

Xarxes de Computadors

Tema 5 – Protocolos de nivel aplicació

Temario

- ▶ 1) Introducción
- ▶ 2) Redes IP
- ▶ 3) Protocolos UDP y TCP
- ▶ 4) Redes de área local (LAN)
- ▶ **5) Protocolos del nivel aplicación**

Tema 5 – Protocolos de nivel aplicación

- ▶ **Introducción**
- ▶ **Protocolos del nivel aplicaciones**
 - ▶ DNS
 - ▶ Email
 - ▶ Web
- ▶ **Otro**
 - ▶ HTML
 - ▶ XML
 - ▶ Charsets

Tema 5 – Protocolos de nivel aplicación

- ▶ **Introducción**
- ▶ Protocolos del nivel aplicaciones
 - ▶ DNS
 - ▶ Email
 - ▶ Web
- ▶ Otro
 - ▶ HTML
 - ▶ XML
 - ▶ Charsets

Tema 5 – Protocolos de nivel aplicación



► Objetivo de la capa aplicación

- Mecanismos y protocolos que proporcionan determinados servicios a la red o al usuario
- Proporcionar formatos estándares a los datos

Tema 5 – Protocolos de nivel aplicación

- ▶ Introducción
- ▶ **Protocolos del nivel aplicaciones**
 - ▶ DNS
 - ▶ Email
 - ▶ Web
- ▶ Otro
 - ▶ HTML
 - ▶ XML
 - ▶ Charsets

Tema 5 – Aplicaciones de red

- ▶ Se incluyen en esta capa aquellos protocolos que hacen funcionar las aplicaciones
- ▶ Sirvan para facilitar su funcionamiento, su aspecto de cara al usuario, su mantenimiento, etc.
- ▶ Por ejemplo
 - ▶ RIP es una aplicación de red ya que facilita la construcción de las tablas de encaminamiento y su mantenimiento en los routers
 - ▶ DHCP es una aplicación de red que permita la autoconfiguración de un host
 - ▶ DNS es un sistema que facilita los usuarios ya que pueden tratar los hosts con nombres en lugar de con números (@IP)

Tema 5 – Protocolos de nivel aplicación

- ▶ Introducción
- ▶ Protocolos del nivel aplicaciones
 - ▶ **DNS**
 - ▶ Email
 - ▶ Web
- ▶ Otro
 - ▶ HTML
 - ▶ XML
 - ▶ Charsets

Tema 5 – DNS

- ▶ Domain Name Server, RFC 1034 y 1035
- ▶ Objetivo:
 - ▶ Permitir al usuario el uso de nombres en lugar de @IP
 - ▶ Por ejemplo es más fácil acordarse
 - ▶ www.upc.edu que 147.83.2.135
 - ▶ mail.google.com que 64.233.184.109
- ▶ Como
 - ▶ manteniendo una base de datos distribuida y jerárquica de asociaciones @IP – nombre (llamados Resource Record, RR)
- ▶ Cada nombre en DNS consiste en un
 - ▶ Nombre del nodo (node name o hostname)
 - ▶ Nombre del dominio (domain name)
- ▶ ICANN
 - ▶ Internet Corporation for Assigned Names and Numbers
 - ▶ Es el organismo que gestiona DNS

