

# Redes

## Tema 2: Capa de aplicación

José Carlos Cabaleiro Domínguez

Escola Técnica Superior de Enxeñaría

# Índice

- 1 Introducción
- 2 Protocolo de transferencia de hipertexto
- 3 Protocolo de transferencia de archivos
- 4 Protocolos de correo electrónico
- 5 DNS: Servicio de nombres de dominio
- 6 Distribución de contenidos

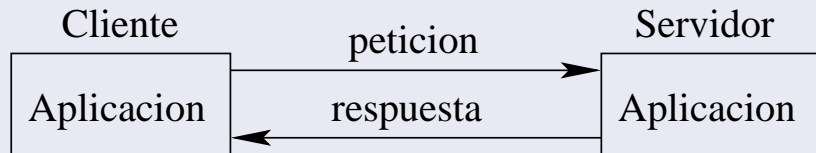
# Índice

- 1 **Introducción**
- 2 Protocolo de transferencia de hipertexto
- 3 Protocolo de transferencia de archivos
- 4 Protocolos de correo electrónico
- 5 DNS: Servicio de nombres de dominio
- 6 Distribución de contenidos

# Introducción

## Capa de aplicación

- Se ocupa de la comunicación entre procesos



# Capa de aplicación

## Protocolos de comunicación

- Necesarios para la comprensión de los mensajes entre los procesos
- Facilitan la programación de las funciones de envío y recepción

## Deben especificar

- Tipo de mensajes que se intercambian: petición, respuesta, etc.
- Reglas que especifican cuándo y cómo se envían los mensajes
- Sintaxis del mensaje: campos
- Semántica de cada campo

# Capa de aplicación

## Servicios que necesita la aplicación de red

- Los protocolos de la capa de aplicación usan protocolos básicos de la capa de transporte
- Transferencia fiable de datos o no fiable
  - TCP proporciona un servicio fiable y orientado a conexión
  - UDP un servicio no fiable más sencillo pero rápido y no orientado a conexión
- Ancho de banda
- Temporización

# Capa de aplicación

## Protocolos a tratar

TCP	HTTP (web) SMTP, POP3 e IMAP (correo) FTP (ficheros)
UDP	DNS (traducciones)

## Agente de usuario

- Interfaz entre el usuario y la aplicación (navegador, gestor de correo, etc.)

# Índice

- 1 Introducción
- 2 Protocolo de transferencia de hipertexto**
- 3 Protocolo de transferencia de archivos
- 4 Protocolos de correo electrónico
- 5 DNS: Servicio de nombres de dominio
- 6 Distribución de contenidos



# HTTP

## Protocolo de transferencia de hipertexto

- Protocolo que define la comunicación entre un servidor web y un cliente web
- Usa TCP
- Es un protocolo sin estado (a veces mecanismos como cookies)
- Usa por defecto el puerto 80

# HTTP

## Conceptos

- Página web (o documento): consta de objetos (archivos direccionables por un URL)
- Las páginas web: archivo HTML base (contiene la distribución y, en general, el texto) y varios objetos
- Navegador: agente de usuario para la web
- Servidor web: alberga objetos
- Solicitud de una página web  $\implies$  el servidor devuelve el documento base

