



1

La capa de aplicación

La Web y HTTP (2ª parte).

Tema 2: La capa de aplicación

2

- 2.1 Principios de las aplicaciones de red.
- 2.2 La Web y HTTP (2ª parte).**
- 2.3 Correo electrónico en Internet.
- 2.4 DNS: el servicio de directorio de Internet.
- 2.5 Distribución de archivos P2P.
- 2.6 Flujos de vídeo y redes de distribución de contenidos.
- 2.7 Programación de sockets: creación de aplicaciones de red.

HTTP

3

□ Interacción usuario-servidor

- Hay sitios Web que requieren identificar los usuarios, porque:
 - Son de acceso restringido (control de acceso), y/o
 - Dan servicio en función de la "identidad" del usuario.

- En consecuencia, HTTP dispone de dos mecanismos para identificación de usuarios:
 - Autorización, y
 - "Cookies".

HTTP

4

□ Autorización HTTP

- Mecanismo "específico" para identificación de usuarios previa solicitud del servidor.
 - Código de Estado "401 Authorization Required".
 - Con cabecera "Authenticate".

- Mensaje de solicitud debe contener "credenciales".
 - En cabecera "Authorization".
 - Credenciales: "nombre de usuario" y "clave".
 - De manera transparente para el usuario.
 - Se teclean una vez, permanecen en caché de la máquina
 - Sin estado.
 - Credenciales en cada solicitud.
 - Con estado.
 - Servidor asocia una "cookie".

"Cookie" de autorización, caso particular de cookies como mecanismo específico de identificación de usuarios

HTTP

5

□ Cookies HTTP (RFC 2965)

- Mecanismo “genérico” para identificación de usuarios.
- Permite identificar la combinación ordenador – navegador.
- Procedimiento utilizado por muchos sitios Web.
 - Para hacer “seguimiento” de la actividad de sus usuarios.
 - P.e, sitios de publicidad, de comercio electrónico, etc.
 - En principio para dar mejor servicio, servicio personalizado...
 - P.e, para autorización, “carritos” de la compra, etc.
 - **!!!Muy controvertido!!!**, de alguna manera se viola la privacidad del usuario.
 - Permite saber “cosas” sobre los usuarios.
 - Nombre, dirección, correo, hábitos de consumo, contenidos que visita, cuántas veces, en qué orden, etc.
 - Posibilita negocios entre “bambalinas”, las compañías publicitarias pueden intercambiar información, etc.

HTTP

6

□ Componentes para el soporte de cookies.

- Cabeceras específicas para cookies.
 - En mensajes de solicitud y de respuesta.
- Archivo de cookies.
 - En el sistema de usuario.
 - Lo gestiona el navegador.
- Base de datos de respaldo.
 - En el sitio (servidor) Web.
 - Lo gestiona el servidor.

HTTP

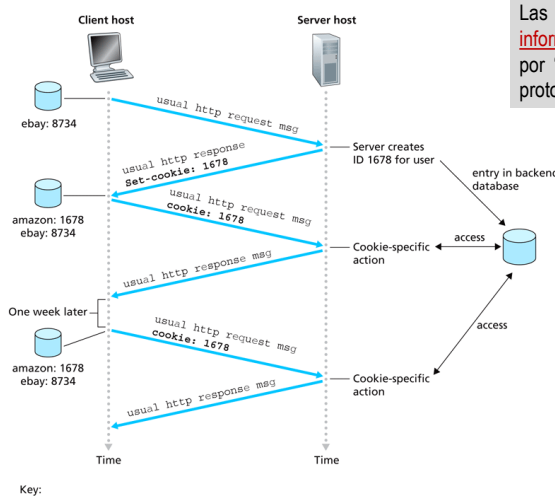
7

- Procedimiento con cookies
 - Usuario accede por primera vez a un sitio Web que implementa Cookies: el servidor:
 - Crea un nº de identificación único para el usuario.
 - Crea una entrada indexada con ese nº de identificación en su base de datos.
 - Responde con un mensaje que incluye cabecera "Set-Cookie" con el nº de identificación.
 - P.e: Set-Cookie: 1678 ← En mensajes de respuesta.
 - Navegador, cuando recibe mensaje de respuesta que contiene una cookie:
 - Añade una línea al archivo de cookies que gestiona. Incluye básicamente:
 - "Nombre del servidor".
 - "Nº de identificación" (la cookie).
 - Luego, cada vez que el usuario desee "navegar" por un sitio Web:
 - Navegador consulta el fichero de cookies.
 - Si hay cookie para el sitio, extrae el nº de identificación y lo incluye en la cabecera "Cookie".
 - P.e: Cookie: 1678 ← En mensajes de solicitud.
 - Así para cada solicitud...

HTTP

8

Ejemplo de procedimiento con cookies



Las cookies proveen cierta información de "estado de sesión" por "encima" de HTTP, que es un protocolo que no maneja estados.