Tema 5 – DNS

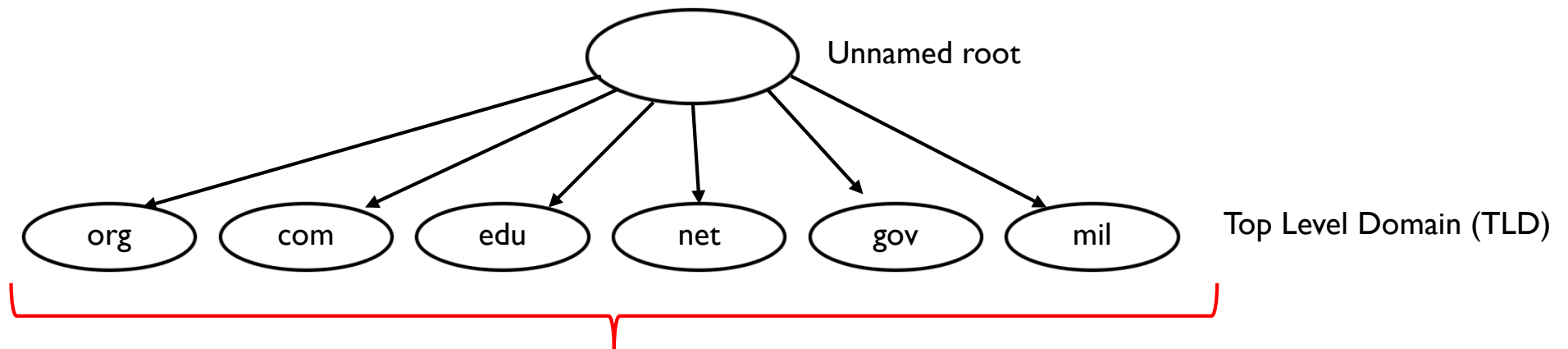
► Jerarquía de los dominios



La base de datos distribuida y jerárquica del DNS tiene estructura a árbol donde la raíz es el único punto que no tiene nombre

Tema 5 – DNS

► Jerarquía de los dominios

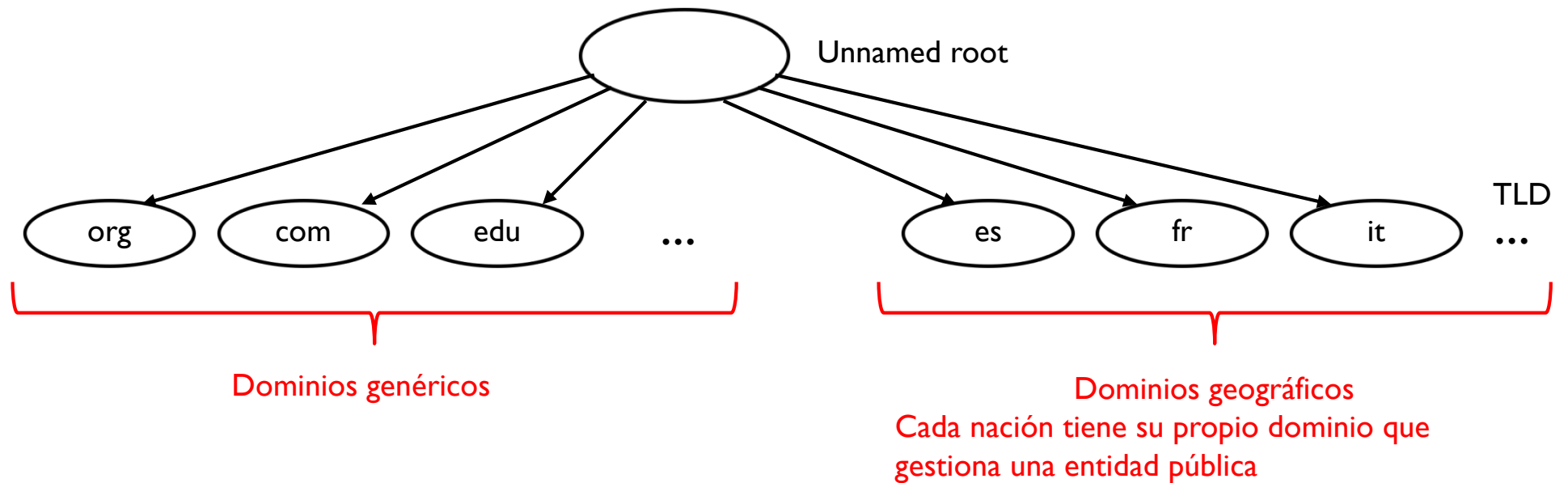


Dominios genéricos

Al principio se definieron 6 de estos dominios. A finales de los '90 empezaron luego a crearse algunos más como arts, info, store, and web, otros a principio del 2000 como aero, coop, museum, etc. Hoy en día hay miles.