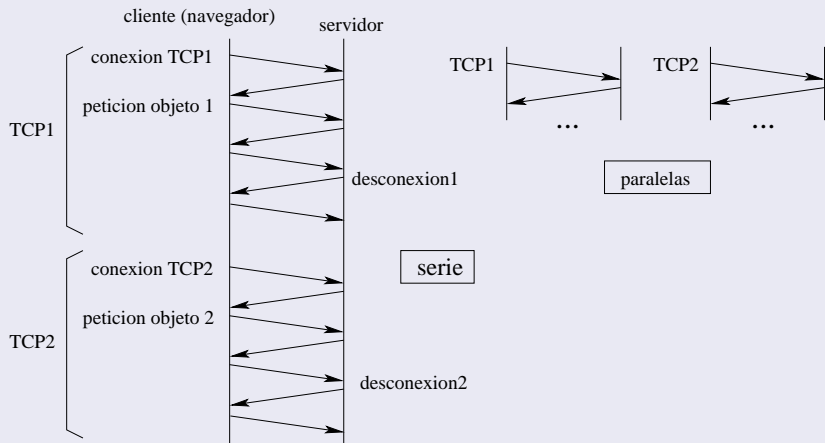
# HTTP

## Conexiones HTTP

- No persistentes: se usa una conexión TCP distinta para transferir cada objeto (HTTP/1.0)
  - Serie: se espera a que acabe la conexión TCP previa
  - Paralelo: se inician varias conexiones TCP a la vez
- Persistentes: se pueden transferir varios objetos (e incluso varias páginas) con la misma conexión TCP
  - Sin entubamiento: el cliente pide un nuevo objeto cuando se ha recibido el previo
  - Con entubamiento: el cliente puede hacer peticiones de varios objetos antes de recibir los anteriores (modo por defecto en HTTP/1.1)

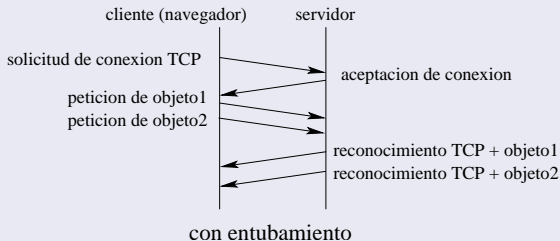
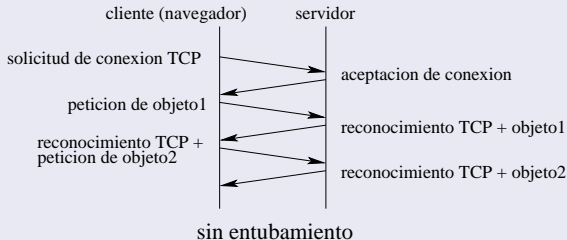
# HTTP

## HTTP no persistente



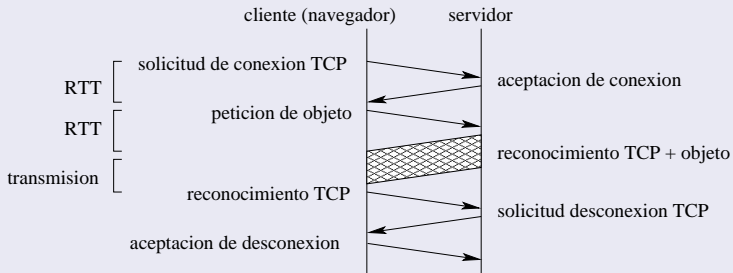
# HTTP

## HTTP persistente



# Tiempo de transferencia de una página web

## Conexiones no persistentes y serie



# Tiempo de transferencia de una página web

## Parámetros

- RTT (tiempo de ida y vuelta): tiempo necesario para que un paquete pequeño vaya del cliente al servidor y vuelva al cliente
- Tiempo de transmisión del archivo: depende del tamaño del archivo

## Todos los tipos de conexiones

- El primer objeto se transfiere en  $2RTT + t_{transmission}$
- Los siguientes objetos dependen del tipo de conexión

# HTTP

## Mensajes HTTP

Dos tipos:

- Petición: para pedir objetos
- Respuesta: contienen los objetos

Ambos mensajes tienen:

- Cabecera: información de control, en ASCII de 7 bits
- Cuerpo: los datos (un objeto, contenido de un formulario, etc.) y son en binario



# Mensajes HTTP

## Mensajes de petición

cabecera	línea de petición	metodo	sp	URL	sp	version	cr	lf	
		nombre campo cabecera				sp	valor	cr	lf
	líneas de cabecera								
		nombre campo cabecera				sp	valor	cr	lf
cuerpo	línea en blanco	cr	lf						
		cuerpo							

# Mensajes HTTP

## Mensajes de petición

- Línea de petición: objeto solicitado
  - Método: GET (página normal) POST (formulario)
- Líneas de cabecera: opciones

