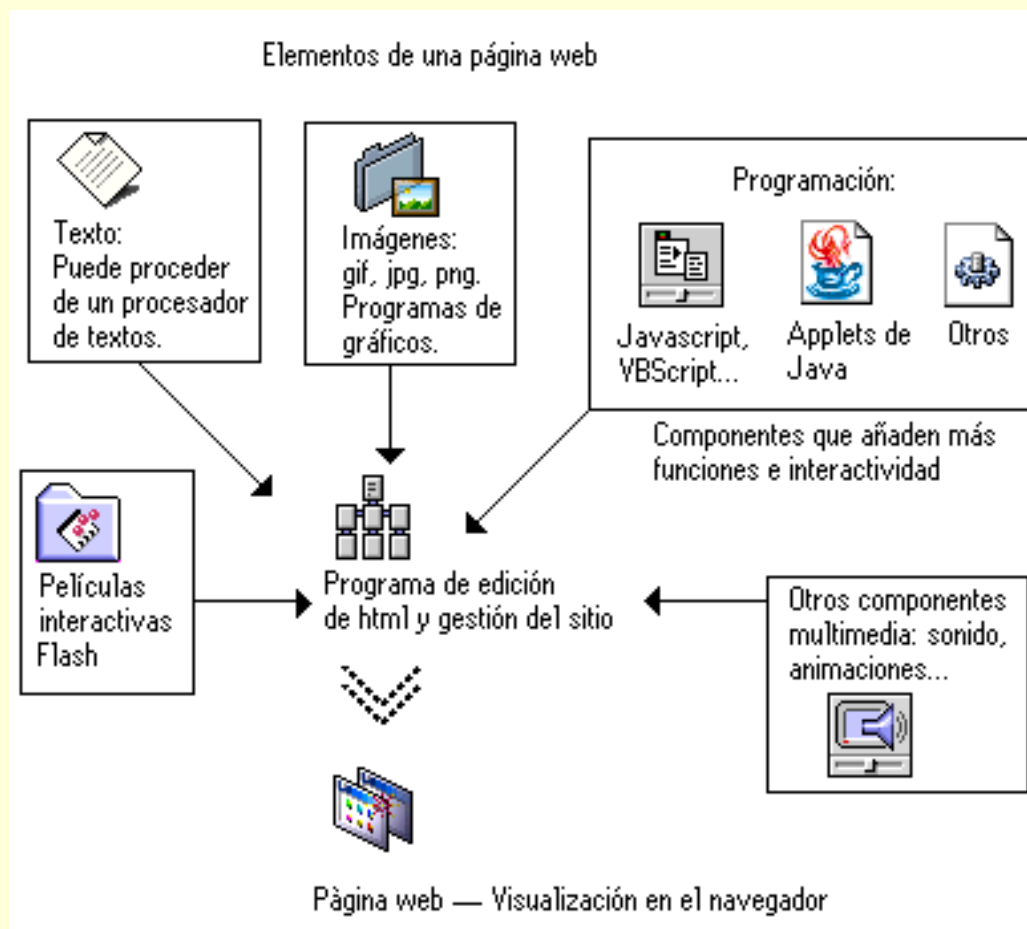
Una vez colocados los archivos en un servidor, cualquier navegador puede acceder a nuestras páginas en Internet



¿qué es una página web?



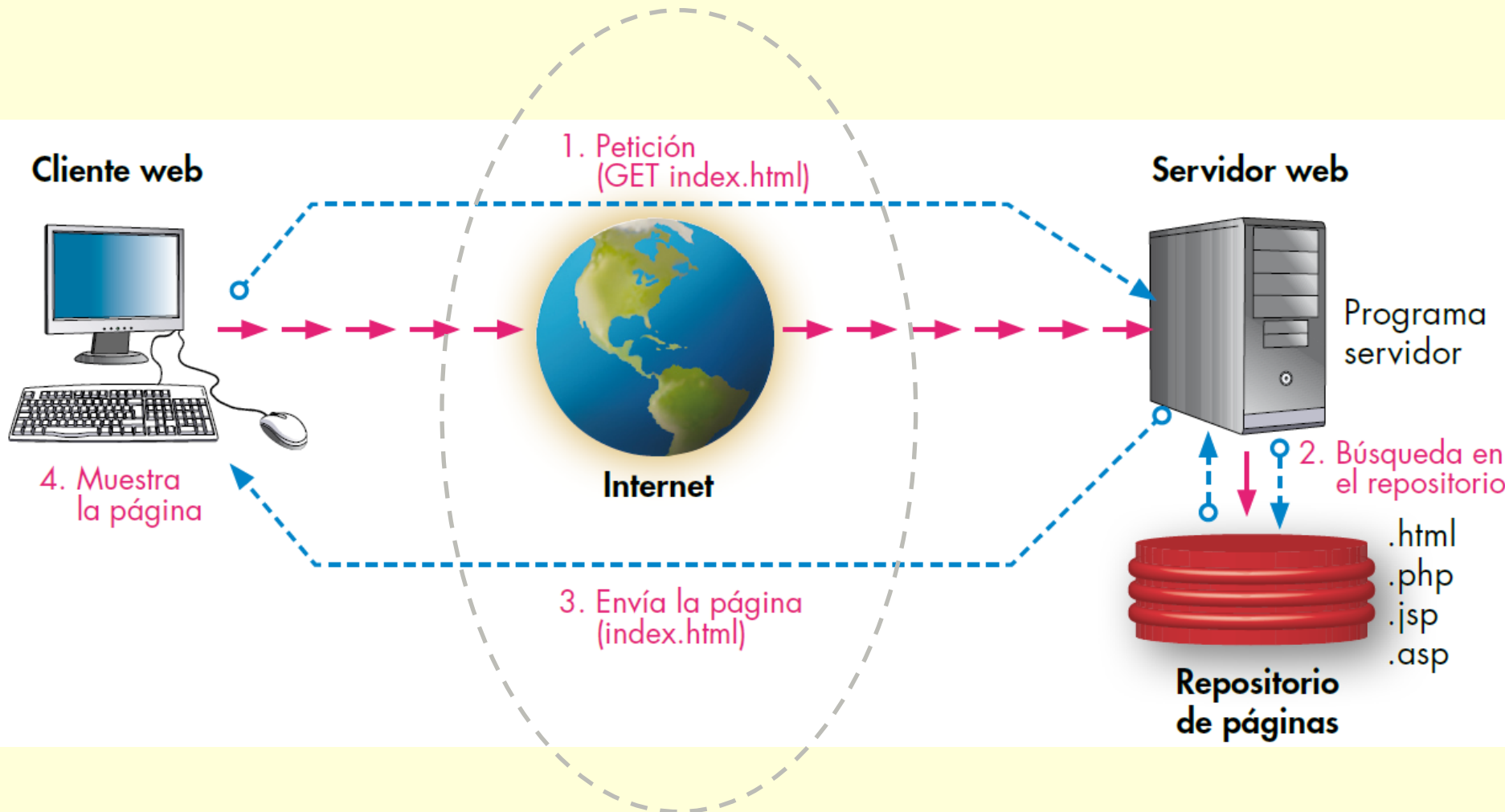
Elementos de una página web



Cliente-Servidor



Descarga de una página web



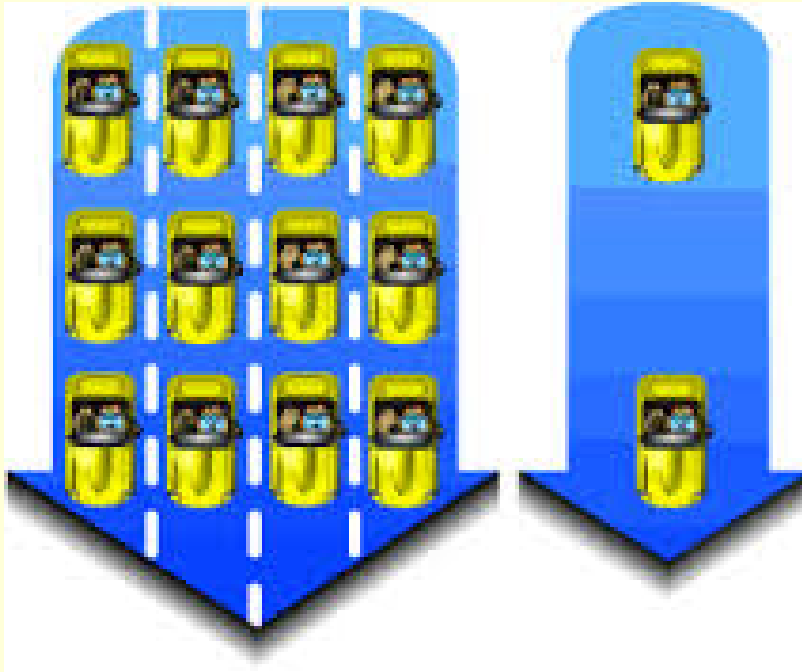
Hay tantas peticiones como elementos en la página

Tiempo de descarga (latencia)



<http://www.testdevelocidad.es/>

Ancho de banda

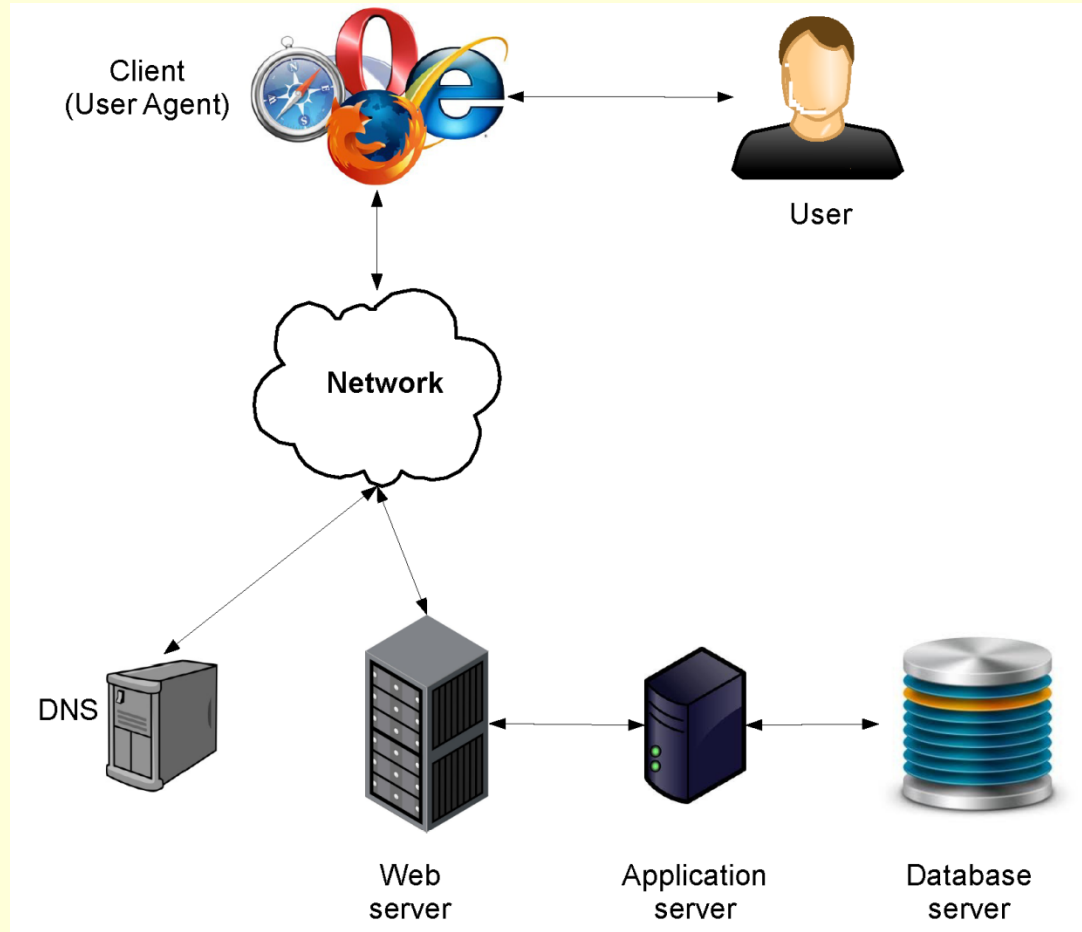


Cantidad de información o de datos que se puede enviar a través de una conexión de red en un período de tiempo dado.

El ancho de banda se indica generalmente en bytes por segundo (Bps), kilobytes por segundo (KBps), o megabytes por segundo (MBps).