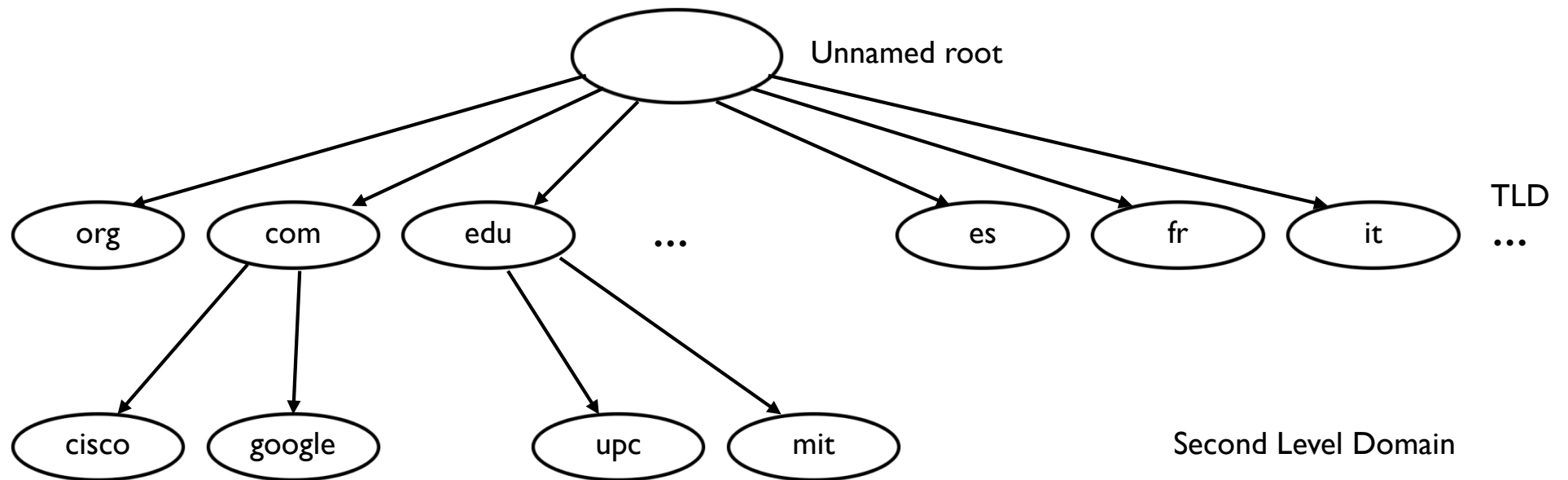
Tema 5 – DNS

► Jerarquía de los dominios



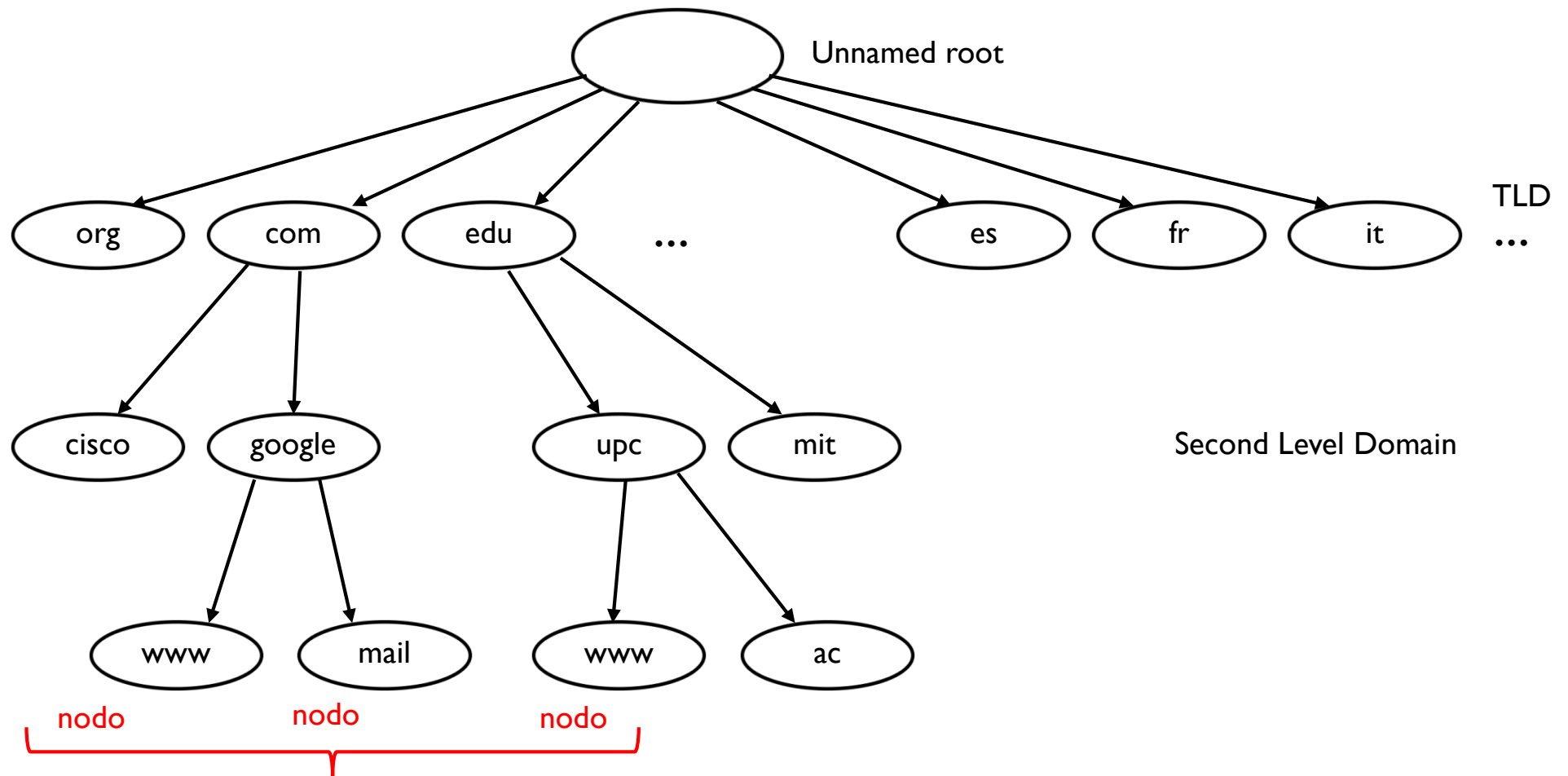
