



Redes de Datos

LSI – Plan 2009

Departamento de Informática - Facultad de Ciencias Exactas, Naturales y Agrimensura

UNIVERSIDAD NACIONAL DEL NORDESTE

Tema 2. Capa de aplicación

Contenido

- El Sistema de Nombres de Dominio (DNS)
- Protocolo de aplicación HTTP/HTTPS
- Almacenamiento: soluciones para redes corporativas (SAN), soluciones para redes compartidas (NAS). Propuestas de alta disponibilidad. Protocolos iSCSI, NFS y CIFS/Samba.

2.2 Capa Aplicación – Protocolo HTTP

- El Protocolo de transferencia de hipertexto (HTTP, HyperText Transfer Protocol) es un protocolo de capa de aplicación Web. Está definido en los documentos [RFC 1945] y [RFC 2616].
- HTTP utiliza TCP como su protocolo de transporte subyacente (en lugar de ejecutarse por encima de UDP).
- El cliente HTTP primero inicia una conexión TCP con el servidor. Luego los procesos de navegador y de servidor acceden a TCP a través de sus interfaces de socket.
- Dado que un servidor HTTP no mantiene ninguna información acerca de los clientes, se dice que HTTP es un protocolo sin memoria del estado



2.2 Capa Aplicación – Protocolo HTTP

- HTTP define la estructura de estos mensajes y cómo el cliente y el servidor intercambian los mensajes.
- HTTP se implementa en dos programas: un programa cliente y un programa servidor. El programa cliente y el programa servidor, que se ejecutan en sistemas terminales diferentes, se comunican entre sí intercambiando mensajes HTTP.
- Una página o documento web consta de objetos. Un objeto es simplemente un archivo (como por ejemplo, un archivo HTML, una imagen JPEG, un applet Java o un clip de vídeo) que puede direccionarse mediante un único URL.
- La mayoría de las páginas web están constituidas por un archivo base HTML y varios objetos referenciados. El archivo base HTML hace referencia a los otros objetos contenidos en la página mediante los URL de los objetos. Cada URL tiene dos componentes: el nombre de host del servidor que alberga al objeto y el nombre de la ruta al objeto.
- Cuando el navegador recibe la página web, la muestra al usuario. Dos navegadores distintos pueden interpretar (es decir, mostrar al usuario) una página web de formas distintas. HTTP no tiene nada que ver con cómo un cliente interpreta una página web.
- Las especificaciones HTTP ([RFC 1945] y [RFC 2616]) únicamente definen el protocolo de comunicación entre el programa HTTP cliente y el programa HTTP servidor.