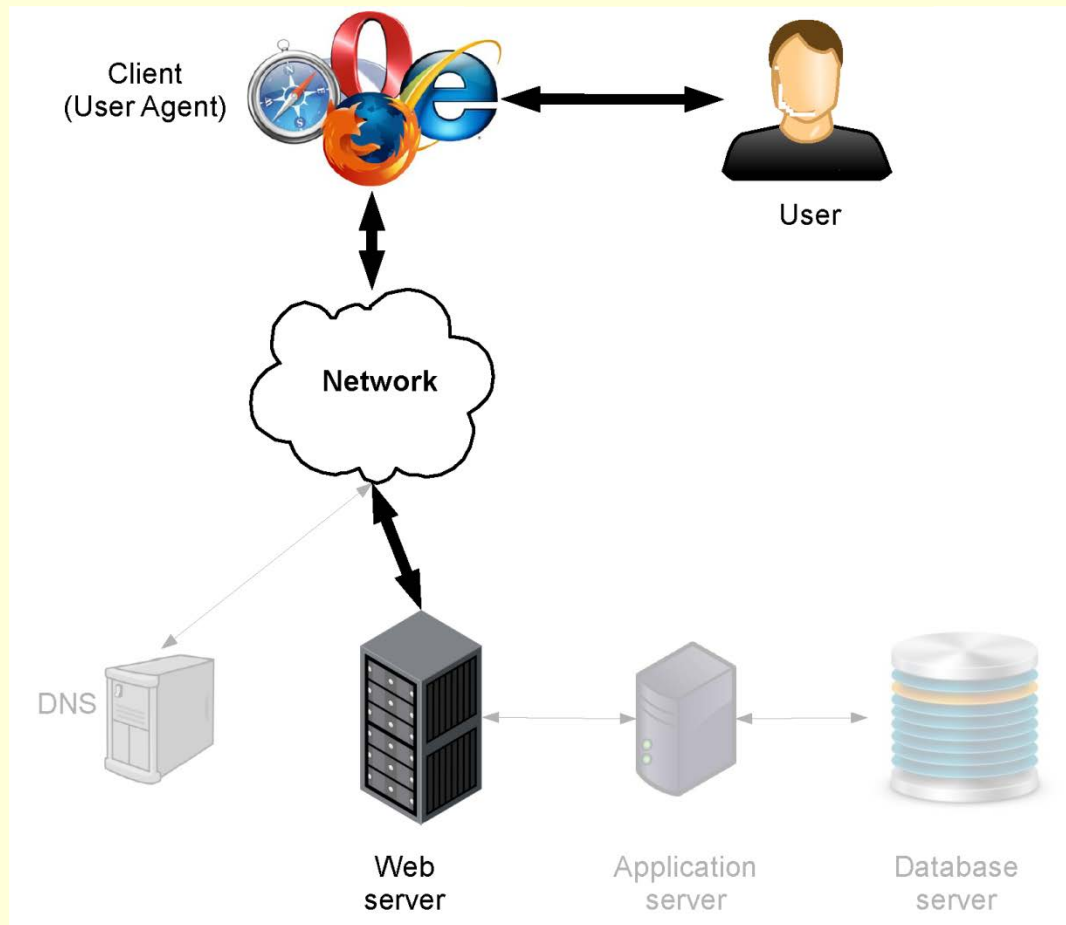
Arquitectura de la WEB

- Elementos de la arquitectura



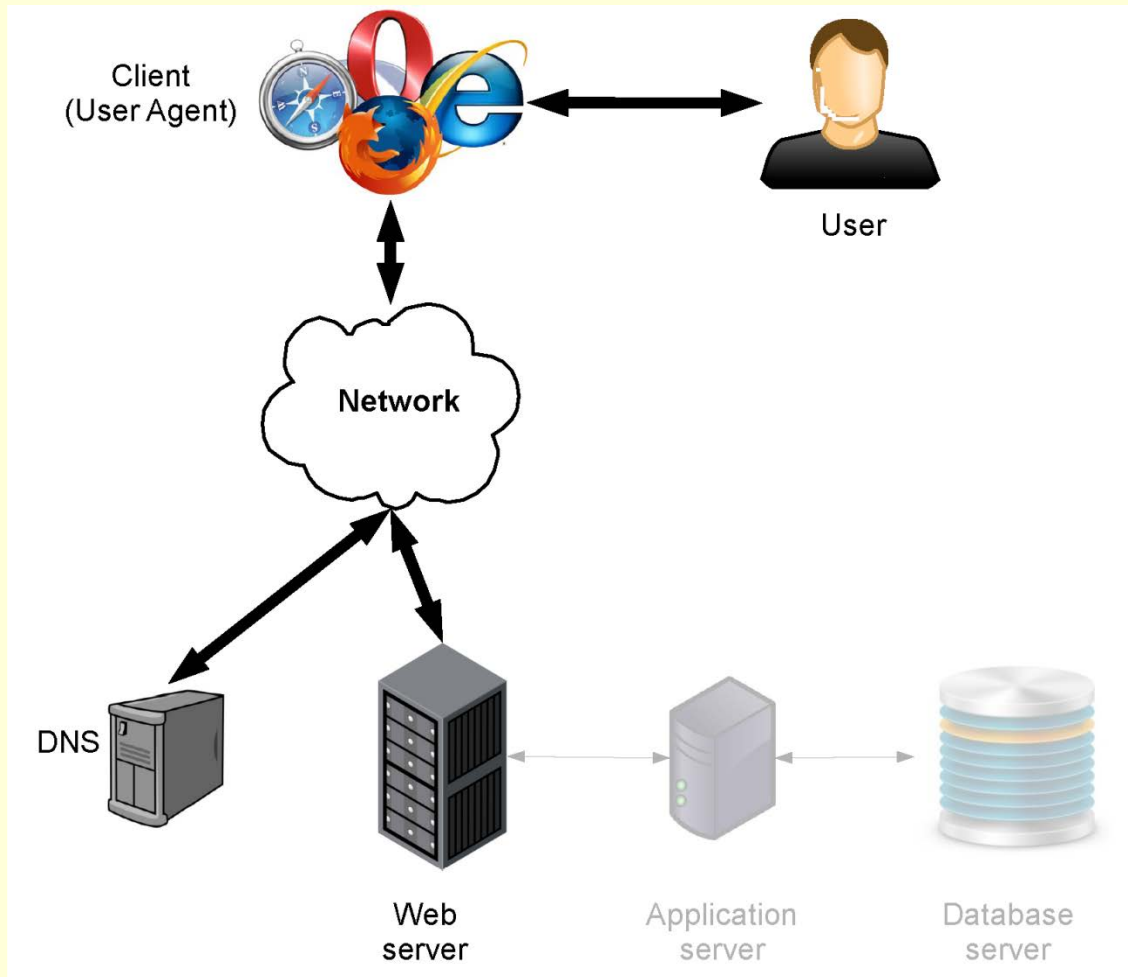
Arquitectura de la WEB

- Ejemplo de navegación utilizando direcciones IP



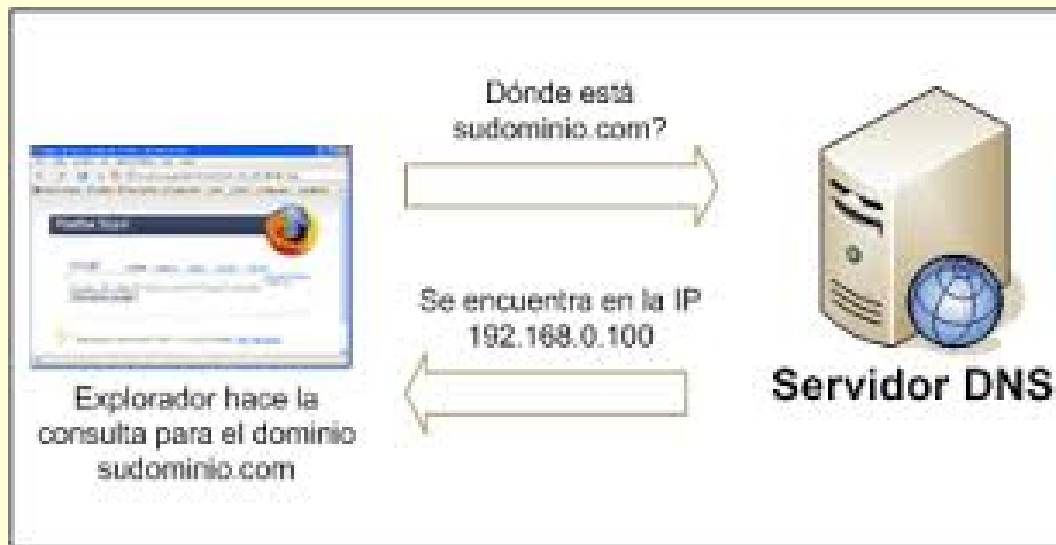
Arquitectura de la WEB

- Ejemplo de navegación a una web estática (direcciones por nombres)