Tema 5 – DNS

► Jerarquía de los dominios



Tema 5 – DNS

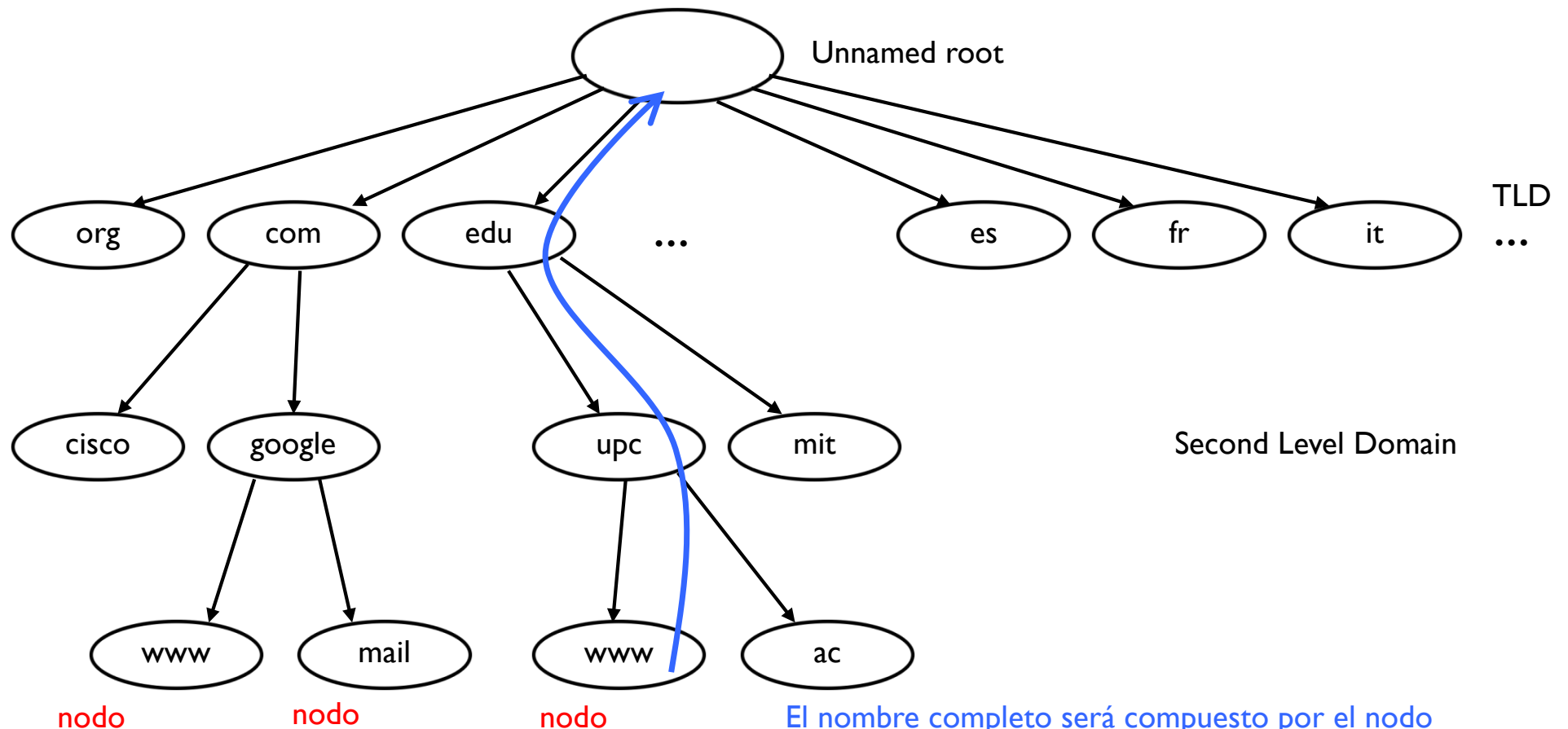
► Jerarquía de los dominios



No hay nada más abajo. Son las hojas del árbol y por lo tanto los nodos finales del DNS