Host:	nombre servidor web (puede haber varios para una IP)
Connection:	close (si queremos conexiones no persistentes)
User-agent:	mozilla/4.0 (el navegador)
Accept-lenguaje:	es (el idioma preferido de la página, si la hay)

- Cuerpo
  - GET: vacío
  - POST: los datos del formulario

# Mensajes HTTP

## Mensajes de respuesta

cabecera	{	línea de estado	version	sp	cod. estado	sp	frase	cr	If
		líneas de cabecera	nombre campo cabecera			sp	valor	cr	If
			nombre campo cabecera			sp	valor	cr	If
cuerpo	{	línea en blanco	cr	If					
		objeto							

# Mensajes HTTP

## Mensajes de respuesta

- Línea de estado

200	OK (el objeto o página se sirve sin problemas)
404	Not found (la página no existe)
400	Bad Request (no se entendió el formato de la petición)

- Líneas de cabecera: opciones

Connection:	close (indica que se usan conexiones no persistentes)
Date:	fecha de envío de la página
Server:	Apache/1.3.0 (el servidor web)
Last-Modified:	cuando se creó o modificó por última vez la página
Content-Length:	tamaño en bytes del objeto, página o imagen
Content-Type:	text/html, image/gif, image/jpeg

- Cuerpo: los datos

# HTTP/2

## Nueva versión del protocolo HTTP

- Basado en el protocolo SPDY, desarrollado por Google (64 % más rápido)
- Especificación publicada en mayo de 2015
- Soportado por las últimas versiones de muchos navegadores

## Mejoras

- Multiplexación de peticiones HTTP sobre una misma conexión TCP
- Compresión de cabeceras
- Soporte para *server push*
- Pipelining de solicitud–respuesta
- Cifrado de la información: uso obligatorio de SSL

# Índice

- 1 Introducción
- 2 Protocolo de transferencia de hipertexto
- 3 Protocolo de transferencia de archivos**
- 4 Protocolos de correo electrónico
- 5 DNS: Servicio de nombres de dominio
- 6 Distribución de contenidos

# FTP

## Protocolo de transferencia de archivos

- Define la comunicación con un servidor de archivos
- Con estado: mantiene información durante toda la sesión
- Usa dos conexiones TCP paralelas (información de control fuera de banda)
  - Conexión de control
    - Usa el puerto 21
    - Para enviar los comandos: nombre y clave de usuario, *dir*, *put*, *get* y recibir las respuestas
    - Es persistente: dura todo el tiempo de la conexión FTP
    - Usa ASCII de 7 bits
  - Conexión de datos
    - Usa el puerto 20
    - Para transmitir datos en respuesta a los comandos
    - Es no persistente: se abre y se cierra una conexión nueva por cada archivo que se transmite
    - Usa ASCII de 7 bits

# FTP

## Comandos y respuestas FTP

- Por la conexión de control
- Comandos: constan de 4 caracteres en mayúsculas y campos adicionales acabadas en cr+lf

USER	nombre de usuario
PASS	clave
LIST	lista de archivos
RETR	nombre de fichero (traer fichero)
STOR	nombre de fichero (almacenar fichero)

- Respuestas: un código de 3 dígitos y una frase explicativa

331	Username OK, password required
125	Data connection already open; transfer starting
425	Can't open data connection
452	Error writing file

## Transmisión cifrada

- FTP, telnet y HTTP sin cifrar. SFTP, SSH y HTTPS no



# Índice

- 1 Introducción
- 2 Protocolo de transferencia de hipertexto
- 3 Protocolo de transferencia de archivos
- 4 Protocolos de correo electrónico**
- 5 DNS: Servicio de nombres de dominio
- 6 Distribución de contenidos