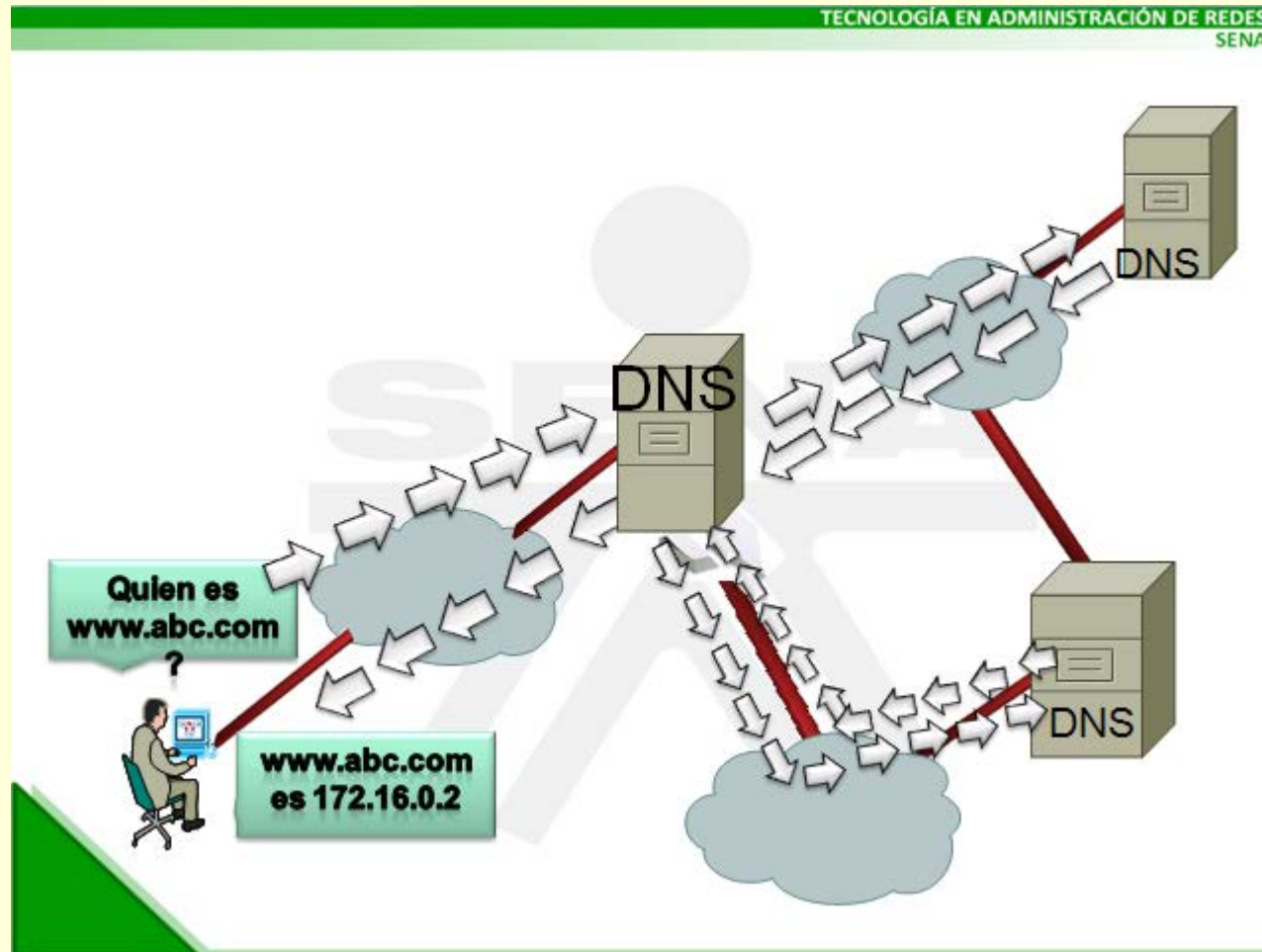
Arquitectura de la WEB

Sistema de nombres de dominio



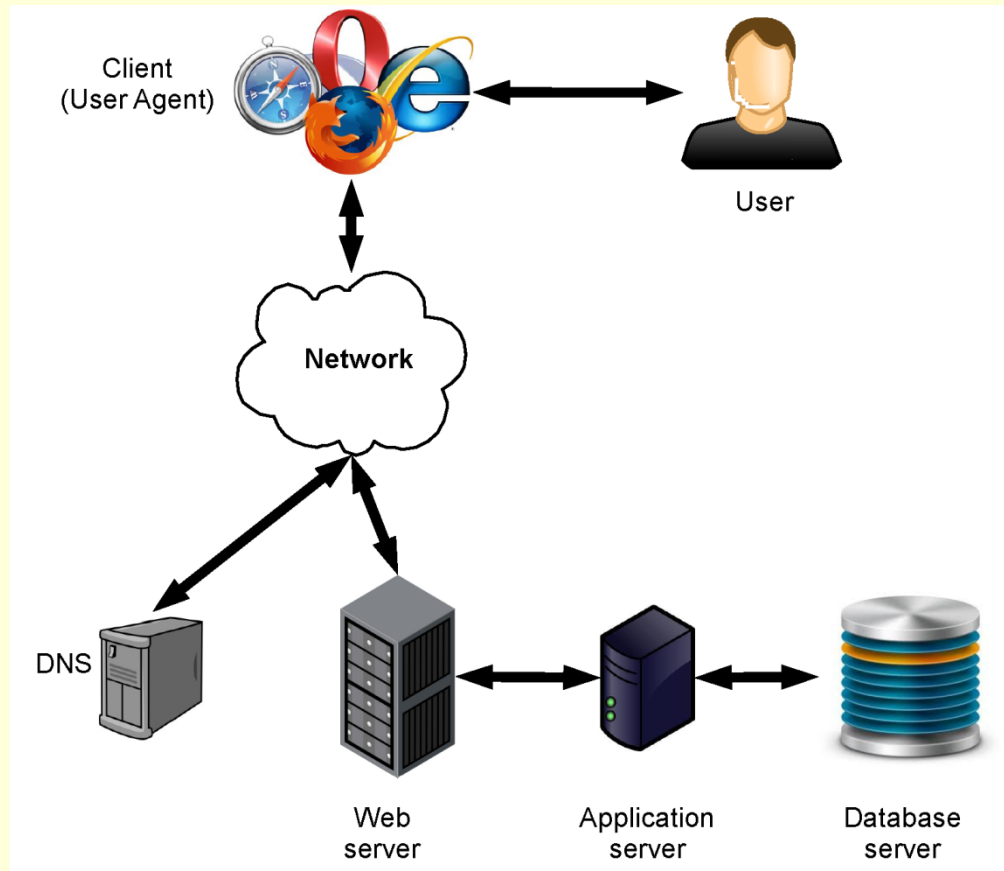
Arquitectura de la WEB

Sistema de nombres de dominio



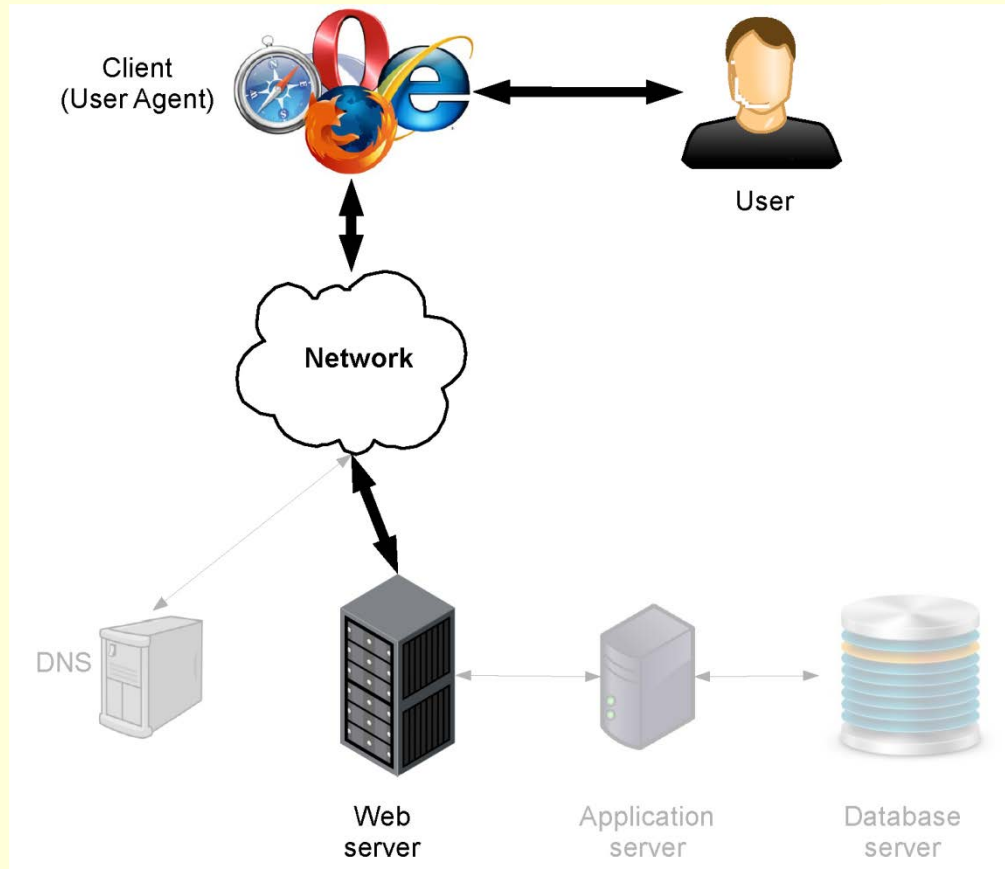
Arquitectura de la WEB

- Ejemplo de navegación a una web social (dinámica) (direcciones por nombres)



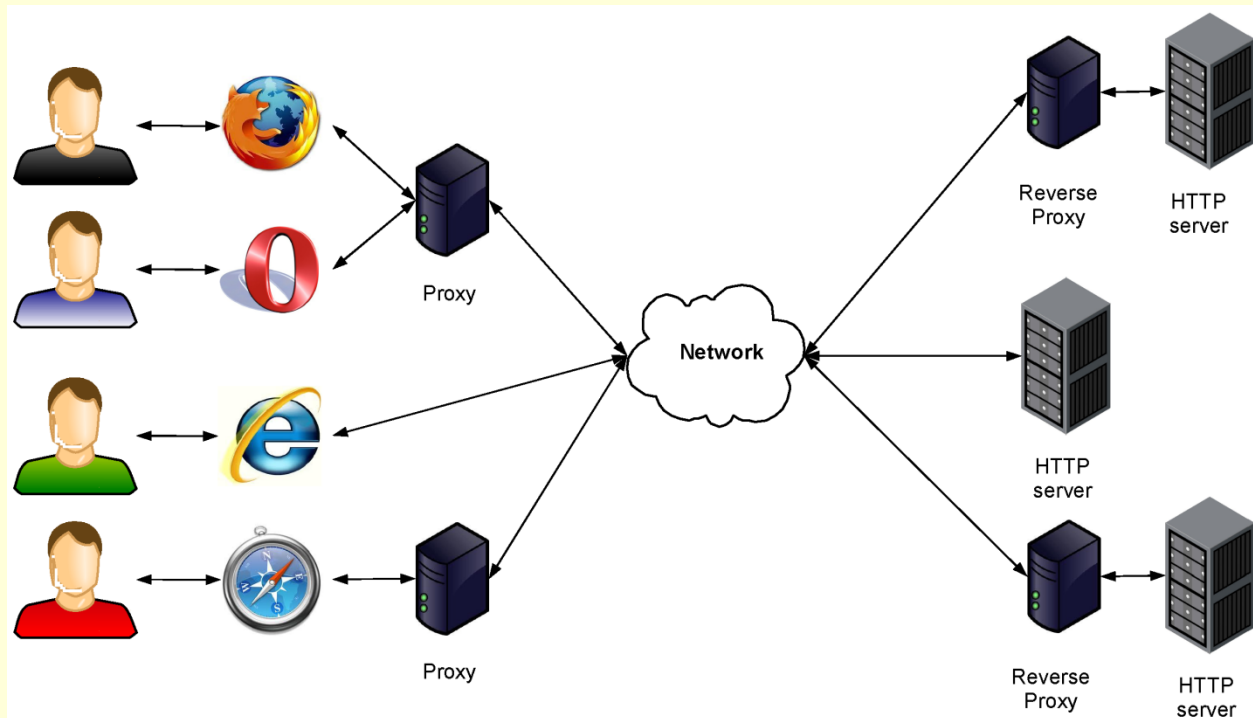
Arquitectura de la WEB

- Nuestro objetivo:
 - Mejorar las prestaciones entre Cliente-Servidor



Evolución de la arquitectura

- Utilización de proxies (intermediarios)



Paradigma Cliente Servidor

- El modelo Cliente/Servidor se formula atendiendo al trabajo que debe realizar una aplicación entre dos procesos (el Cliente y el Servidor)
- En la web todos los computadores se comunican por pares siguiendo el modelo Cliente-Servidor
- Los Servidores proveen recursos o servicios:
 - DNS
 - HTTP
 - Aplicaciones
 - Bases de datos
- Los Clientes requieren estos servicios
- En general Clientes y Servidores residen en distintas máquinas

Los Clientes

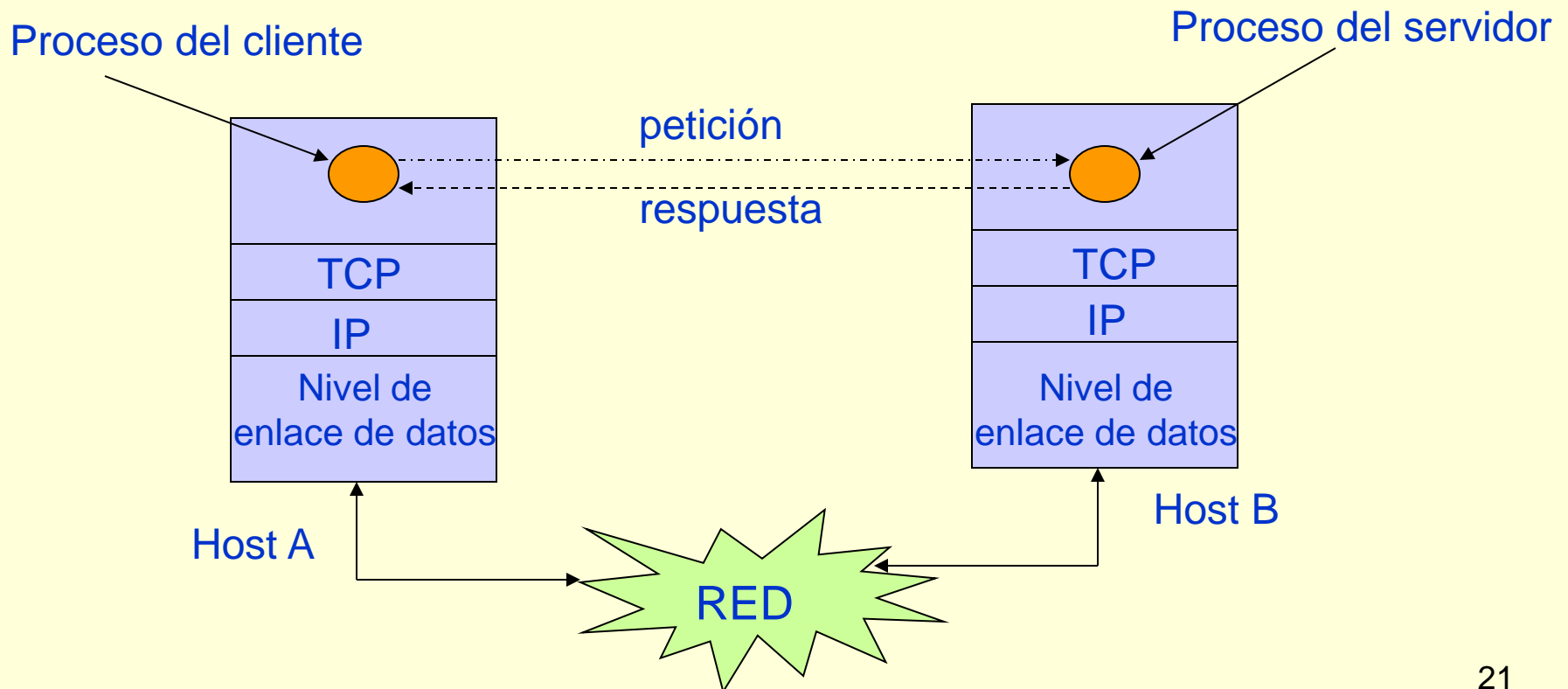
- Interactúan con el usuario
- Transforman las peticiones de los usuarios en peticiones al Servidor (o servidores) siguiendo el protocolo HTTP
- Los requerimientos HW son menores que para el Servidor
- Objetivos en su diseño:
 - Proporcionar un entorno amigable
 - Mejorar la productividad de las navegaciones (botón atrás, apertura de nuevas ventanas o tabs)
 - Reducir la latencia percibida por el usuario (caching y prebúsqueda)
- Ejemplos:
 - Mozilla Firefox
 - Internet Exploree
 - Google Crome
 - Safari

Los Servidores

- Proveen servicio a sus Clientes
 - Utiliza una máquina más potente que el cliente
 - Ejecuta servicios que requieren HW o SW específico
 - Son entidades pasivas: escuchan las peticiones de los clientes, las ejecutan y las contestan. Nunca inicia un mensaje hacia un cliente.
 - Requiere un SO que permita accesos concurrentes. Pueden utilizar:
 - Varios servidores
 - Procesos
 - Threads
- Objetivo en su diseño:
 - Aumentar su productividad (servir a más clientes)
 - Disminuir su tiempo de servicio

Comunicación Cliente-Servidor

- El protocolo implementado entre el Cliente y el Servidor es un protocolo de *petición/respuesta*

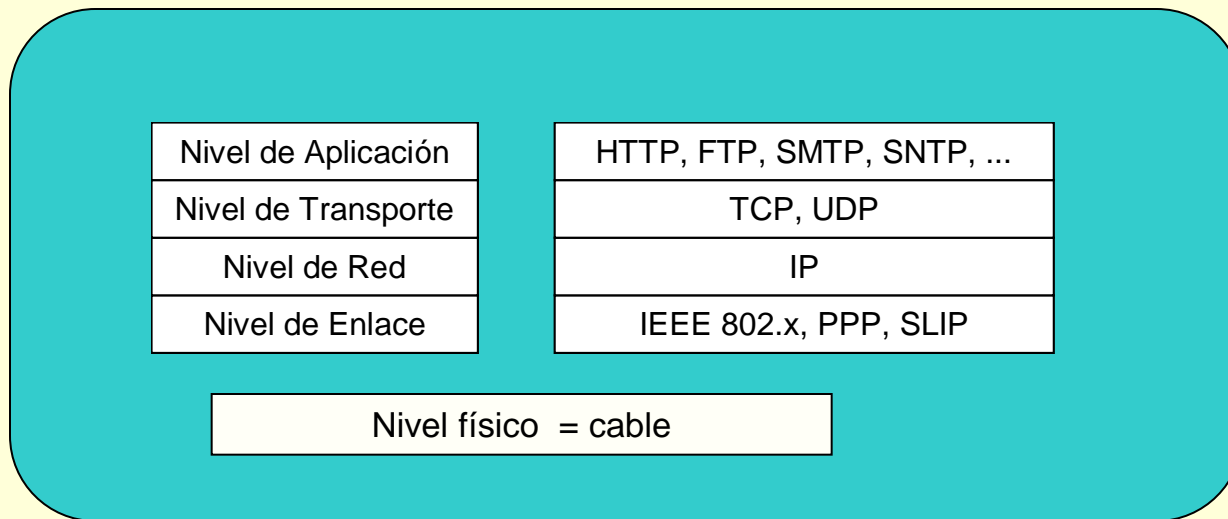


Comunicación Cliente-Servidor

- Se requieren diferentes protocolos:
 - A nivel de transporte:
 - TPC (Transmission Control Protocol)
 - Permite la comunicación de dos nodos en diferentes medios físicos.
 - Incluye control de errores, flujo y congestión
 - A nivel de aplicación:
 - HTTP (HyperText Transfer Protocol)
 - Permite la comunicación de dos procesos en nodos diferentes
 - Sigue el modelo TPC/IP