Tema 5 – DNS

► Jerarquía de los dominios

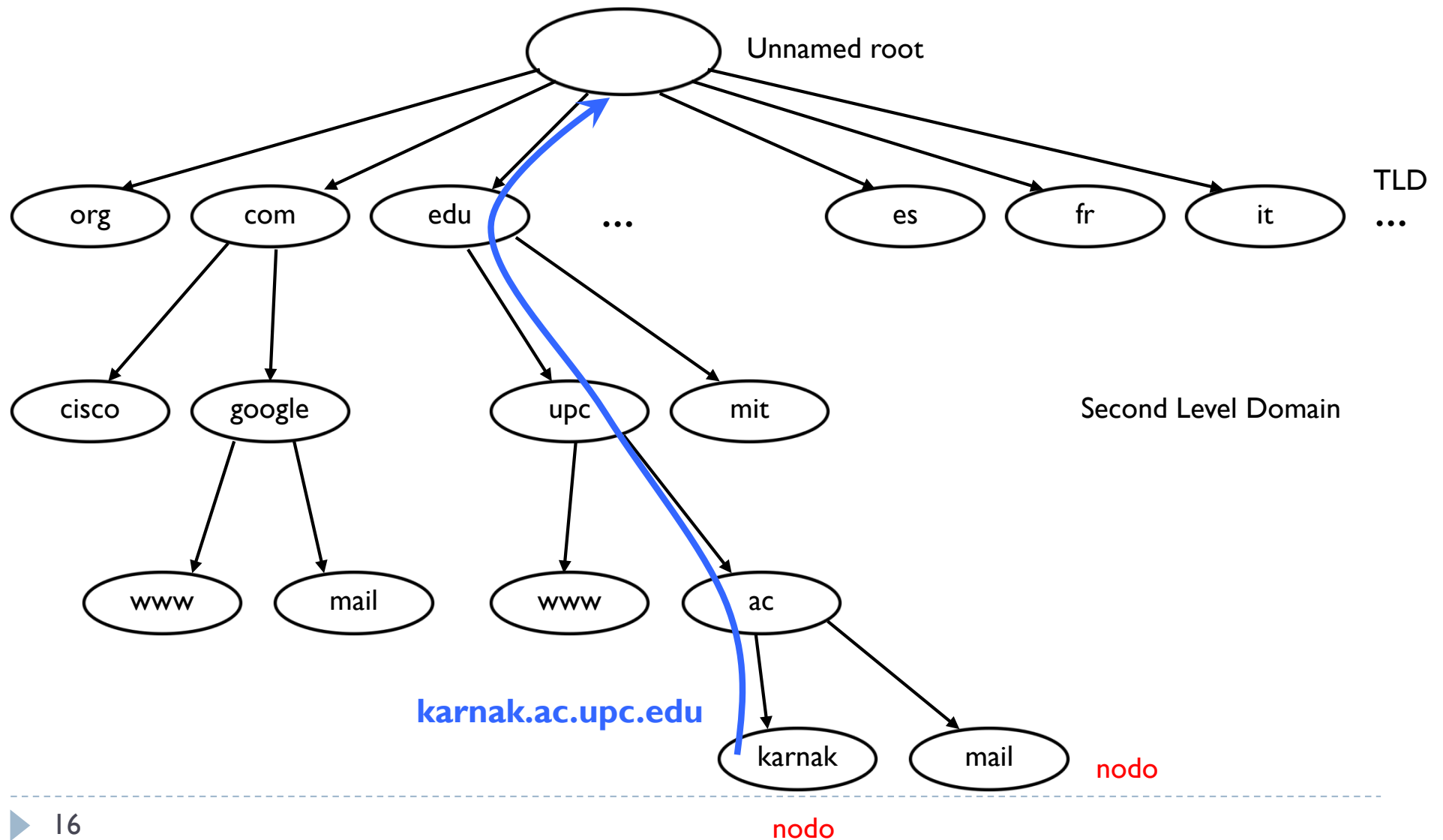


El nombre completo será compuesto por el nodo seguidos de recorrer el árbol al revés hasta el unnamed root y separando cada nivel con un punto