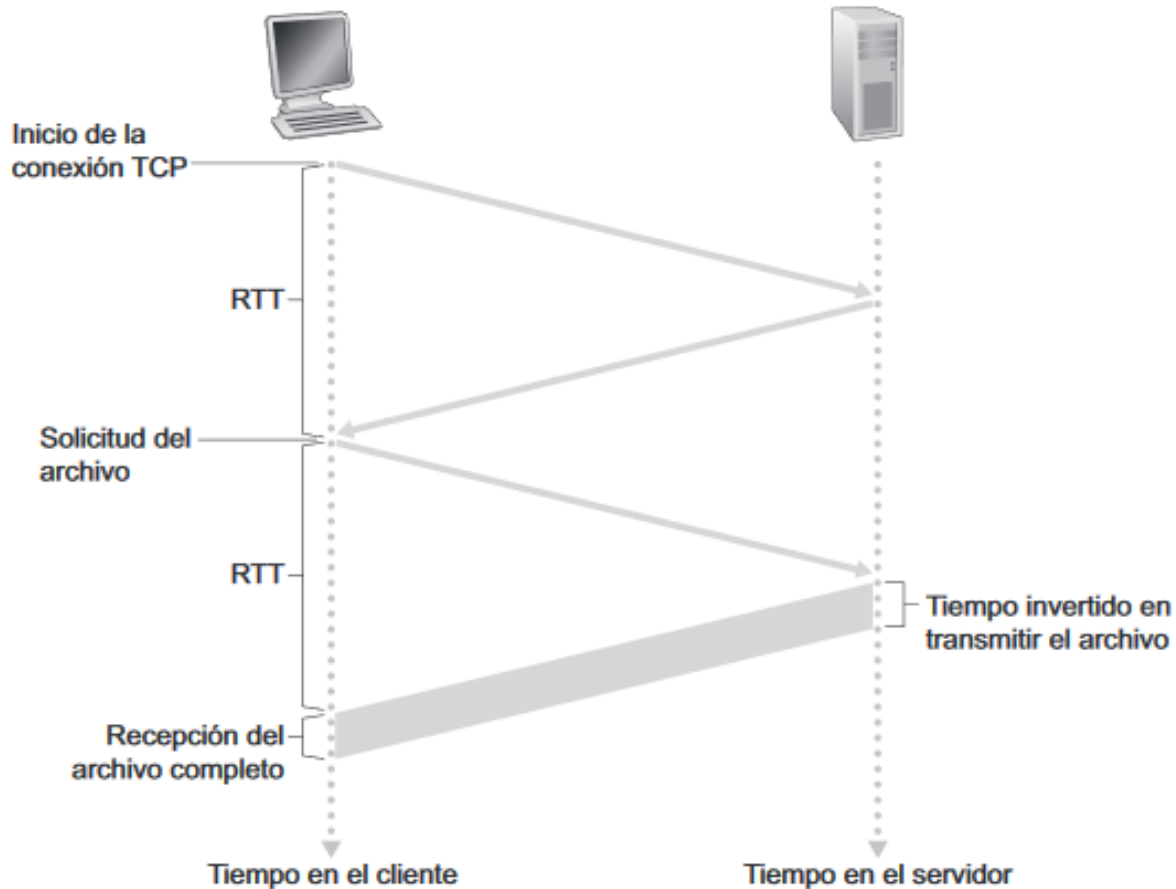
2.2 Capa Aplicación – Protocolo HTTP

Conexión con Persistencia, y sin Persistencia.

- Muchas aplicaciones de Internet, el cliente y el servidor están en comunicación durante un periodo de tiempo amplio, haciendo el cliente una serie de solicitudes y el servidor respondiendo a dichas solicitudes.
- Dependiendo de la aplicación y de cómo se esté empleando la aplicación, las solicitudes pueden hacerse una tras otra, periódicamente a intervalos regulares o de forma intermitente. Cada par solicitud/respuesta puede enviarse a través de una conexión TCP separada o bien, enviarse todas las solicitudes y sus correspondientes respuestas a través de la misma conexión TCP.
- HTTP emplea conexiones persistentes en su modo por defecto, los clientes y servidores HTTP sin embargo se pueden configurar conexiones no persistentes.
- Cada conexión TCP transporta exactamente un mensaje de solicitud y un mensaje de respuesta.
- El cliente (navegador web) puede ser configurado para controlar el grado de paralelismo. En sus modos por defecto, la mayoría de los navegadores abren de 5 a 10 conexiones TCP paralelo y cada una de estas conexiones gestiona una transacción solicitud-respuesta.
- En cuanto a capacidades.. se define el tiempo de ida y vuelta (RTT, Round-Trip Time), que es el tiempo que tarda un paquete pequeño en viajar desde el cliente al servidor y volver de nuevo al cliente.
- El tiempo RTT incluye los retardos de propagación (Ethernet) de los paquetes, los retardos de cola en los routers y switches intermedios y los retardos de procesamiento de los paquetes.

2.2 Capa Aplicación – Protocolo HTTP

Conexión con Persistencia, y sin Persistencia.



- El modo por defecto de HTTP utiliza conexiones persistentes con procesamiento en cadena.
- HTTP/2 [RFC 7540] amplía la funcionalidad de HTTP 1.1, permitiendo entrelazar múltiples solicitudes y respuestas en la misma conexión, y proporcionando también un mecanismo para priorizar los mensajes de solicitud y respuesta HTTP dentro de dicha conexión.

2.2 Capa Aplicación – Protocolo HTTP

Formato de mensajes.