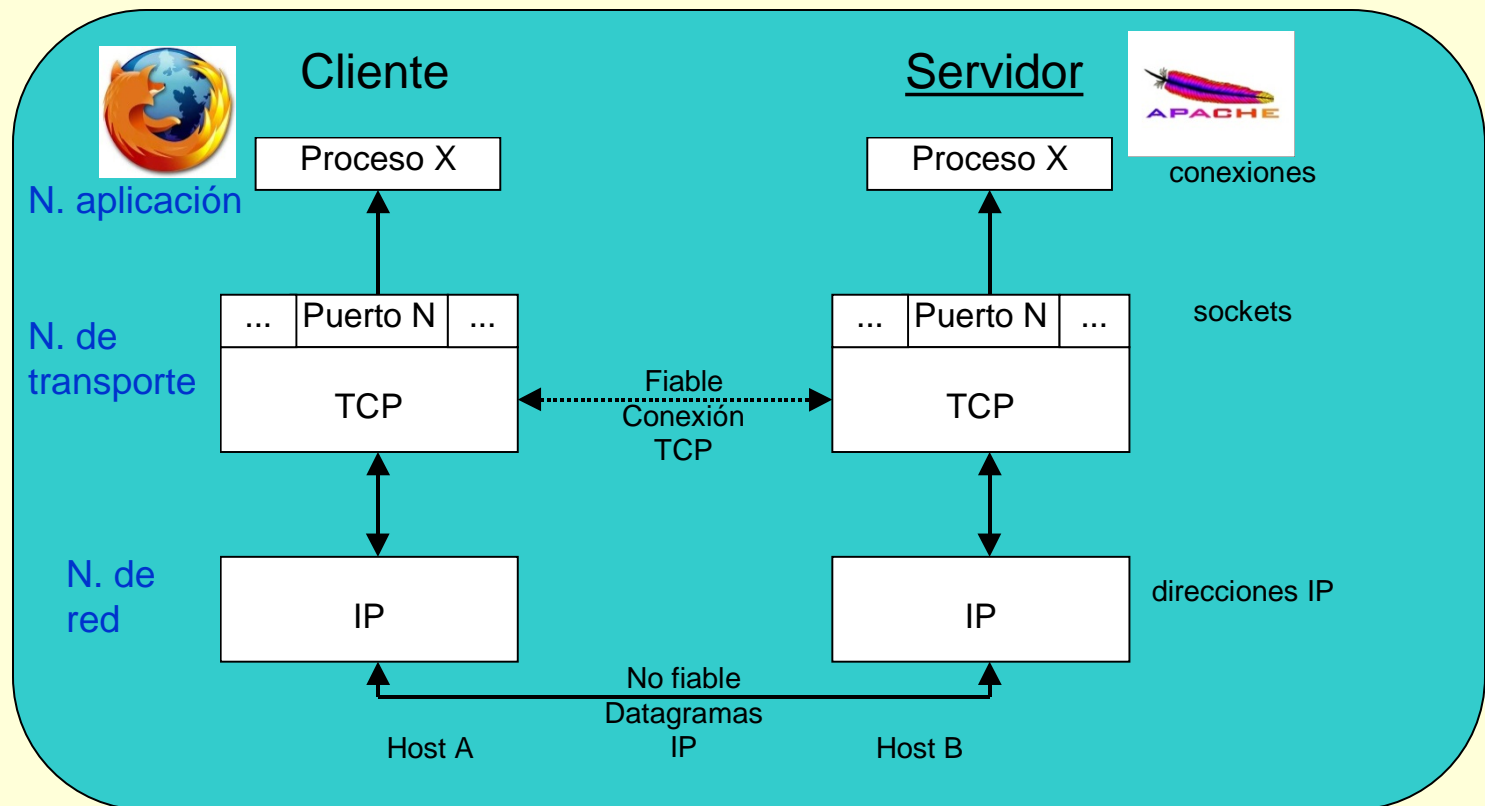
Arquitectura TCP/IP

- La familia de protocolos TCP/IP se estructura en cuatro niveles o capas:



Arquitectura TCP/IP

- La arquitectura de la Web se establece por encima de la infraestructura de la red Internet, que utiliza la pila de protocolos TCP/IP para la gestión de la comunicación.



Arquitectura TCP/IP

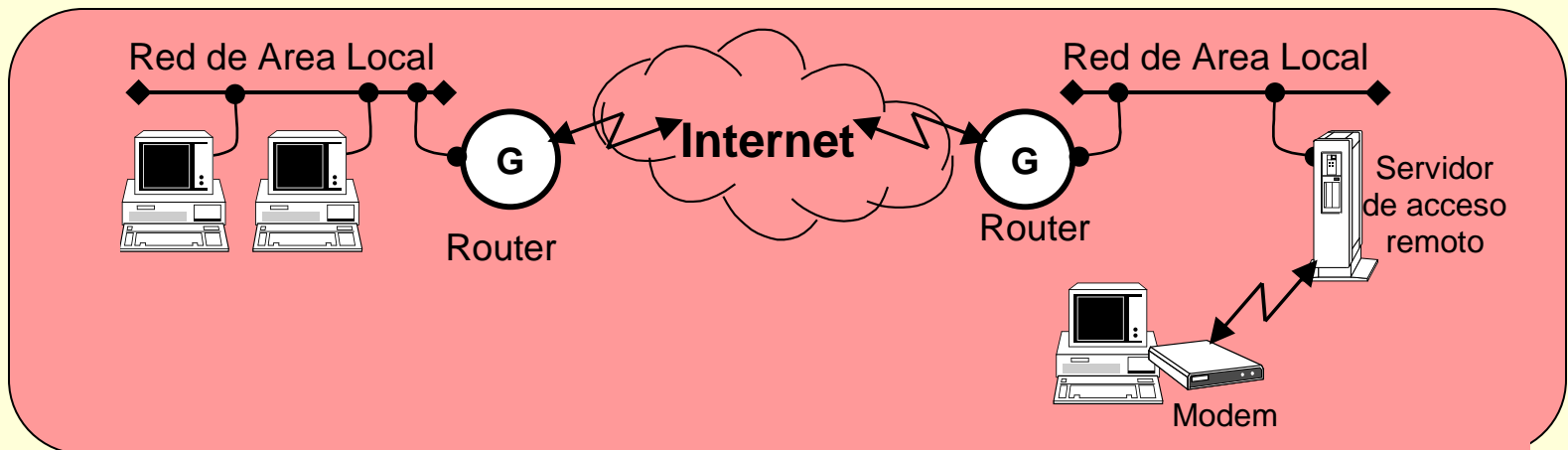
LOS PAQUETES

Una característica importante de este nivel es la unidad máxima de transferencia (MTU Maximum Transmission Unit) que es el tamaño máximo de datos (en bytes) que se pueden enviar en una trama de datos (1492 bytes en la Ethernet). Debido a estas restricciones los paquetes deberán ser fragmentados en origen y re-ensamblados en destino. A cada paquete hay que añadirle una cabecera con información de control para permitir el re-ensamblado del mensaje



Arquitectura TCP/IP

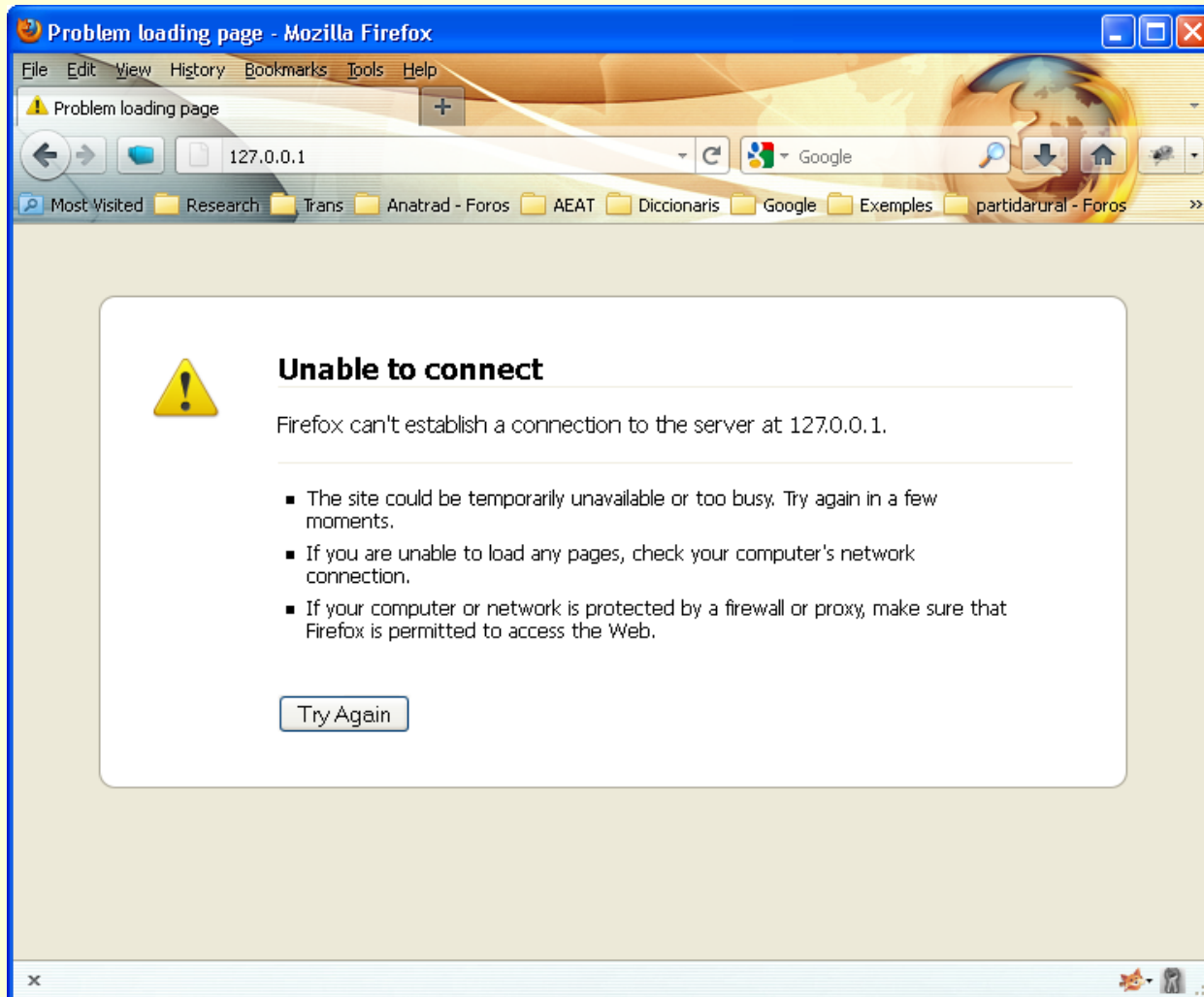
- La red Internet es una Red de Redes (el camino entre el host origen y el destino puede incluir varias redes diferentes)
- El protocolo IP se encarga de encaminar los paquetes.
- En este nivel se definen las direcciones IP.



