

1 La capa de aplicación.

La Web y HTTP (1ª parte).

Introducción a la Web

2

WWW: World Wide Web → La Web

- Servicio para acceso a información, a recursos vinculados y distribuidos en muchos hosts.
- Permite:
 - ▣ Recibir y publicar información
 - ▣ “Navegar” por un “océano” de información a través de hiperenlaces, y la interacción de diferentes páginas Web.
- Funciona:
 - ▣ Bajo demanda.
 - Usuarios reciben y publican “lo que quieren” y “cuando quieren”
 - ▣ Mediante el modelo Cliente – Servidor.

Introducción a la Web

3

- Elementos de la arquitectura Web
 - Páginas.
 - HTML, hipertexto, hiperenlaces.
 - Objetos
 - Pueden residir o no en el mismo host.
 - Direccionado
 - URL: Uniform Resource Locator.
 - Entidades funcionales.
 - Cliente Web (navegador).
 - Servidor Web.
 - Protocolo HTTP.

Introducción a la Web

4

- Páginas
 - Contienen “objetos”
 - Resultado que muestra el navegador tras seguir instrucciones del fichero HTML.
 - Generalmente formada por:
 - Fichero HTML base.
 - Referencia a los otros objetos de la página mediante sus correspondientes URL.
 - Otros objetos referenciados en diferentes formatos.
 - Texto ordinario, hipertexto, imágenes, vídeos, audio ...
 - Vinculados o no a otras páginas.
- Objetos
 - Archivos:
 - HTML, JPEG, etc.
 - Cada uno direccionable a través de un URL.

Introducción a la Web

5

□ Direccionado

- En base a un URL (Universal Resource Locator), elemento para el direccionamiento de recursos.
- Parámetros del URL:
 1. Identificador de servicio (protocolo).
 2. Nombre de dominio del host servidor.
 3. Ruta de acceso al objeto.
 4. Nombre del objeto.

Hay definidos otros tipos de URL,
según tipo servicio (protocolo de acceso)

□ Sintaxis URL.

- Caracteres “://” separan primer parámetro del segundo.
- Caracter “/” separa resto de parámetros entre sí.

□ Ejemplo:

■ http://www.escuela.es/dptotic/imagen1.jpg

1 2 3 4

Introducción a la Web

6

□ Cliente Web (navegador)

- Comprende:
 - Agente de Usuario.
 - Software que actúa como interfaz entre usuario y aplicación Web.
 - Permite “atrapar” y visualizar páginas, y “navegar” por sus contenidos.
 - Presenta diferentes características de configuración.
 - Cliente HTTP.
 - Software de comunicación que implementa la parte cliente HTTP.
- Navegadores más conocidos:
 - MS Internet Explorer.
 - Firefox.
 - Chrome, y otros más...

Introducción a la Web

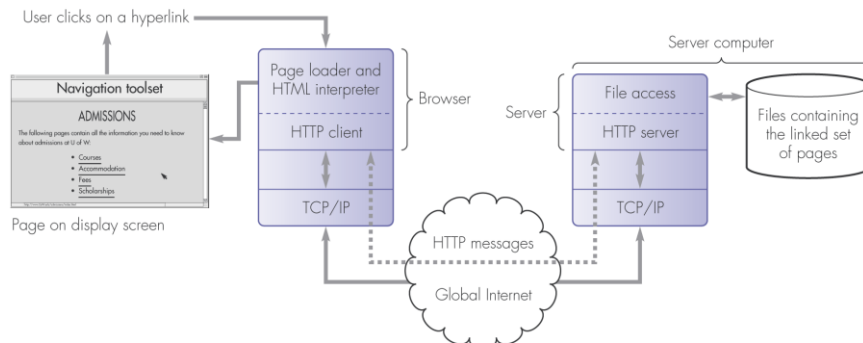
7

- Servidor Web
 - Comprende tres entidades funcionales:
 - Software de acceso a ficheros.
 - Base de datos.
 - Almacena objetos.
 - Cada objeto direccionable con el correspondiente URL.
 - Servidor HTTP.
 - Software de comunicación que implementa la parte servidor HTTP.
 - Servidores Web más conocidos:
 - Apache.
 - MS Internet Information Server.