Figure 2.10 • Keeping user state with cookies

HTTP

9

□ Caché WEB

- Procedimiento para mejorar las prestaciones del servicio Web. Permite reducir:
 - El tiempo de obtención de un objeto (latencia), en particular para:
 - Objetos grandes y poco ancho de banda (AB) de acceso.
 - Sobre todo si **AB limitante** entre cliente y servidor < **AB limitante** entre cliente y caché Web.
 - El tráfico en la red → Mayor rendimiento para todas las aplicaciones.
 - De manera global y especialmente en los enlaces de acceso a Internet. P.e, en un acceso institucional.
 - Permite aprovechar mejor la tasa del acceso disponible → Reducción de costos.
 - La carga de trabajo de servidores Web.
- Es almacenamiento en memoria temporal (caché) de copias de objetos Web.
 - En el cliente (lo gestiona el navegador).
 - En un servidor de red, como proxy (intermediario) Web.
- Problema: actualidad de los objetos en caché.
 - Si un objeto se actualiza/modifica en el servidor, la copia en caché NO es válida.
 - Solución: **GET condicional**.

HTTP

10

Todo mensaje de respuesta relativo a un objeto contiene la cabecera "Last-modified".
Valor de la cabecera "Last Modified" = fecha/hora de creación/actualización del objeto.

GET condicional

- Mecanismo HTTP para comprobar y garantizar la actualidad de los objetos en caché.
- Se implementa mediante mensaje GET:
 - Solicitud condicionada, se pide un objeto sólo si ha sido modificado respecto a la copia en caché.
 - Con cabecera específica "If-Modified-Since".
 - Contiene fecha/hora de creación/actualización del objeto que está en caché.
 - Información obtenida del mensaje de respuesta que antes portó el objeto (o información del objeto).

Respuesta al GET condicional

- Si el objeto solicitado NO ha sido modificado.
 - Servidor responde con "**304 Not modified**" y el cuerpo del mensaje vacío.
- Si el objeto solicitado ha sido modificado.
 - Servidor responde con "**200 OK**" y el cuerpo del mensaje contiene el objeto solicitado.
 - Cabecera "Last-modified" contiene fecha/hora de la última actualización.

HTTP

11

- Ejemplo de “GET condicional”.

Mensaje de Solicitud (GET normal)

GET /fruta/kiwi.gif HTTP/1.0
User-agent: Mozilla/4.0

Mensaje de Solicitud (GET condicional)

GET /fruta/kiwi.gif HTTP/1.0
User-agent: Mozilla/4.0
If-modified-since: Mon, 22 Jun 1998 09:23:24

Mensaje de Respuesta

HTTP/1.0 200 OK
Date: Wed, 12 Aug 1998 15:39:29
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 22 Jun 1998 09:23:24
Content-Type: image/gif

(.....contenido del objeto.....)

Mensaje de Respuesta (objeto válido en caché)

HTTP/1.0 304 Not Modified
Date: Wed, 19 Aug 1998 15:06:12
Server: Apache/1.3.0 (Unix)

(.....cuerpo vacío, sin contenido.....)

HTTP

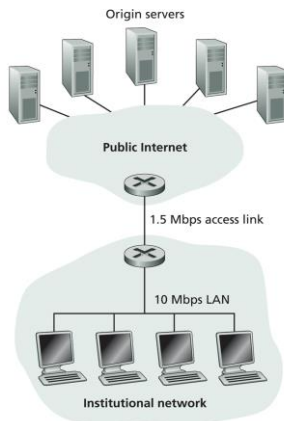
12

- Proxy: caché Web para uso compartido → **Servidor proxy**.
 - Satisface solicitudes HTTP en nombre de servidores Web. Tiene doble funcionalidad:
 - De cara al cliente Web actúa como servidor Web.
 - De cara al servidor Web actúa como cliente Web.
 - Almacena en disco copias de objetos recientes solicitados por clientes.
 - Utilizado también como mecanismo para:
 - Replicación/distribución de contenidos, control de acceso.
 - P.e: quién, tipo de contenidos, protocolos, etc.
 - Porcentaje de aciertos depende del nº usuarios.
 - Más usuarios → Mayor porcentaje de aciertos de la caché (en la práctica 20 – 70 % de aciertos).
 - $\% \text{ de aciertos} = \text{N}^\circ \text{ total de aciertos} / \text{N}^\circ \text{ total de solicitudes}$.
 - Ejemplo de lo que puede aportar un proxy Web.
 - Ver análisis en el libro de texto: **desde página 92 (final) a página 94, figuras 2.12 y 2.13.**