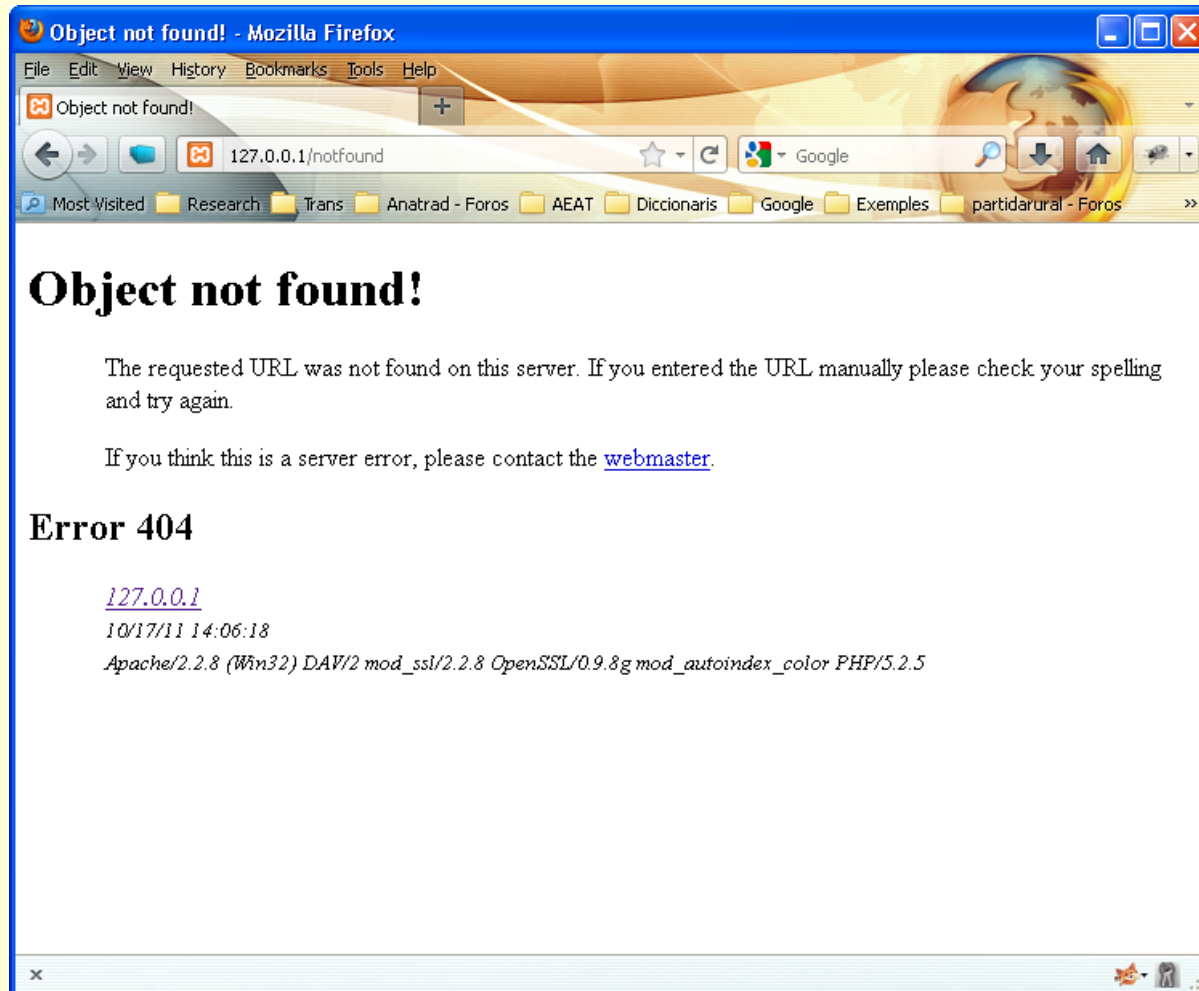
Errores (nivel Internet)



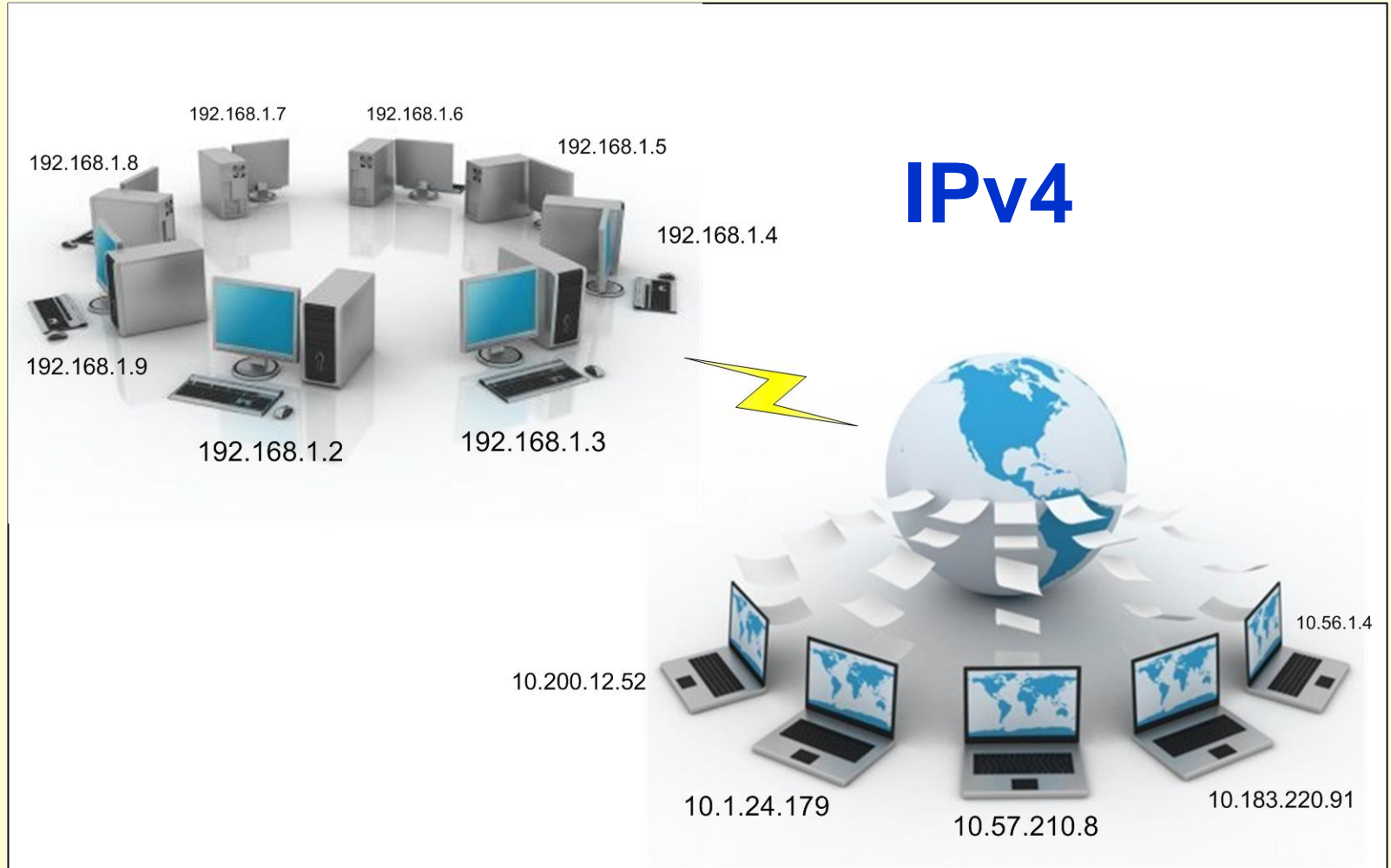
Errores (nivel transporte)



Errores (nivel aplicación)



Direcciones IP



Direcciones IP

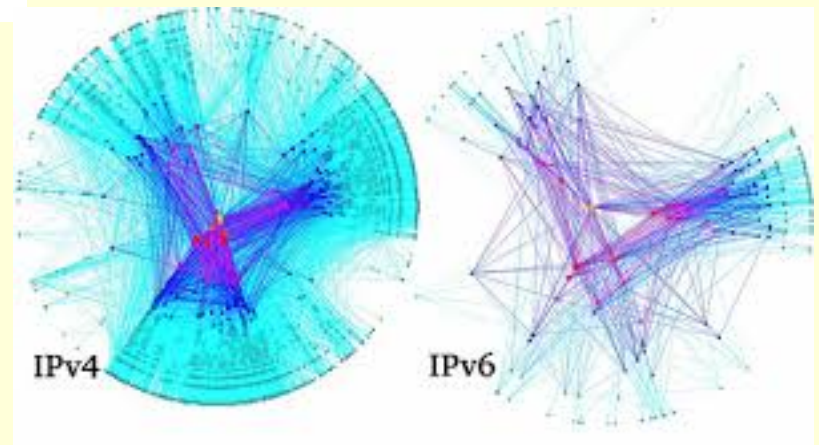
IPv4 Address - 32 bits

208.93.105.218

IPv6 Address - 128 bits

2610:18:cc0:8:0000:0000:1:8010

* Address example



Direcciones IP

- Las direcciones IPv4 están constituidas por 32 bits agrupados en cuatro octetos (IPv6 utiliza direcciones de 128 bits).
- Identifica adaptadores de red. Cada conexión a una red IP debe tener una dirección IP única.
 - Un router tiene al menos dos direcciones IP
 - Un ordenador portátil puede tener una dirección IP para conexión wifi y otra para Ethernet.
- Las direcciones IP constan de dos partes:
 - El número de red
 - El número de máquina (host).
- Todas las máquinas de una misma red local comparten el mismo número de red
- No pueden haber dos máquinas con el mismo número de host en la misma red local.

Direcciones IP

- Existen varias clases de direcciones.
 - Las direcciones de la clase A utilizan sólo el primer octeto para identificar la red
 - Las direcciones de la clase B utilizan los dos primeros octetos para el número de red
 - Los rangos de la clase C utilizan los tres primeros octetos para el número de red

											1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	3	3
	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
Clase A	0	Red								Host																						
Clase B	1	0	Red													Host																
Clase C	1	1	0	Red																		Host										
Clase D	1	1	1	0	Dirección Multicast																											

Direcciones IP

Clase de dirección IP	Valores de octetos				Máscara de subred
	1º	2º	3º	4º	
A	Nº Red 1 ~ 126	Nº Host 0 ~ 255	Nº Host 0 ~ 255	Nº Host 1 ~ 254	255.0.0.0
B	Nº Red 128 ~ 191	Nº Red 0 ~ 255	Nº Host 0 ~ 255	Nº Host 1 ~ 254	255.255.0.0
C	Nº Red 192 ~ 223	Nº Red 0 ~ 255	Nº Red 0 ~ 255	Nº Host 1 ~ 254	255.255.255.0
D Multicast	224 ~ 239	X	X	X	-
E Experimental	240 ~ 254	X	X	X	-

											1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	3	3		
	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
Clase A	0	Red								Host																						
Clase B	1	0	Red												Host																	
Clase C	1	1	0	Red																		Host										
Clase D	1	1	1	0	Dirección Multicast																											

Direcciones IP

- Hay además dos casos especiales:
 - Si el número de host es igual a 0 (158.42.0.0 por Ej.), se está refiriendo a la subred en cuestión. Operación entre redes.
 - Si el número de host es todo 1's se trata de una dirección de difusión (broadcast) a todos los hosts de la subred
- Existen además tres rangos de direcciones privadas que se pueden utilizar dentro de las redes internas de las organizaciones

Clase	Inicio del rango	Fin del rango	Máscara de subred
A	10.0.0.0	10.255.255.255	255.0.0.0
B	172.16.0.0	172.31.255.255	255.255.0.0
C	192.168.0.0	192.168.255.255	255.255.255.0

Direcciones IPv6

- Aparece en los 90 como una ampliación de IPv4
- Permite ofrecer nuevos servicios y crear nuevos tipos de aplicaciones.
- Requiere actualizar el software de los distintos componentes de la web (clientes, ISP, routers, servidores, DNS,...).

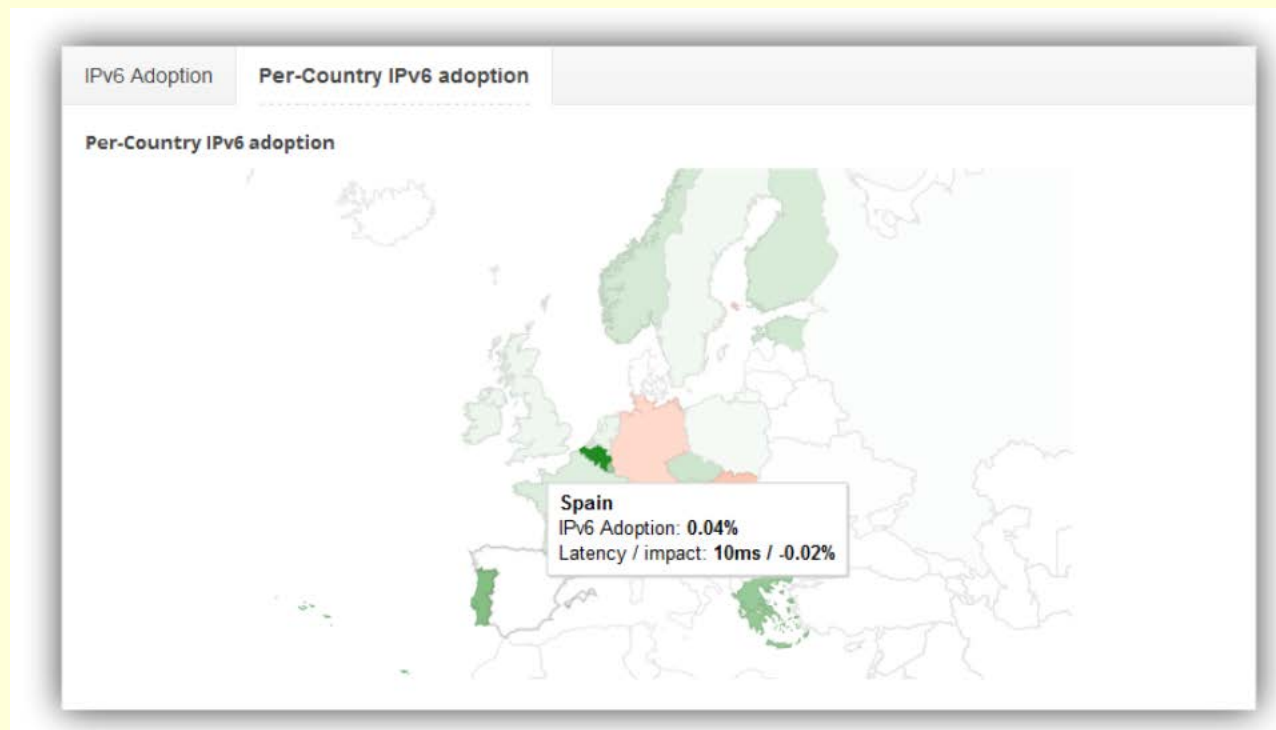
PROBLEMA

- Pasos hacia la implantacion de IPv6
 - Actualización de infraestructuras
 - Diseño de hojas de ruta para la progresiva adaptación
 - Compromisos adquiridos por proveedores de infraestructura y servicios (ATT, Cisco, Akamai, Google, Facebook, Microsoft Bing, Yahoo, etc.)
<http://www.worldipv6launch.org>



Fuente: <http://www.redeszone.net/2016/01/04/conoce-el-uso-de-ipv6-en-el-mundo-en-enero-de-2016/>

El uso de IPv6 en España supone un aumento de latencia medio de 10ms



Asignación de direcciones

- En el caso de las direcciones IP, la autoridad máxima es la IANA (Internet Assigned Number Authority <http://www.iana.org/>).
- A escala continental, la IANA delega grandes bloques de direcciones IP a los denominados registros regionales.
- También es la autoridad máxima del espacio de nombres de Internet (DNS).
- Las organizaciones y usuarios finales han de obtener las direcciones IP necesarias para conectarse a Internet a través de su proveedor de acceso a Internet, quien a su vez las habrá obtenido bien de su proveedor de tránsito, bien del registro regional correspondiente.