Mensaje de solicitud HTTP

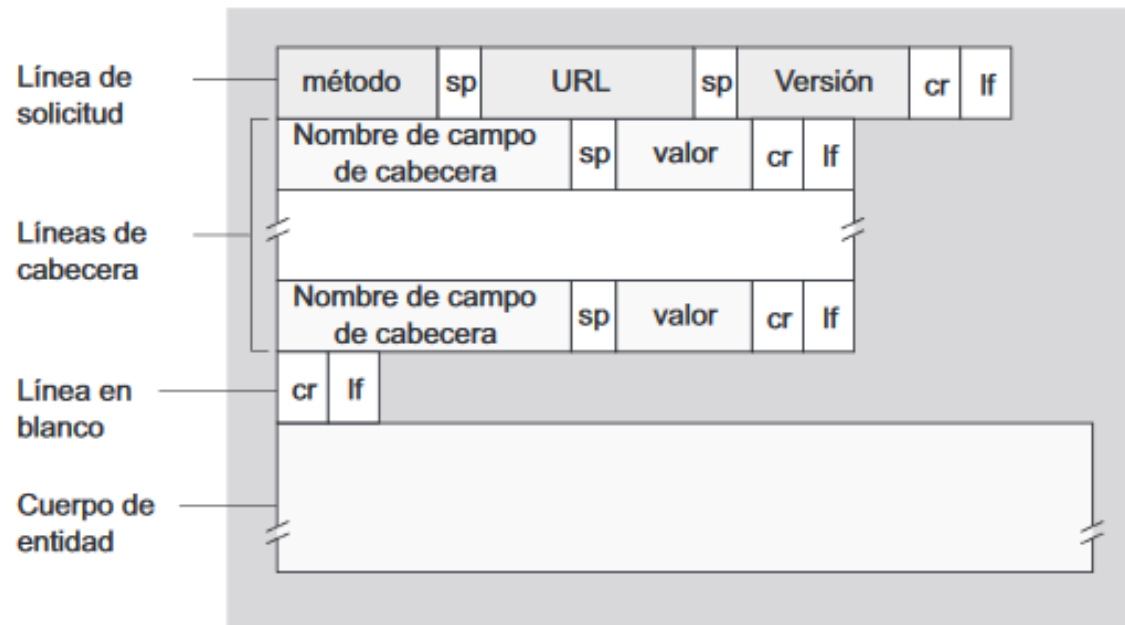
GET /unadireccion/pagina.html HTTP/1.1

Host: www.unaescuela.edu

Connection: close

User-agent: Mozilla/4.0

Accept-language: fr



Métodos:

- GET
- PUT
- HEAD
- DELETE

2.2 Capa Aplicación – Protocolo HTTP

Formato de mensajes.

Mensaje de respuesta HTTP

HTTP/1.1 200 OK

Connection: close

Date: Sat, 07 Jul 2007 12:00:15 GMT

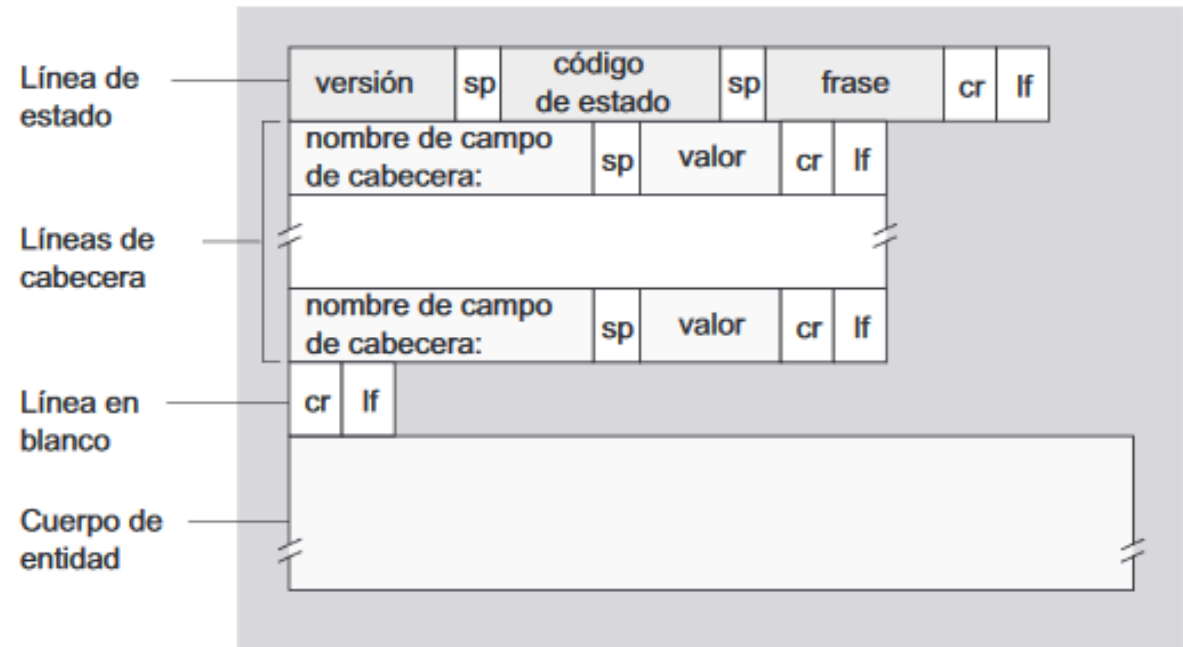
Server: Apache/1.3.0 (Unix)

Last-Modified: Sun, 6 May 2007 09:23:24 GMT

Content-Length: 6821

Content-Type: text/html

(datos datos datos datos datos ...)