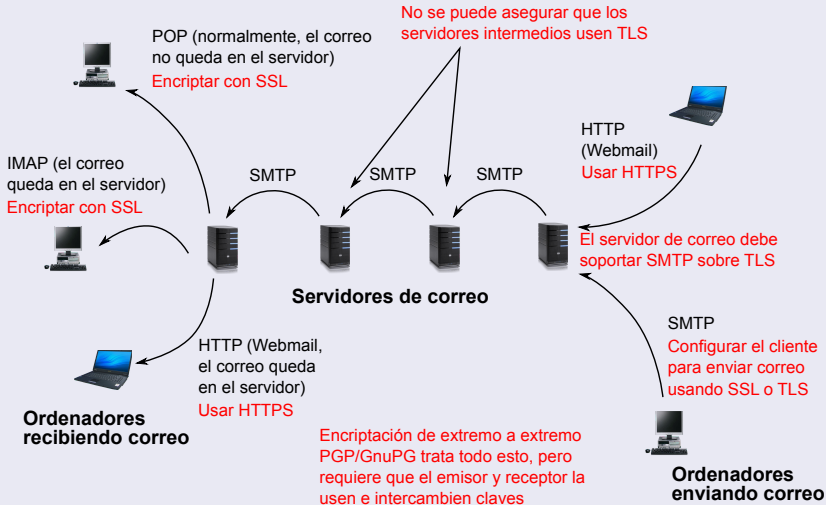
# Correo electrónico

## Protocolos de correo electrónico

- Para enviar correo al servidor de correo o entre servidores: SMTP, HTTP (si se usa un navegador)
- Acceso al correo por el agente de usuario: POP3, IMAP, HTTP (si se usa un navegador)

# Correo electrónico

## Protocolos de correo electrónico



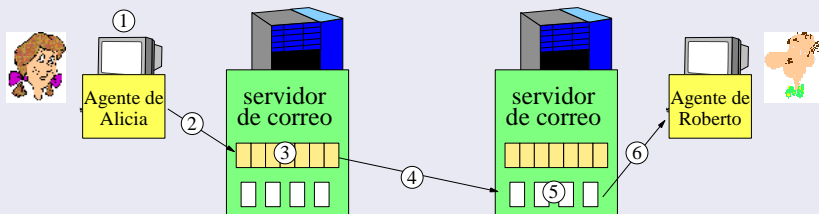
# SMTP

## Características del funcionamiento del correo electrónico

- Buzón de correo en el servidor
- El agente de usuario (AU) se comunica con el servidor
- Si el servidor destino está fuera de servicio, se reintenta (30 minutos)
- SMTP usa el puerto 25
- Usa conexiones TCP persistentes: mantiene abierta la conexión durante toda la sesión (varios mensajes)
- Emplea ASCII de 7 bits tanto para las cabeceras como para los cuerpos. Una imagen, el AU la transforma en ASCII

# SMTP

## Funcionamiento del correo electrónico



- 1 Alicia usa su AU para escribir un mensaje
- 2 El AU de Alicia envía el mensaje a su servidor de correo: se almacena en una cola de mensajes
- 3 El lado cliente de SMTP (servidor de Alicia) abre una conexión TCP con el servidor de Roberto
- 4 Tras una sincronización inicial (envío dirección destino y remite), el cliente envía el mensaje sobre la conexión TCP
- 5 El servidor de Roberto lo deposita en su buzón de correo
- 6 Roberto usa su AU para leer el mensaje

# SMTP

## Mensajes SMTP

Tres tipos de mensajes:

- Comandos: palabra en mayúscula seguida de los parámetros

HELO	nombre servidor
MAIL FROM:	dirección de correo del remitente
RCPT TO:	direccion de correo del destinatario
DATA	(el contenido del correo)
QUIT	

- Respuestas: código numérico seguido de una frase aclarativa

220	nombre del servidor
250	comando que se ejecutó satisfactoriamente
354	envíe el correo, terminando en .
221	cierre de conexión

- Datos: el contenido de los correos: todos los objetos encapsulados en un único archivo codificado en ASCII de 7 bits (con las llamadas extensiones MIME)

# SMTP

## Ejemplo de transacción típica entre cliente (C) y servidor (S) de correo

```
S: 220 smtp.usc.es SMTP USC Server
C: HELO smtp.usc.es
S: 250 smtp.usc.es
C: MAIL FROM: pepito@usc.es
S: 250 Ok
C: RCPT TO: juanito@yahoo.es
S: 250 Ok
C: DATA
S: 354 Enter data, end data with <CR><LF>.<CR><LF>
C: Hola
C: Esto es una prueba, adios
C: .
S: 250 Ok: message queued as C5F0B13210D
C: quit
S: 221 Bye
```