Asignación de direcciones

- La asignación de direcciones puede realizarse estática o dinámicamente.
 - La asignación de **estática** consiste en asignar una dirección fija a cada nodo.
 - La asignación **dinámica** se realiza en el arranque del nodo mediante la consulta a un servidor de direcciones, de manera que el nodo puede recibir una dirección distinta cada vez que arranca

Asignación de direcciones

- Existen varios protocolos de asignación de direcciones:
 - BOOTP (Boot Protocol: Unix)
 - WINS (Windows Name Service: Windows)
 - DHCP (Dinamic Host Configuration Protocol: Windows NT, Unix, NetWare).
- La información que un Cliente necesita para conectarse a Internet es:
 - IP _____ ■ _____ ■ _____ ■ _____
 - MSK _____ ■ _____ ■ _____ ■ _____
 - GTW _____ ■ _____ ■ _____ ■ _____
 - DNS1 _____ ■ _____ ■ _____ ■ _____
 - DNS2 _____ ■ _____ ■ _____ ■ _____

El servicio de nombres

- El servicio de nombres se implementa para ocultar al usuario las direcciones IP, ya que éstas constituyen un sistema poco amigable para el direccionamiento de servidores y nodos.
- El sistema de direccionamiento por nombres es un sistema jerárquico por dominios.
 - Ejemplo: **hermes.disca.upv.es**
- La autoridad máxima del espacio de nombres de Internet (DNS) es la IANA (Internet Assigned Number Authority).
- La raíz del DNS es gestionada por el InterNIC (<http://www.internic.net/>) por delegación de la IANA.

El servicio de nombres

- Bajo la raíz se encuentran los distintos dominios de primer nivel (Top Level Domains o TLD's) gestionados por distintos registros delegados de Internet. Algunos de ellos son: COM, NET, ORG, EDU, GOV, MIL e INT.
- Además de los dominios de primer nivel uno por cada país administrados por cada uno de los países.
- Diferentes nombres pueden apuntar a la misma dirección IP (servidores virtuales)
- Ventaja adicional: permite el cambio de dirección IP en un host de forma transparente al usuario

Uniform Resource Identifier (URI)

- Es el identificador asociado a la una unidad elemental (básica) de la web = recurso. Ejemplos:
 - Un documento HTML
 - Una imagen
 - Un archivo pdf
- Un recurso puede tener varias URI, pero una URI identifica un único recurso
- No es dependiente del contexto
- Ejemplos:
 - `http://www.myserver.com/some/path/file.html#part1`
URI identifica \part1" del recurso

`mailto:john.doe@server.com?subject=Hello`

El buzón de correo es `john.doe@server.com`

El asunto del mensaje es "Hello"

Uniform Resource Locator (URL)

- Las URL son un subtipo de URI
- La URL muestra como puede localizarse el recurso.

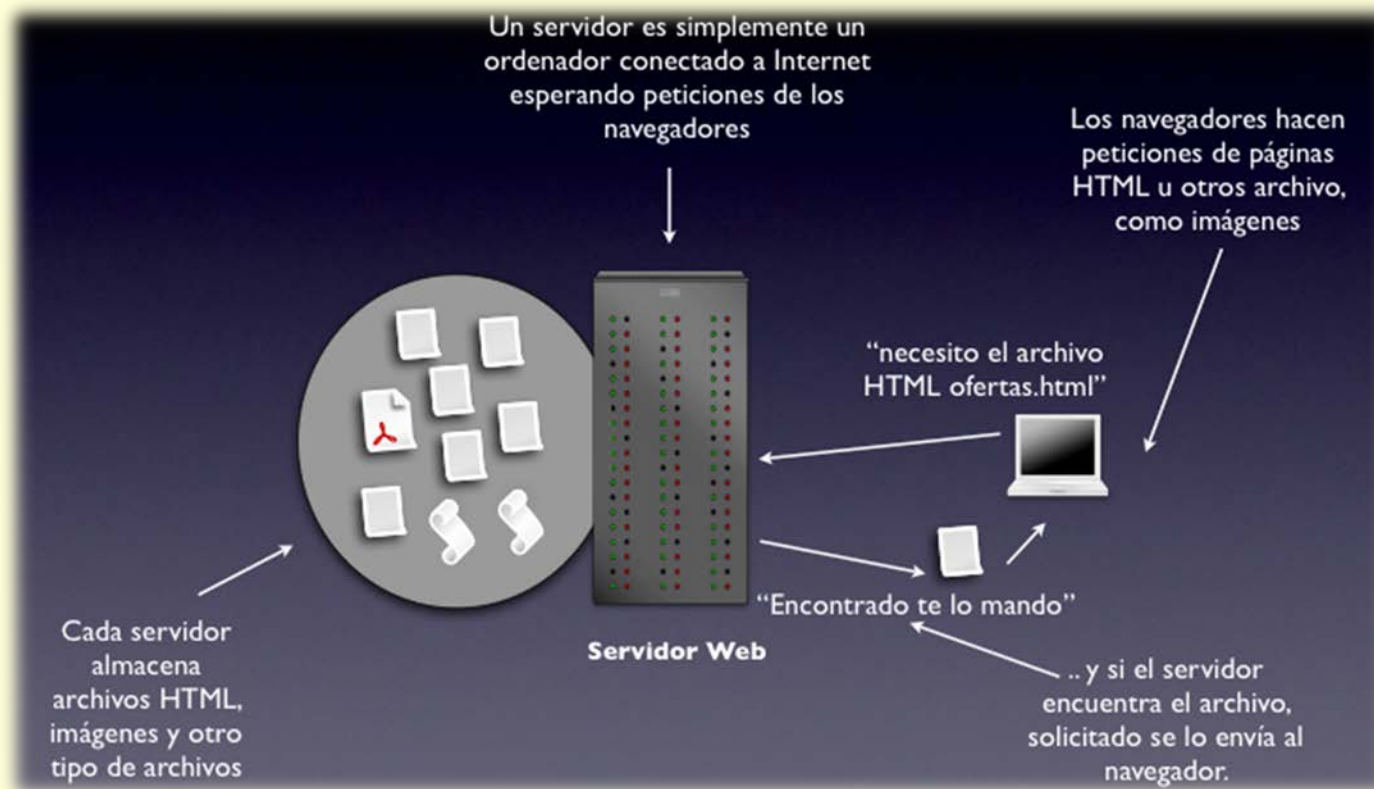
```
http://www.myserver.com/path/file.html
```

Indica:

- Conectar con el host mediante HTTP `www.myserver.com`
 - Solicitar el recurso `/path/file.html`
- Uniform Resource Name (URN) es otro tipo de URI. NO da información sobre cómo localizar el recurso.

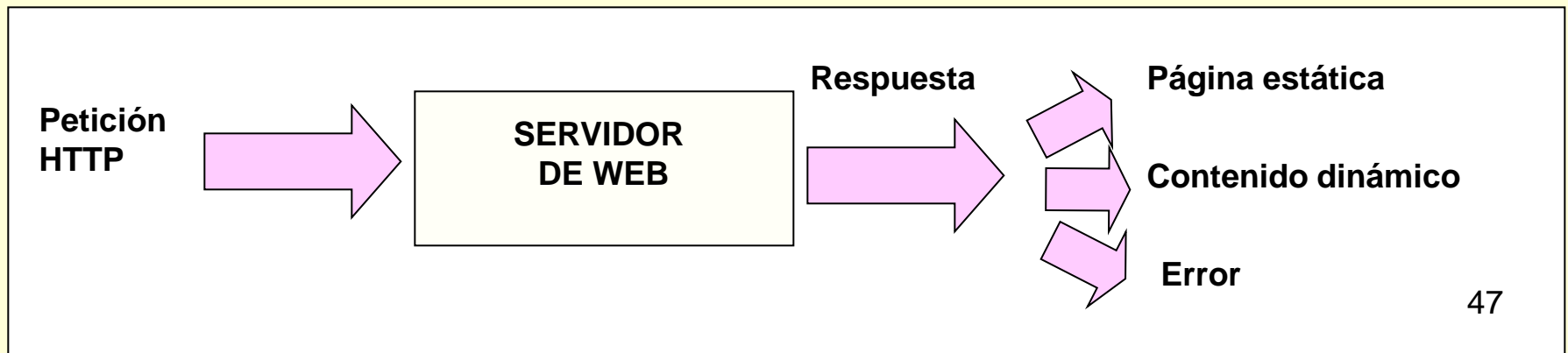
i `urn:isbn:978-0135094051`

Servidores web



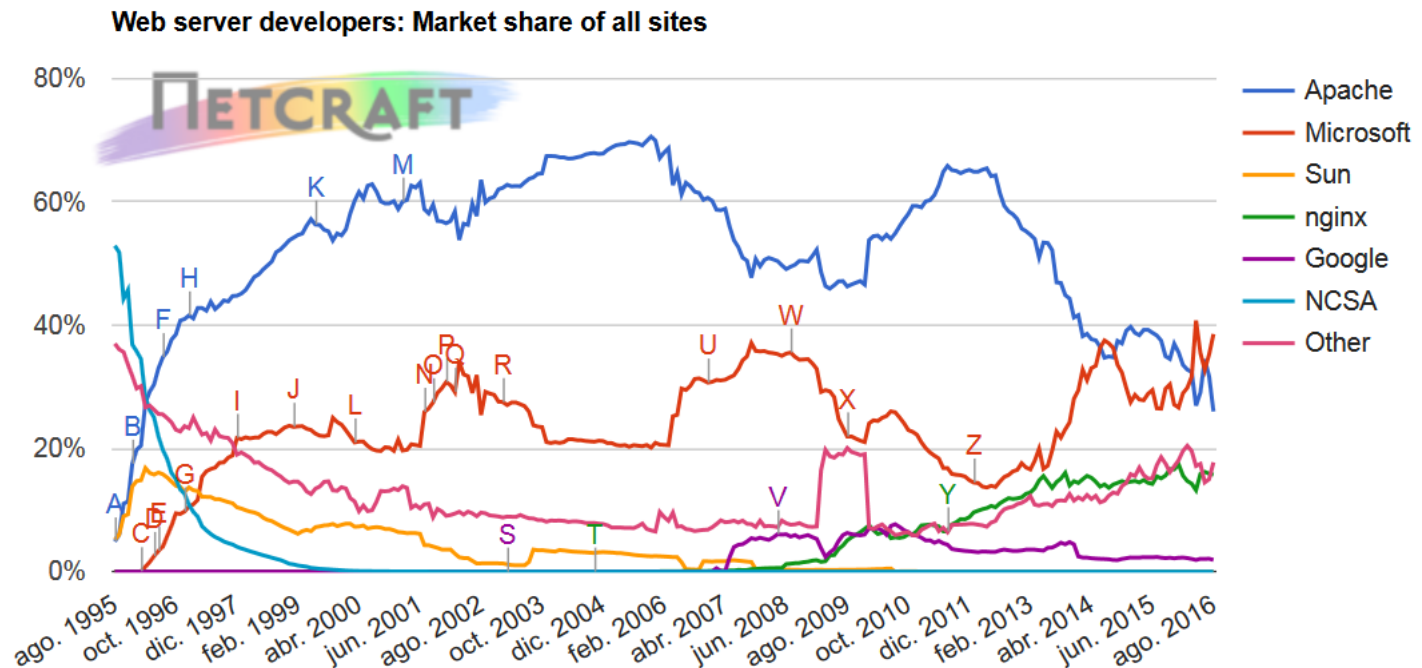
Servidores web

- A veces los recursos se generan según las peticiones de los Clientes
- Se dice entonces que el Servidor genera respuestas DINAMICAS basadas en la petición del cliente y otra información adicional (Bases de datos)
- Las respuestas dinámicas pueden dar lugar a archivos HTML dinámicos o de otro tipo
- Tecnologías usadas:
CGI, Perl, PHP, ASP, JSP, etc.