Introducción a la Web

8

- Elementos de la arquitectura Web



Note: Hyperlinks contain a URL which includes the domain name of the server computer and the name of the file containing the HTML code of the selected page

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

HTTP

9

- HTTP: Hyper Text Transfer Protocol
 - ▣ Protocolo para la transferencia de recursos.
 - ▣ Recursos: archivos y resultados, p.e. de la:
 - Consulta a una base de datos.
 - Ejecución de un programa.
 - Traducción automática de un documento.
 - ▣ Dos versiones compatibles entre si:
 - HTTPv1.0, especificado en la RFC 1945.
 - HTTPv1.1, especificado en la RFC 2616.
 - ▣ Opera en base al modelo Cliente – Servidor.

HTTP

10

- HTTP: Hyper Text Transfer Protocol
 - ▣ Basado en simples “transacciones”: petición – respuesta.
 - Cliente solicita objetos, genera “peticiones”.
 - Mensajes ASCII (formato entendible por humanos).
 - Servidor transfiere objetos, genera “respuestas”.
 - Mensajes ASCII tipo MIME (RFC 822).
 - Es básicamente un protocolo de “demanda”.
 - ▣ Protocolo SIN estado y NO conectivo.
 - El servidor no guarda información de los clientes ni de las transacciones.
 - Simplifica el diseño de servidores → Servidores Web de gran rendimiento, pueden manejar muchas conexiones TCP y las correspondientes transacciones HTTP.
 - ▣ Protocolo no fiable y no seguro.

Los protocolos con estado son complejos

- Requieren mantener la “historia” pasada (el estado).
- Si servidor/cliente falla, sus puntos de vista del estado pueden ser inconsistentes

HTTP

11

- Protocolo de Transporte
 - El estándar NO especifica. En la práctica TCP → HTTP no requiere procedimientos de control de errores.
 - Toda transacción HTTP (solicitud –respuesta) requiere conexión TCP.
- Procedimiento
 - Cliente solicita conexión TCP al puerto 80 del servidor.
 - Servidor acepta conexión del cliente (se crea “socket de conexión” en el servidor).
 - Navegador y servidor intercambian mensajes HTTP.
 - Se cierra la conexión TCP.
- Tipos de conexiones TCP para sesiones HTTP
 - No persistentes.
 - Solución única para HTTPv1.0. Soportada también por HTTPv1.1.
 - Persistentes.
 - Solución por defecto para HTTPv1.1.

HTTP

12

- Conexión TCP no persistente
 - Conexión TCP propia para transferir cada objeto.
 - Diferentes objetos → Diferentes conexiones TCP.
 - Gestión de tantas conexiones TCP como número de objetos a transferir.
 - Independiente de la ubicación física de los objetos.
 - Cada conexión TCP sólo soporta una transacción HTTP (petición y respuesta).
 - Modos:
 - Serie.
 - Sólo una conexión TCP abierta en cada momento.
 - Paralelo.
 - Posibles varias conexiones TCP abiertas a la vez.

HTTP

13

Conexión TCP no persistente SERIE

Para "N" objetos a descargar, todos de igual tamaño, el tiempo total de descarga es:

$$T_{T-NPS} = N \cdot (2 \cdot RTT + X_p)$$

Supuestos:

- 1) Despreciables los tiempos de:
 - Procesamiento en navegador y servidor Web.
 - Transmisión de los segmentos TCP que no portan información del objeto.
- 2) Solicitud del objeto en el tercer segmento de apertura de la conexión TCP.
 - No siempre es así.

Lo representado en la figura acontece después que se ha descargado el fichero HTML base.