# SMTP

## Protocolo inseguro

- No se pide nombre y clave de usuario: permite enviar correos a cualquiera

## Comparación con HTTP

- Ambos transmiten archivos y usan conexiones TCP persistentes
- HTTP es un protocolo de demanda (el cliente demanda el archivo) y SMTP de oferta (el cliente oferta el archivo)
- SMTP requiere que la cabecera y el cuerpo sean de ASCII de 7. En HTTP el cuerpo puede contener datos binarios
- HTTP envía objetos en archivos diferentes, SMTP van encapsulados en el mismo archivo que contiene el texto
- HTTP no tiene estado y SMTP debe recordar la fase de la sesión en la que se encuentra



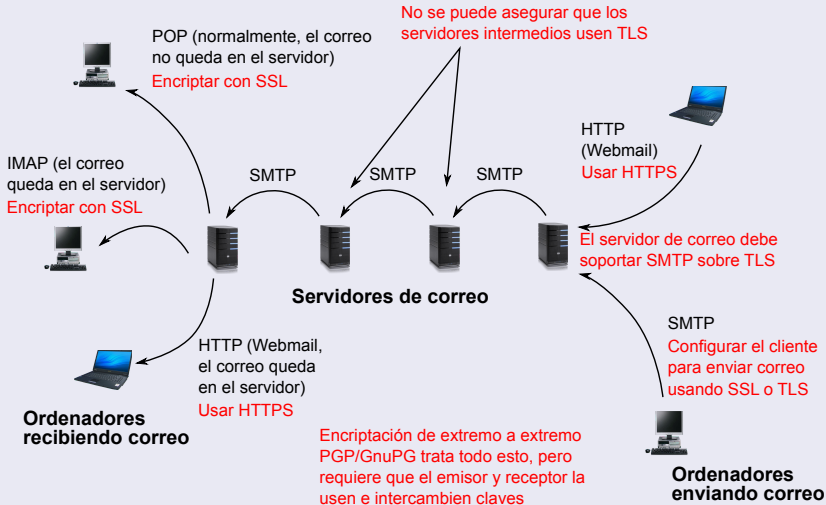
# Correo electrónico

## Protocolos de acceso al correo

- Los agentes de usuarios para transferir el correo del buzón destino al ordenador local
- No se usa SMTP porque es un protocolo de oferta  $\Rightarrow$  ordenador local siempre encendido esperando correos
- Se usan protocolos que permiten descargar los correos: POP3, IMAP y HTTP

# Correo electrónico

## Protocolos de correo electrónico



# POP3: Protocolo de oficina postal, versión 3

## POP3

- Protocolo sencillo para acceso al correo
  - Descargar y borrar del servidor
  - Descargar y mantener en el servidor
- Conexión TCP con el puerto 110 del servidor

## Protocolo simple: tres fases

- Autorización: autenticación del usuario
- Transacción
  - Recuperar los mensajes
  - Marcar/desmarcar para borrado
  - Estadísticas de correo
- Actualización: borrado de los mensajes marcados
  - Termina la sesión POP3

# POP3: Protocolo de oficina postal, versión 3

## Mensajes POP3

Tres tipos de mensajes:

- Comandos: palabra de 4 caracteres seguida de los parámetros

user	nombre del usuario
pass	palabra clave
list	
retr	número de correo (traer correo)
dele	número de correo (borrar correo)
quit	

- Respuestas: básicamente de dos tipos

+OK frase explicativa
-ERR frase explicativa

- Datos: datos de la respuesta
  - La lista de mensajes, los contenidos de los correos, etc.
  - El correo completo en un único mensaje (SMTP generaba un mensaje por cada línea de cabecera)