Respuestas HTTP:

200 OK:

La solicitud se ha ejecutado con éxito y se ha devuelto la información en el mensaje de respuesta.

301 Moved Permanently:

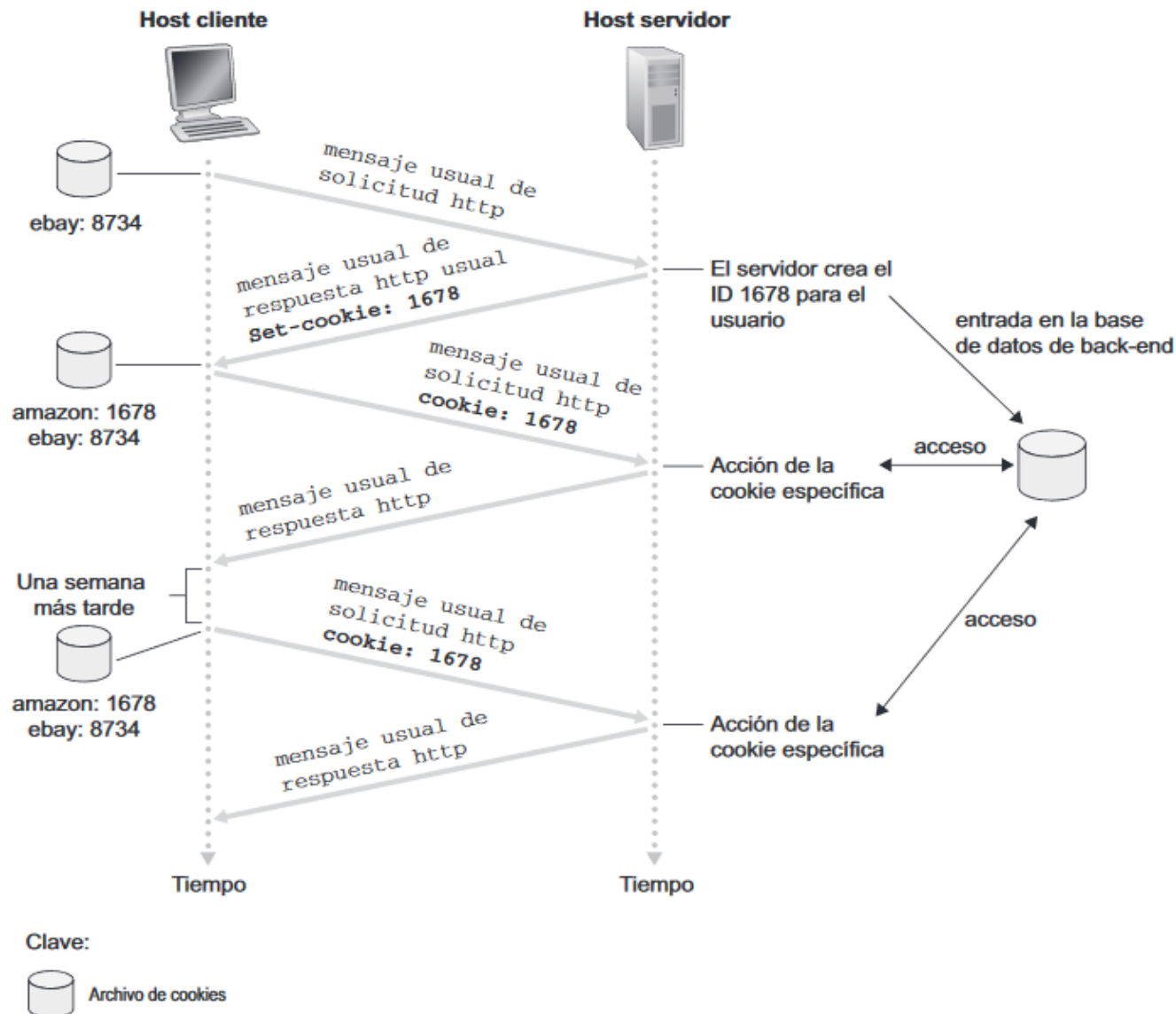
El objeto solicitado ha sido movido de forma permanente; el nuevo URL se especifica en la línea de cabecera Location: del mensaje de respuesta. El software cliente recuperará automáticamente el nuevo URL.

400 Bad Request:

Se trata de un código de error genérico que indica que la solicitud no ha sido comprendida por el servidor.