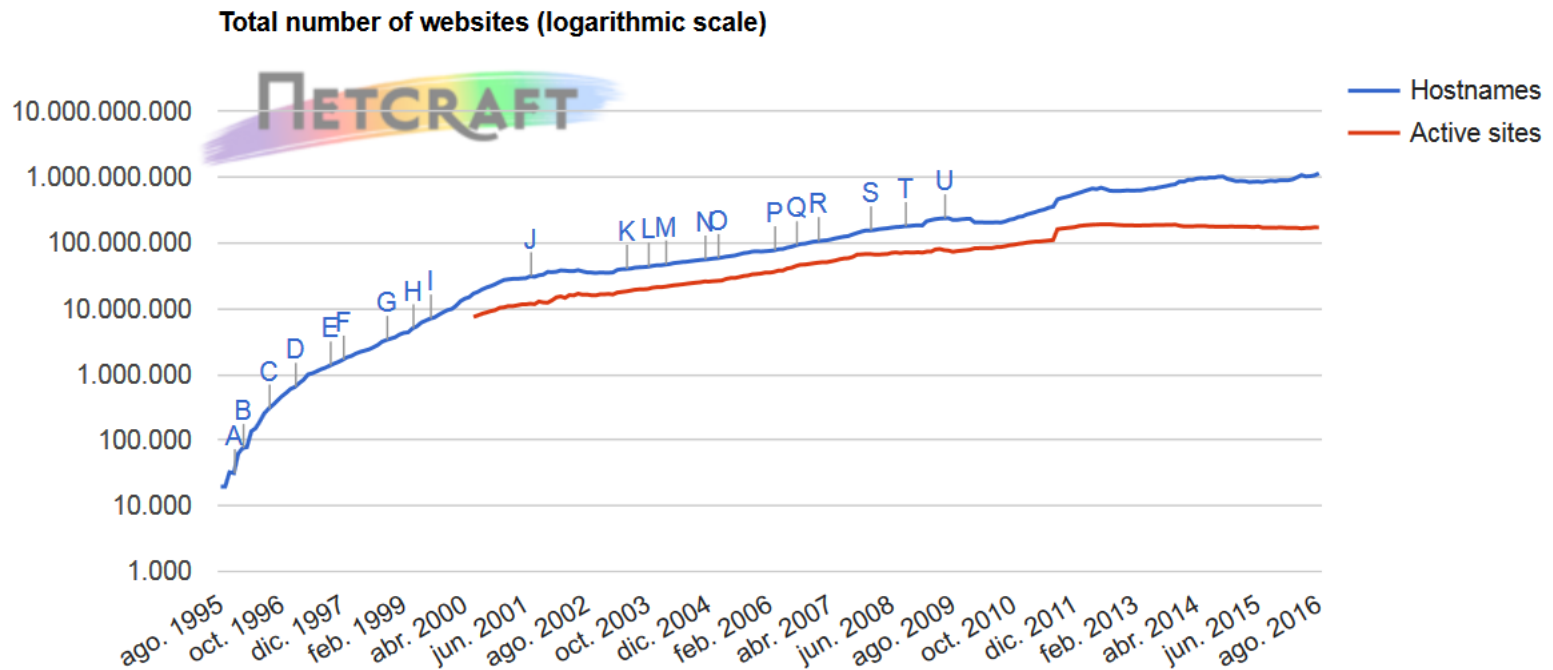
Evolución de los servidores

Presencia de mercado desde agosto de 1995 hasta agosto de 2016



Servidores en Internet

Sitios activos en relación al número de dominios desde agosto de 1995 junio 2016



Información sobre servidores

- **ServerWatch** analiza los servidores desde diferentes perspectivas

<http://www.serverwatch.com/stypes/index.php/V2Vi/>

- **NetCraft Web Server Survey** realiza un seguimiento del software utilizado. Se publica mensualmente.

<http://www.netcraft.com>

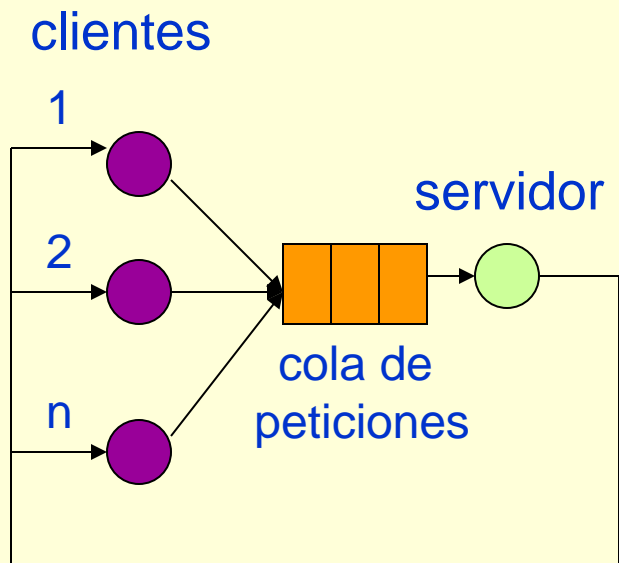
- **SPEC** ofrece comparativas

<http://www.spec.org/benchmarks.html>

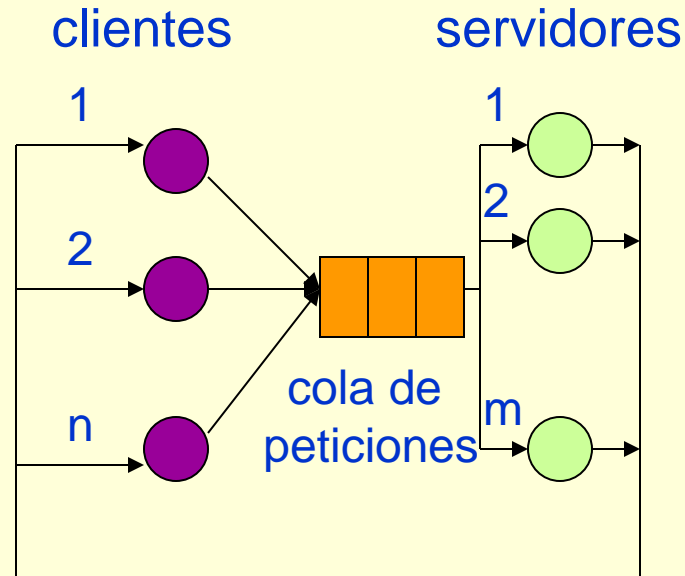
- **W3Techs** provee información sobre las tecnologías utilizadas.

<http://w3techs.com/>

Evolución del software del Servidor



Servidor de proceso simple



Servidor de proceso múltiple

Evolución del software del Servidor

Antecedentes

- Las peticiones HTTP están asociadas al puerto 80 del computador
- Existe un proceso llamado *inetd* que cada vez que llega una petición de internet a un determinado puerto, crea mediante un *fork()* y un *exec()* un proceso asociado a ese puerto para dar el servicio correspondiente.
- El servidor **standalone** mantiene siempre un proceso escuchando asociado al *http*.

Servidores Forking

- El *httpd* hace solo un *fork* y se copia a sí mismo evitando el *exec*. Útil para peticiones accesos a páginas *html* que generan muchas operaciones *http*.
- Los servidores tipo **Apache 1** son *preforking*. Cuando se inicia el *http* en el Apache inmediatamente se inician varios (5) procesos adicionales para manejar las múltiples peticiones. Mejoran el servicio pero consumen mucha CPU.

Evolución del software del Servidor

Servidores de hilos (Multi-threaded)

- Los threads son hilos independientes de ejecución dentro de un proceso simple.
- La creación de un thread y un cambio de contexto es 10 veces más rápido que la creación de procesos. Aligeran las conexiones http.
- Netscape Enterprise 2.0 y siguientes son servidores thread que, actualmente pueden servir miles de conexiones por segundo.

Servidores mixtos

- Tienen varios procesos paralelos esperando las conexiones de los clientes
- Cada proceso tiene varios threads para servir las peticiones
- Es la configuración por defecto del servidor Apache 2

Evolución del software del Servidor

- **Conexiones Keepalive**

- Los servidores mantienen las conexiones TCP abiertas (el proceso siempre está escuchando en el puerto) para servir múltiples archivos en una sola conexión.
- Esta técnica forman parte del standard HTTP 1.1.
- El problema es que los clientes se pueden desconectar sin notificárselo al servidor, la conexión se mantiene consumiendo recursos. Hay que poner un parámetro de timeout.

Evolución del software del Servidor

- **Cuándo falla un servidor**
 - Los servidores (preforking o threaded) manejan un nº limitados de procesos o hilos y rechazan el resto de conexiones (mensaje “Connection Refused”).
 - ¿qué pasaría si un servidor fuera capaz de manejar infinitas conexiones?
 - El manejo y planificación de los distintos procesos que compitan por la CPU consumiría casi todo el tiempo de la CPU.
 - La CPU se convertiría en un cuello de botella y las prestaciones se reducirían

Evolución del software del Servidor

- **Parámetros comunes en los servidores**

- **Nº de procesos ó Nº de Threads:** mejores prestaciones ejecutando sólo los procesos que se puedan mantener en RAM.
- **Conexiones persistentes:** Habilitar las conexiones persistentes y poner alto el nº de peticiones permitidas por conexión (*Max-KeepAliveRequest* 200) ahorra el overhead de lanzar nuevas conexiones. Poner el timeout bajo (\approx 15 sec.) *KeepAliveTimeout* 15, reduce el impacto de clientes muertos.
- **Loggins:** Desactivar la facilidad de DNS inverso. El servidor maneja las direcciones IP.

Clientes



Navegadores



Los Clientes

- Los Clientes suelen ser los navegadores
- Funciones de los navegadores:
 - Convertir las acciones del usuario en peticiones HTTP
 - Interpretar las URI según el protocolo HTTP
 - Resolver las direcciones por nombres localmente o accediendo al DNS
 - Gestionan las conexiones con el Servidor:
 - Usualmente abren varias conexiones con el servidor simultáneamente
 - Deciden cuando abrir una nueva conexión, reutilizar una existente o cerrarla
 - Recibir las respuestas del servidor, procesarlas y mostrarlas en la pantalla

Los Clientes

- Funciones de los navegadores (cont.)
 - La unidad básica de navegación es la página, compuesta de varios objetos o recursos
 - Los navegadores deben componer las páginas identificando los objetos que las componen
 - Gestionar la cache

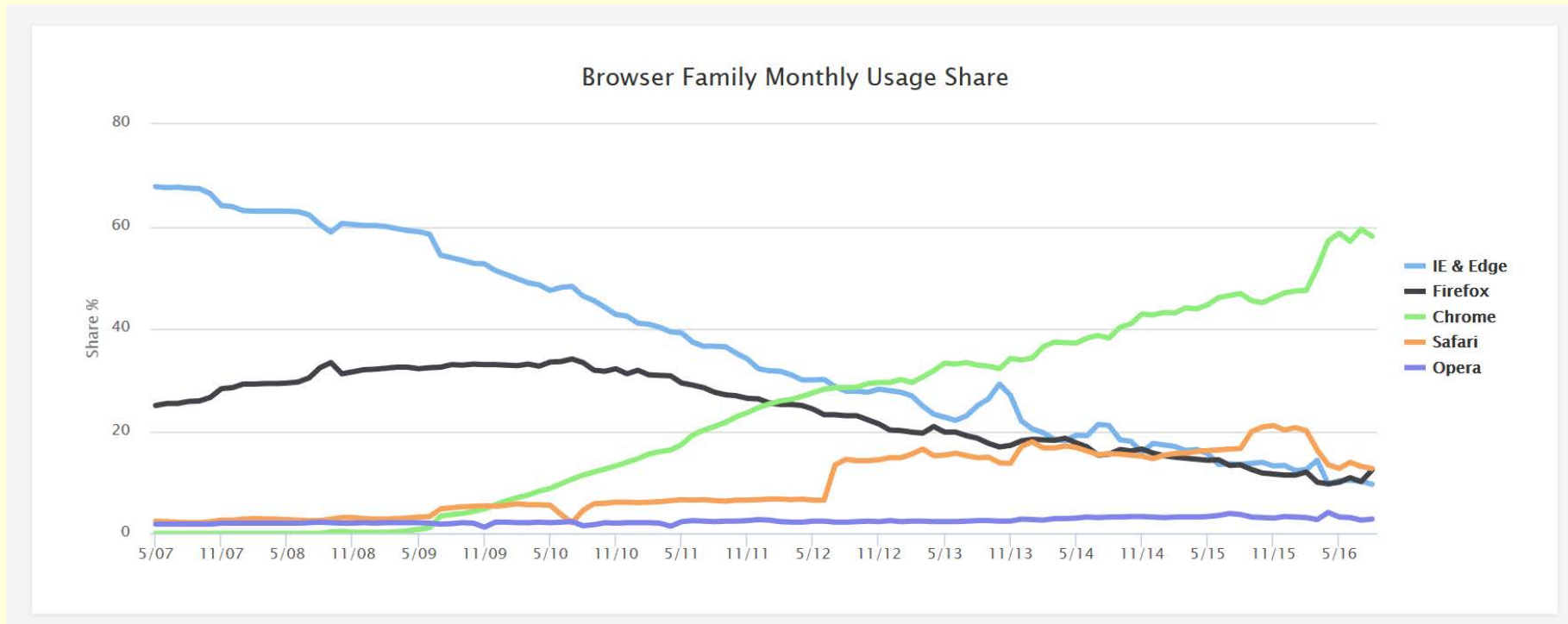
Los Clientes

- Existen varias tecnologías que permiten procesar datos en el Cliente
- En general se emplean para cálculos básicos, detección de errores, control de la GUI, ...
- La programación en el lado del Cliente la ejecuta el navegador (browser)
- Los programas (scripts) se encuentran a menudo incluidos en los documentos HTML, aunque también pueden estar en archivos distintos.
- Algunas tecnologías soportadas por los Clientes:
 - JavaScript, Flash, Java, ActiveX, VBScript, ...

Parámetros de los navegadores

- Número de conexiones:
 - El navegador puede lanzar varias conexiones para solicitar los recursos
 - Se puede gestionar:
 - El número total de conexiones
 - El número de conexiones por servidor
 - El número de conexiones persistentes por servidor
- Un número alto de conexiones puede mejorar la latencia pero afecta negativamente a las prestaciones del servidor
- El estándar HTTP recomienda hasta 2 conexiones persistentes al mismo servidor pero los navegadores actuales suelen abrir más.