# POP3: Protocolo de oficina postal, versión 3

## Otras características

- Usa conexiones persistentes: se usa la misma conexión TCP mientras dure la sesión en la que pueden descargarse varios correos
- POP3 se puede usar para descargar y borrar (retr+dele) los correos del servidor o para descargar y mantener en el servidor (retr)
- Necesita cuenta con contraseña
- No mantiene información de estado entre sesiones POP3
- Definido en el RFC 1939

# IMAP: Protocolo de acceso a mensajes de Internet

## Inconvenientes de POP3

- Acceso al correo en múltiples dispositivos
- No permite carpetas en el servidor

## IMAP

- Más funcionalidades que POP3  $\Rightarrow$  más complejo
- El servidor asocia cada mensaje con una carpeta
  - Cuando llega un mensaje va a la carpeta INBOX
  - Permite leer, borrar, mover a otra carpeta. . .
- Comandos para crear carpetas, realizar búsquedas, descargar partes del mensaje. . .
- Mantiene información de estado entre sesiones
- Definido en el RFC 3501

# Índice

- 1 Introducción
- 2 Protocolo de transferencia de hipertexto
- 3 Protocolo de transferencia de archivos
- 4 Protocolos de correo electrónico
- 5 DNS: Servicio de nombres de dominio**
- 6 Distribución de contenidos

# DNS

## Servicio de nombres de dominio

- Para traducir nombres de hosts  $\rightleftarrows$  direcciones IP

## Componentes

- Numerosos servidores de nombres distribuidos por Internet
- La base de datos del DNS está distribuida de forma jerárquica
- Un protocolo que permite a los hosts pedir traducciones a los servidores y que los servidores se intercambien datos entre ellos

## Protocolo sin conexión

- DNS usa UDP, el puerto 53 y no tiene estado



# DNS

## Servicios proporcionados

- Traducir nombres de hosts  $\Leftrightarrow$  direcciones IP y obtención de alias
- Informa de los servidores autorizados para un dominio
- Alias de servidores de correo: permite simplificar las direcciones
- Distribución de la carga
  - Se asigna varias IPs a un nombre de host
  - Servidores espejo: con el mismo contenido
  - Devuelve de forma cíclica una IP del conjunto de las asignadas

# DNS

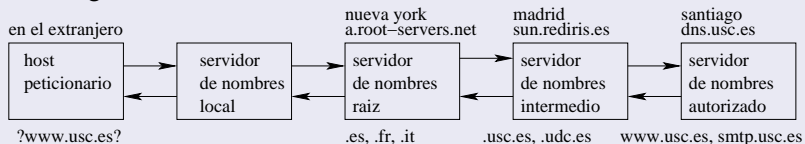
## Tipos de servidores

- Servidores locales: atienden las consultas de los hosts
- Servidores autorizados o autoritativos
  - Para que un host sea accesible en Internet, debe estar registrado en un servidor autorizado
  - Normalmente pertenecen al ISP
  - Cada host debe estar en dos servidores autorizados por fiabilidad
  - Muchos servidores autorizados se comportan como locales
- Servidores raíz: sobre 400 en Internet, gestionados por 13 organizaciones
  - Información de los dominios de primer nivel (TLD)
- Servidores intermedios o TLD: información sobre los niveles intermedios

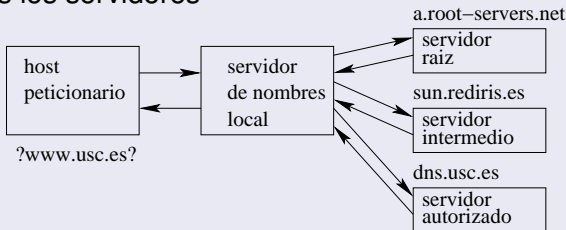
# DNS

## Tipos de consultas

- Consultas recursivas: cada servidor DNS interroga al siguiente



- Consultas iterativas: el servidor DNS local contacta con todos los servidores



# DNS

## Caché de DNS

- Los servidores almacenan copias locales de las correspondencias que obtienen
- Se borran después de cierto tiempo sin usar (dos días)
- En todos los niveles de la jerarquía, incluso en los hosts locales