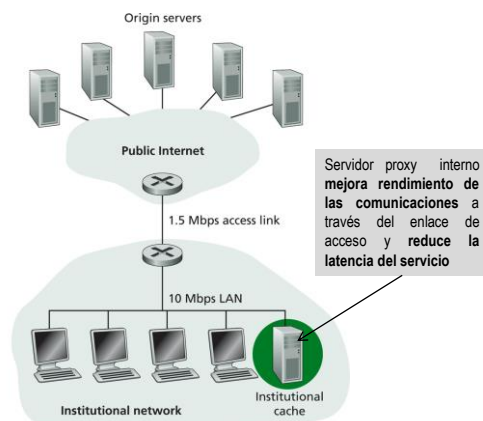
HTTP

13

“Cuello de botella” sin proxy



Servidor proxy en la Intranet



El servidor proxy en la Intranet requiere:
Intercepción: mecanismo de re direccionamiento de solicitudes Web a un proxy, o
Configuración del Navegador Web

Normalmente todo ISP dispone de un Proxy

HTTP

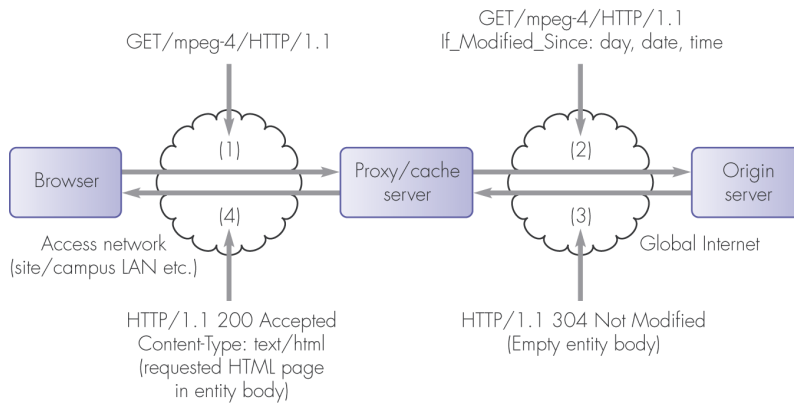
14

- Configuración de Navegadores.
 - ▣ Normalmente pueden ser configurados para dirigir solicitudes a un proxy o a varios...
 - ▣ Se requiere información por parte del ISP:
 - Nombre de la/s máquina/s que actúa/n como proxy.
 - Puerto a través del cual da/n el servicio.
 - ▣ P.e, en el navegador “Internet Explorer” de Microsoft (v.8).
 - En Herramientas:
 - Opciones Internet → Conexiones.
 - Configuración LAN → Servidor Proxy.
 - En el recuadro que aparece introducir información suministrada por el ISP.

HTTP

15

“GET condicional” con servidor Proxy.



Fuente: Halsall, *Computing Networking and the Internet*, 5th Edition © Pearson Education Limited 2005

HTTP

16

□ Reglas para el caché Web.

Determinan cuándo servir un contenido/objeto a partir de la caché.

- En el protocolo HTTP (v1.0 y v.1.1):
 - Cabeceras “Expires” y “Cache-control”.
- En el formato HTML:
 - “Meta Tags”.
 - Son directivas HTML para gestión de la caché Web.
 - Poco efectivas, los navegadores y proxy normalmente no leen el contenido del cuerpo de mensajes HTTP.
- Por parte del administrador de la caché.
 - Usuario del navegador.
 - Administrador del proxy.

HTTP

17

- Reglas más comunes del caché Web.
 - No se guarda objeto en caché si:
 - Cabeceras de mensajes de respuesta establecen “no caché”, y/o
 - Mensajes de respuesta sin información de “frescura”.
 - Si objeto en caché es “fresco”:
 - Se sirve de inmediato, sin consultar el servidor Web.
 - Objeto es “fresco” si:
 - Tiene cabecera de “tiempo de expiración” o de “control de edad” vigente.
 - Es reciente y tiene fecha de modificación de hace mucho tiempo.
 - Si objeto en caché no es “fresco”:
 - Se consulta el servidor para comprobar validez de la caché o descargar copia “fresca”.

Si no se dispone de conexión con el servidor la caché Web puede servir objetos sin consultar el servidor, pero no da certeza de la actualidad de los objetos.

HTTP

18

- Permanencia de objetos en Caché Web.
 - Según de tipo de contenido:
 - Muy dinámicos, que varían con frecuencia.
 - Se hacen obsoletos muy rápidamente.
 - En principio nunca deberían guardarse en caché.
 - Estáticos, o cuasi-estáticos, que varían muy poco o casi nada en el tiempo.
 - Adecuados para caché Web.
 - Elementos de referencia, entre otros, las cabeceras:
 - Last-Modified, Expires, Cache-Control.



HTTP

19

- Otros mecanismos de red para caché Web.
 - Caché jerárquico.
 - Caché pasivo.
 - Caché activo.
 - Caché cooperativo (requiere protocolo de coordinación entre proxys).

- Otros mecanismos para mejora del servicio Web.
 - Réplica de servidores.
 - “Espejos” (mirrors) o “granjas” (farms) de servidores.