Evolución de los navegadores

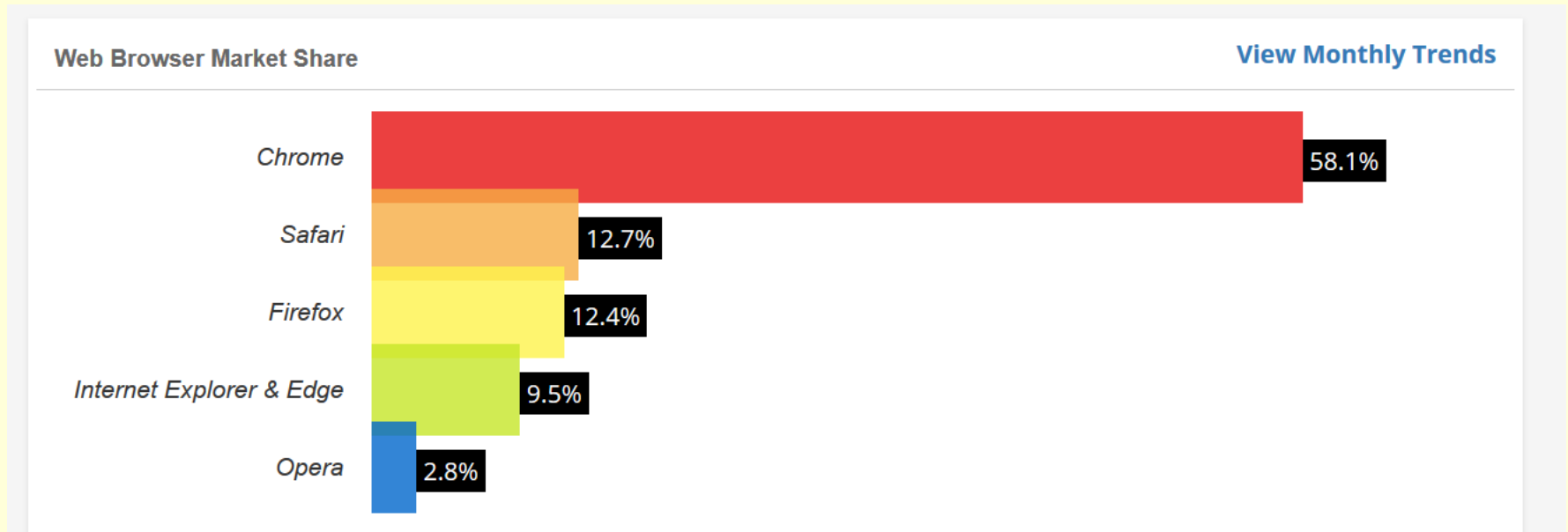


Evolución de las familias de navegadores desde mayo de 2007 hasta mayo de 2016

Fuente: <http://www.w3counter.com>

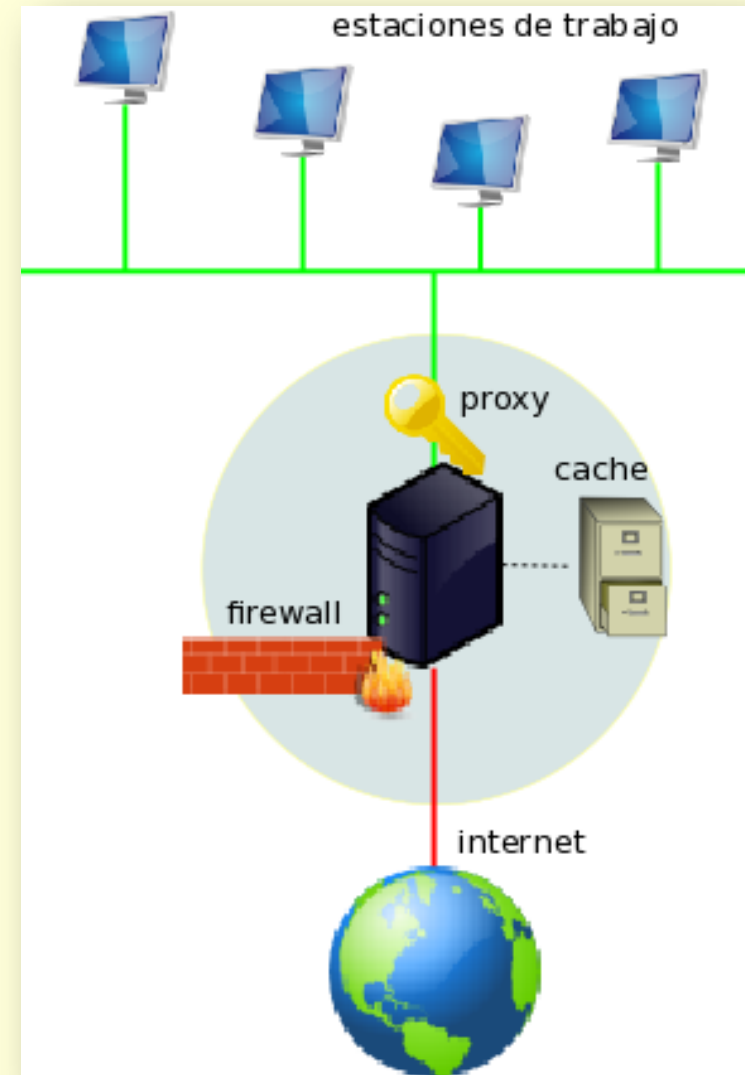
Evolución de los navegadores

Agosto 2016



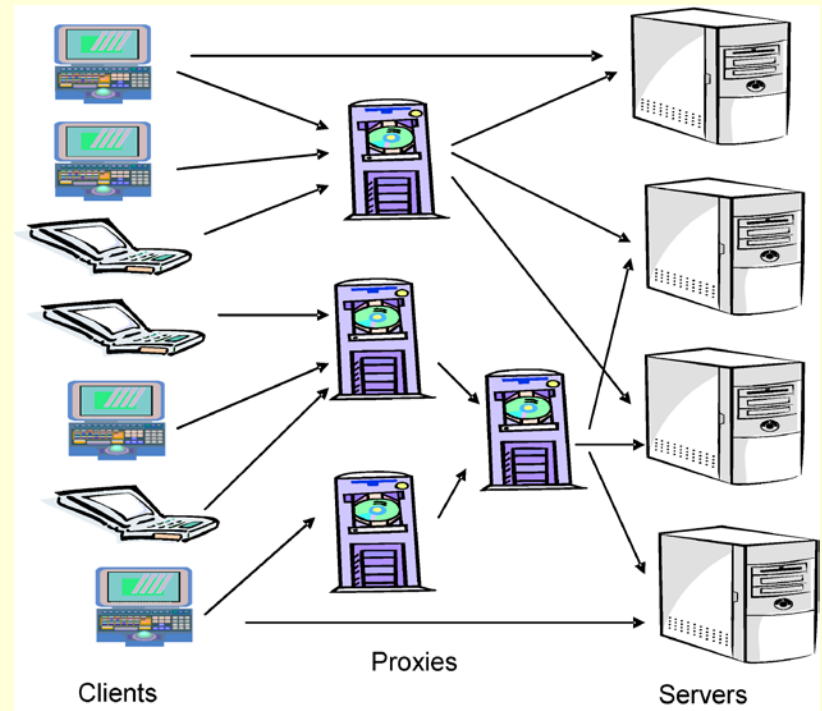
Proxies

- Es un intermediario en la transacción WEB
- Se sitúan entre el cliente y el servidor y actúan como ambos.
- A menudo ofrecen mecanismos de seguridad y actúan como firewalls.
- Permiten y almacena peticiones desde la red interna a internet.
- Se les conoce como pasarelas a nivel de aplicación (modelo OSI)
- Utiliza dos conexiones TCP una para el cliente y otra para el servidor



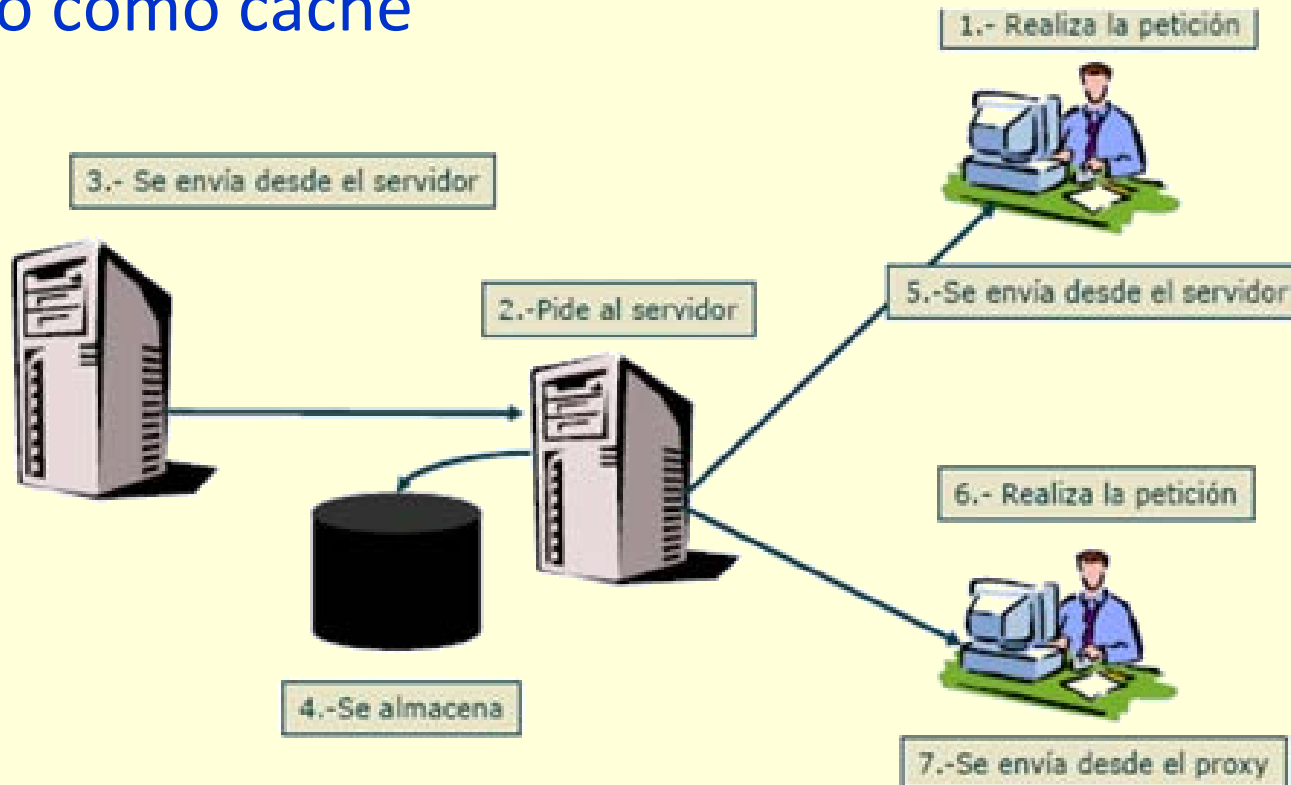
Proxies

- Utilidad de los Proxies:
 - control de acceso
 - filtrado de paquetes
 - control de virus
 - traducción de direcciones
 - **Caching**



Proxy cache

- Uso como cache



Proxies

- Ejemplos de proxies:
 - Squid cache
 - Microsoft Forefront Threat Management Gateway (Microsoft Internet Security and Acceleration Server)
 - Apache Trac Server
 - WinGate
- Algunos servidores web populares también ofrecen capacidad de proxy
 - Apache web server
 - nginx
 - lighthttpd

Prestaciones de la Web

- Los estudios de evaluación de las prestaciones de una arquitectura web se enfocan desde 3 perspectivas diferentes:
 - Servidor
 - Red
 - Cliente

Prestaciones en el Servidor

- Desde el punto de vista del Servidor nos interesa:
 - El número de peticiones que puede servir por unidad de tiempo
 - El tiempo en servir una petición
 - Un servidor proporciona buenas prestaciones si es capaz de servir muchas peticiones por segundo
 - Los servidores poco cargados responden más rápido
 - Métricas usadas:
 - Peticiones servidas por segundo
 - Tiempo de servicio

Prestaciones en la Red

- Las prestaciones de la red están relacionadas con la LATENCIA (tiempo necesario para transferir la información)
- La red proporciona buenas prestaciones cuando sirve un alto número de bytes por segundo
- Una red poco cargada es más rápida
- Principales componentes de la **latencia** de red:
 - **Tiempo de propagación:** depende de la distancia entre nodos
 - **Tiempo de transmisión:** depende del ancho de banda disponible y la cantidad de bytes transferidos
 - **Tiempo de encolamiento:** depende de los puntos intermedios, la tecnología y la carga

Prestaciones en el usuario

- Los usuarios perciben buenas prestaciones cuando el tiempo de descarga de las páginas es reducido
- Métrica principal: **LATENCIA PERCIBIDA**. Tiempo desde que se hace la petición hasta que se obtiene la respuesta.
- Determinantes de esta latencia:
 - El tiempo dedicado a obtener los recursos que forman una página.
Depende de:
 - El tiempo de búsqueda en el DNS
 - La latencia de red
 - El tiempo de servicio
 - El tiempo formateando la página para visualizarla

Latencia percibida por el usuario

- **Firebug:** extensión para Firefox para la ayuda avanzada de diseño de sitios web
- Proporciona la siguiente información por cada recurso descargado:
 - DNS Lookup: tiempo de resolución del DNS
 - Connecting: Tiempo requerido para establecer una conexión TCP.
 - Blocking: Tiempo transcurrido esperando conexión de red por encolamiento. Se muestra solo cuando esto ocurre.
 - Sending: Tiempo necesario para transferir la petición al servidor.
 - Waiting : Tiempo esperando la respuesta (hasta la obtención del primer byte).
 - Receiving: Tiempo necesario para la descarga.

Latencia percibida por el usuario