# DNS

## Mensajes DNS

### Consultas y respuestas:

- Cabecera con información de control
  - Identificación: 16 bits que identifican la consulta (y su correspondiente respuesta)
  - Señales: 4 bits que indican si consulta o respuesta, tipo de consulta, etc.
  - Tamaño de los campos del cuerpo
- Cuerpo: 4 campos
  - Cuestiones: una o varias preguntas. Nombre o dirección a traducir, etc
  - Respuestas: una o varias respuestas. Pareja host/IP
  - Servidores autorizados. Permite hacer una cadena de consultas
  - Información adicional

# DNS

## Mensajes DNS

cabecera	identificacion	senales
	num. cuestiones	num. respuestas
	num. s. autorizados	num. inf. adicional
cuerpo	cuestiones	
	respuestas	
	servidores autorizados	
	informacion adicional	

Ejemplo:

1 1  
2 2

www.usc.es?

www.usc.es -> 193.144.74.224

usc.es -> dns.usc.es, dns2.usc.es

direcciones IP de los servidores autorizados

# DNS

## Mensajes DNS

Esos campos contienen registros de recursos: 4-tupla

Tipo	Nombre	Valor	TTL
A	nombre de host	dirección IP	
NS	dominio	servidor autorizado para el dominio	
CNAME	alias	nombre de host	
MX	alias de correo	servidor de correo	
AAAA	nombre de host	dirección IPv6	

TTL: tiempo durante el cual es válida la respuesta

# Índice

- 1 Introducción
- 2 Protocolo de transferencia de hipertexto
- 3 Protocolo de transferencia de archivos
- 4 Protocolos de correo electrónico
- 5 DNS: Servicio de nombres de dominio
- 6 Distribución de contenidos



# Distribución de contenidos

## Proporcionar contenidos

El acceso a servidores centralizados puede ser lento:

- El camino de los mensajes es lento o está congestionado
- El servidor está sobrecargado

⇒ se distribuyen (y duplican) los contenidos en distintas zonas y se dirigen las peticiones al servidor de menor tiempo de respuesta

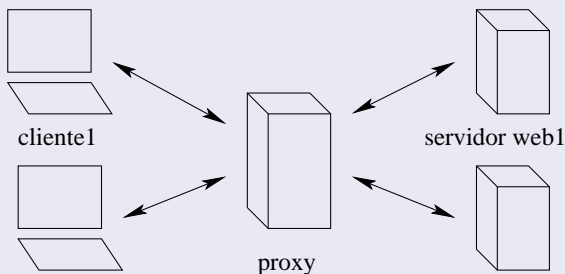
## Métodos

Método de distribución	Quién proporciona la infraestructura
Caché web (o servidor proxy)	ISP
Redes de distribución de contenidos	distribuidor de contenidos
Redes P2P (de igual a igual)	usuario

# Distribución de contenidos

## Caché web o servidor proxy

- Un servidor intermedio (proxy) por el que pasan todas las peticiones web de los hosts de una red
- Mantiene copias de los contenidos durante cierto tiempo
- El usuario debe configurar el navegador
- Permite un esquema jerárquico
- Proporcionado por el ISP



# Distribución de contenidos

## Redes de distribución de contenidos (CDN)

- Empresas que poseen *centros de host de Internet* (compañía CDN) alquilan su infraestructura
- La CDN replica los contenidos de sus clientes en los servidores CDN y los mantiene actualizados
- La CDN proporciona un mecanismo para que el contenido sea entregado por el servidor CDN que pueda hacerlo más rápidamente
- Acceso a los contenidos mediante dos técnicas:
  - Redirección de objetos
  - Balanceo de las peticiones usando el DNS

# Distribución de contenidos

## Redes P2P

- Todos los usuarios son a la vez servidores y clientes usando una aplicación
- Se necesita un nodo de arranque
- Se necesita construir un directorio

Tipo de directorio	Ejemplo
centralizado	napster
no centralizado	kaaza
inundación de consultas	gnutella

- BitTorrent
  - Cada archivo se comparte en una red P2P propia
  - Se necesita un *tracker* para cada red
  - No se necesita construir un directorio. La información de cada archivo va en los .torrent