2.2 Capa Aplicación – Protocolo HTTP

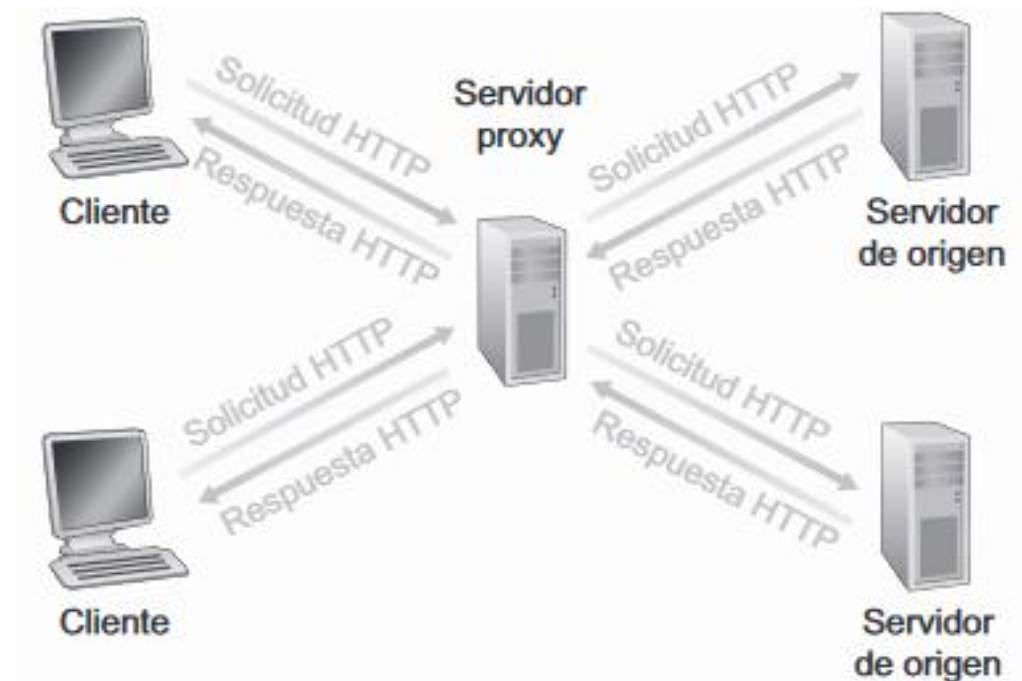
Interacción usuario-servidor: cookies.



2.2 Capa Aplicación – Protocolo HTTP

Almacenamiento en caché web

- Una caché web, o servidor proxy, es una entidad de red que satisface solicitudes HTTP en nombre de un servidor web de origen.
- Dispone de su propio almacenamiento en disco y mantiene en él copias de los objetos solicitados recientemente.
- El caché web, es a la vez un servidor y un cliente. Servidor: cuando recibe solicitudes de y envía respuestas a un navegador. Cliente: cuando envía solicitudes a y recibe respuestas de un servidor de origen.



2.2 Capa Aplicación – Protocolo HTTP

¿Por qué debería preocuparme por HTML5?

- En el núcleo de HTML5 se incluyen una serie de nuevos elementos semánticos, así como varias tecnologías y API relacionadas. Estas adiciones y cambios en el lenguaje permite a los desarrolladores crear páginas web más fáciles de codificar, usar y acceder.
- Estos nuevos elementos semánticos, junto con otros estándares como WAI-ARIA y Microdata ayudan a que los documentos sean de fácil lectura beneficiando tanto la accesibilidad y la optimización de motores de búsqueda.
- Los elementos semánticos, en particular, han sido diseñados con la Web dinámica en mente, con un enfoque particular en la fabricación modular de páginas más accesibles.
- Las API asociadas con HTML5 ayudan a mejorar en una serie de técnicas que los desarrolladores web han estado utilizando durante años. Muchas de las tareas comunes se han simplificado, poner más poder en manos de los desarrolladores.
- La introducción de audio y vídeo en HTML5 significa que habrá una menor dependencia de plugins y software de terceros al publicar contenido multimedia en la Web.

Fuente: HTML5 & CSS3 FOR THE REAL WORLD, BY ALEXIS GOLDSTEIN, LOUIS LAZARIS and ESTELLE WEYL

FUENTES

- Redes de Computadoras, un enfoque descendente - James F. Kurose (Author), Keith W. Ross (Author) – Capítulo 2.2.1
- Fuente: HTML5 & CSS3 FOR THE REAL WORLD, BY ALEXIS GOLDSTEIN, LOUIS LAZARIS and ESTELLE WEYL

Ejercicios previstos:

- Instalación de Apache Win/Linux, gestión de sitio monolítico, múltiples sitios.
- Gestión de difusión HTTP por Microservicios.