Tema 2: La capa de Aplicación

HTTP

14

Conexión TCP no persistente PARALELO

El tiempo total de descarga para "N" objetos a descargar, todos de igual tamaño y grado de paralelismo "N", es:

$$T_{T-NPP} \approx 2 \cdot RTT + X_p \lll T_{T-NPS} = N \cdot (2 \cdot RTT + X_p)$$

Supuestos:

- 1) Despreciables los tiempos de:
 - Procesamiento en navegador y servidor Web.
 - Transmisión de los segmentos TCP que no portan información del objeto.
- 2) Solicitud del objeto en el tercer segmento de apertura de la conexión TCP.
 - No siempre es así.

Lo representado en la figura acontece después que se ha descargado el fichero HTML base, y conlleva a compartir "simultáneamente" el ancho de banda entre los "N" objetos.

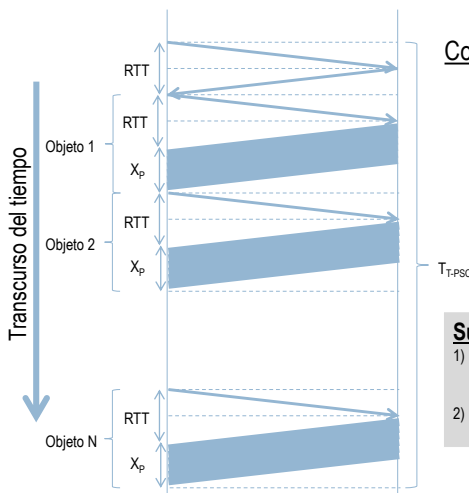
HTTP

15

- Conexión TCP persistente
 - Por una misma conexión TCP:
 - Discurren todas las transacciones HTTP.
 - Se transfieren todos los objetos de una página si están todos en el mismo servidor.
 - Transcurrido un tiempo (configurable) sin transacciones el servidor cierra la conexión TCP.
 - Modos.
 - Sin canalización (sin "pipelining").
 - Sólo una transacción HTTP a la vez.
 - Con canalización (con "pipelining").
 - Admite varias transacciones HTTP a la vez → Procesamiento en cadena.
 - Solución por defecto para HTTPv1.1

HTTP

16



Conexión TCP persistente SIN CANALIZACIÓN

Para "N" objetos a descargar, todos de igual tamaño, el tiempo total de descarga es:

$$T_{T-PSC} = RTT + N.(RTT + X_p)$$

Supuestos:

- 1) Despreciables los tiempos de:
 - Procesamiento en navegador y servidor Web.
 - Transmisión de los segmentos TCP que no portan información del objeto.
- 2) Solicitud del objeto en el tercer segmento de apertura de la conexión TCP.
 - No siempre es así.

Lo representado en la figura acontece después que se ha descargado el fichero HTML base.

HTTP

17

Conexión TCP persistente CON CANALIZACIÓN

El tiempo total de descarga para "N" objetos a descargar, todos de igual tamaño y un grado de canalización "N", es:

$$T_{T-PCC} \approx (RTT + RTT + X_p) = 2RTT + X_p \lll T_{T-PSC} = RTT + N.(RTT + X_p)$$

Supuestos:

- 1) Despreciables los tiempos de:
 - Procesamiento en navegador y servidor Web.
 - Transmisión de los segmentos TCP que no portan información del objeto.
- 2) Solicitud del objeto en el tercer segmento de apertura de la conexión TCP.
 - No siempre es así.

Lo representado en la figura acontece después que se ha descargado el fichero HTML base, y conlleva a compartir "simultáneamente" el ancho de banda entre los "N" objetos.

Universidad de Alcalá

HTTP

18

Conexión NO Persistente	Conexión Persistente
<ul style="list-style-type: none"> □ HTTPv1.0 y HTTPv1.1. □ Sólo un objeto por conexión TCP. □ Modos: <ul style="list-style-type: none"> ▣ Serie y paralelo. □ Retardo $\approx 2.RTT^*$ por objeto □ Para reducir retardo total por página: <ul style="list-style-type: none"> ▣ Navegador abre varias conexiones en paralelo (modo paralelo). <ul style="list-style-type: none"> ■ Por defecto entre 5 y 10. ■ 1 → Modo serie. 	<ul style="list-style-type: none"> □ HTTPv1.1, por defecto. □ Múltiples objetos por conexión TCP □ Modos: <ul style="list-style-type: none"> ▣ Sin canalización y con canalización. □ Sin canalización. <ul style="list-style-type: none"> ▣ Retardo $\approx 1.RTT^*$ por objeto □ Con canalización. <ul style="list-style-type: none"> ▣ Retardo \approx de <u>algo más de un RTT*</u> para todos los objetos ▣ Modo por defecto.

*: excluyendo los tiempos de transmisión de cada objeto.

Universidad de Alcalá Tema 2: La capa de Aplicación

HTTP

19

- Tipos de mensajes
 - Métodos.
 - Solicitudes tipo ASCII.
 - Códigos de estado.
 - Respuestas tipo ASCII MIME.

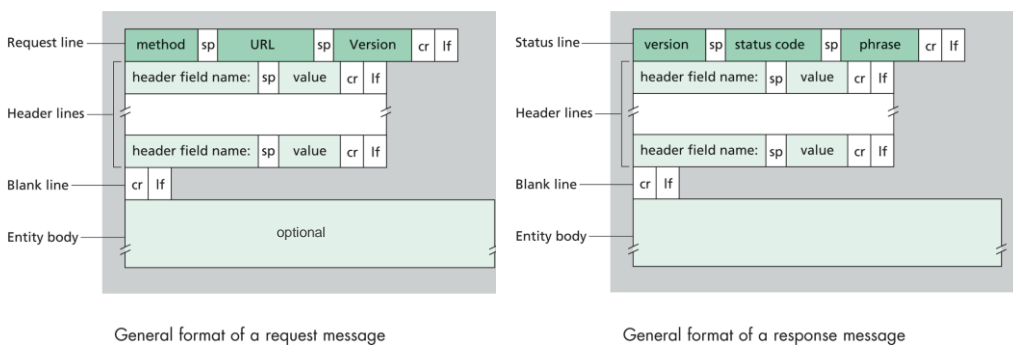
- Formato de mensajes.

Línea de inicio	← de solicitud o de respuesta.
Cabeceras	← de solicitud o de respuesta.
Línea en blanco	← como separador.
Cuerpo	← opcional.

HTTP

20

Formato general de mensajes HTTP de "solicitud" y de "respuesta"



HTTP

21

- Más sobre mensajes HTTP
 - Cabeceras de solicitud.
 - Dan información del navegador y eventualmente del usuario.
 - Cabeceras de respuesta.
 - Dan información del servidor y del "recurso".
 - Cuerpo.
 - En mensajes de "solicitud":
 - Vacío, o
 - Parámetros relativos al "recurso" solicitado.
 - En mensajes de "respuesta":
 - "Recurso" solicitado o
 - Texto relativo a un error.

HTTP

22

- Métodos (solicitudes) más significativos
 - **GET**: para descargar página/objeto (obtener información).
 - Mensaje de solicitud más utilizado.
 - Si es el caso, envío de parámetros conjuntamente con el URL.
 - **POST**: para descargar objeto (enviar información).
 - Envío de parámetros/información en el cuerpo del mensaje. P.e, para completar un formulario.
 - Lo que devuelve el servidor depende del contenido enviado.
 - **HEAD**: similar a GET, pero para descargar sólo el encabezado de la página/objeto.
 - Para supervisión y depuración. P.e, última modificación de un objeto, comprobar enlaces, etc.
 - **PUT**: para cargar nueva página/objeto.
 - **DELETE**: para eliminar página/objeto.
 - **OPTIONS**: para consultar opciones.
 - P.e, propiedades del servidor o de algún recurso.

HTTP

23

□ Códigos de estado (respuestas)

- Código de tres caracteres numéricos.
- Primer carácter indica **tipo** de código de estado.
- Tipos:
 - 1xx: información, respuesta provisional (100–199).
 - 2xx: éxito en la resolución de la solicitud (200-299).
 - 3xx: redirigir solicitud, necesarias más acciones (300-399).
 - 4xx: error del cliente (400-499).
 - 5xx: error del servidor (500-599).
- Ejemplos en el libro de texto.
- Más información en la RFC 2616.

HTTP

24

□ Cabeceras

¿Qué cabeceras se utilizan en los mensajes HTTP? Depende:

- Por parte del navegador, en los métodos:
 - Tipo de mensaje.
 - Versión del protocolo.
 - Configuración hecha por el usuario (p.e, idioma preferido).
 - Tipo y versión del navegador.
 - Disponibilidad en caché del objeto.
- Por parte del servidor, en los códigos de estado:
 - Tipo de mensaje.
 - Versión del protocolo.
 - Configuración del servidor.
 - Información relativa al recurso (si es el acaso).
 - Cómo tratar el recurso (caché, no caché, cookie, no cookie, etc.).

HTTP

25

Algunas cabeceras comunes de mensajes de SOLICITUD/RESPUESTA

Cabecera	Contenido
Date	Fecha y hora en que se envía el mensaje
Connection	Indica en relación con la conexión TCP cerrar o mantener

Algunas cabeceras de mensajes de SOLICITUD

Cabecera	Contenido
User-Agent	Información del Navegador y su plataforma
Accept	Tipos de contenidos que el Cliente puede aceptar
Accept-Charset	Conjunto de caracteres que el Cliente puede aceptar
Accept-Encoding	Codificaciones que el Cliente puede aceptar
Accept-Language	Idiomas que el Cliente puede manejar
Host	Nombre de dominio del Servidor
Authorization	Credenciales del Cliente (nombre de usuario y clave)
Cookie	Notificación de "cookie" previamente establecida por un Servidor

HTTP

26

Algunas cabeceras de mensajes de RESPUESTA

Cabecera	Contenido
Server	Información del Servidor
Content-Encoding*	Tipo de codificación del contenido
Content-Language	Idioma utilizado por el público al que va destinado el contenido
Content-Length*	Nº de octetos del contenido
Content-Type*	Tipo MIME del contenido
Last-Modified	Fecha y hora de la última modificación del recurso
Location	Indica al Cliente dónde enviar solicitud para el recurso solicitado
Set-Cookie	Cookie que el Servidor envía al Cliente
Authenticate	Indicación al Cliente que se requiere autenticación

*: también pueden estar en cabeceras de mensajes de solicitud.

HTTP

27

- Ejemplo: cliente solicita el objeto "doc1.html".

```
GET /doc1.html HTTP/1.0
Accept: text/html
Accept: image/gif
User-Agent: Lynx/2.2 libwww/2.14
From: pepe@info.ula.es
/* esto es una línea en blanco */
```

- Método GET, fichero que se solicita y versión HTTP.
- Lista de tipos MIME que puede aceptar.
- Identificación del SW de la máquina cliente (normalmente SO y navegador).
 - Permite al servidor optimizar ficheros para tipo particular de navegador.
- Dirección de correo electrónico del usuario.
- Línea en blanco, final de cabeceras HTTP.

HTTP

28

- Ejemplo: servidor responde.

```
HTTP/1.0 200 OK
Date: Friday, 23-Feb-01 16:30:00 GMT
Server: Apache/1.1.1
Content-type: text/html
Content-length: 230
/* esto es una línea en blanco */
<HTML><HEAD><TITLE> ..... </HTML>
```

- Versión 1.0 de HTTP.
- Código de estado 200 → Solicitud procesada satisfactoriamente.
- Fecha y hora en la que se envía el mensaje.
- Tipo de servidor: Apache.
- Tipo de contenido: texto en formato HTML.
- Longitud en octetos del contenido: 230.
- Línea en blanco, final de cabeceras HTTP.
- Cuerpo del mensaje con el recurso.