→ **www.upc.edu**

Tema 5 – DNS

► Jerarquía de los dominios

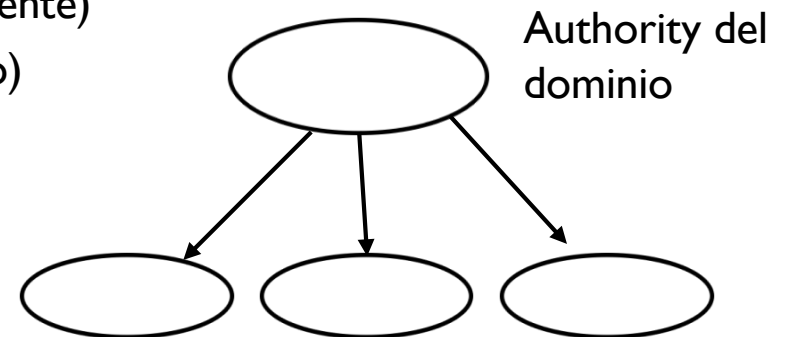


Tema 5 – DNS

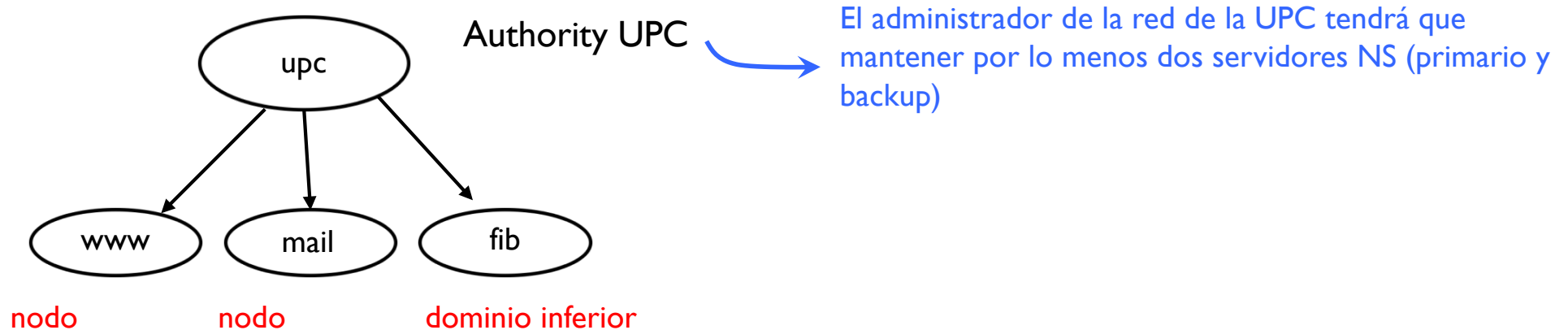
- ▶ No confundir
- ▶ Este es un nombre de un host, en particular el servidor de paginas web www de la UPC
 - ▶ www.upc.edu
- ▶ Este no es otro nombre de un host
 - ▶ www.upc.edu/agenda/
 - ▶ Es una página interna del mismo servidor de la UPC que se llama www.upc.edu
- ▶ No hay diferencias entre mayúsculas y minúsculas
 - ▶ www.upc.edu y WWW.UPC.EDU son los mismos nombres

Tema 5 – DNS

- ▶ Los Name Servers (NS) son los servidores que se ocupan de cada dominio (o zona)
 - ▶ Son las authorities del dominio
 - ▶ En cada dominio hay dos servidores NS
 - ▶ Servidor NS primario
 - ▶ Servidor NS de backup (se hace una copia del primario generalmente cada 3 h)
 - ▶ Estos servidores NS mantienen una parte de la base de datos, en concreto
 - ▶ Nombres y @IP de los nodos del dominio
 - ▶ Nombres y @IP de las authorities (los NS) del nivel inferior
 - ▶ Cada entrada de esta base de datos se llama Resource Record (RR)
 - Hay RR permanentes (se configuran manualmente)
 - Hay cached RR (se eliminan pasado un tiempo)



Tema 5 – DNS ejemplo



- ▶ Los NS del dominio UPC tendrán que mantener los RR del propio dominio
- ▶ Por un lado las asociaciones nombre de los nodos - @IP
 - ▶ El nodo www (servidor de paginas web de la UPC) y su @IP 147.83.2.135
 - ▶ El nodo mail (servidor de correos de la UPC) y su @IP 147.83.2.34
- ▶ Por el otro los nombre y @IP de las authorities del nivel inferior
 - ▶ El NS del subdominio citm y su @IP 147.83.5.55

Tema 5 – DNS ejemplo

- ▶ www.fib.upc.edu
- ▶ ¿cuál es el dominio?

Tema 5 – DNS ejemplo

- ▶ www.fib.upc.edu
- ▶ ¿cuál es el dominio?
fib.upc.edu
- ▶ ¿cuál es el nombre del nodo de este dominio?

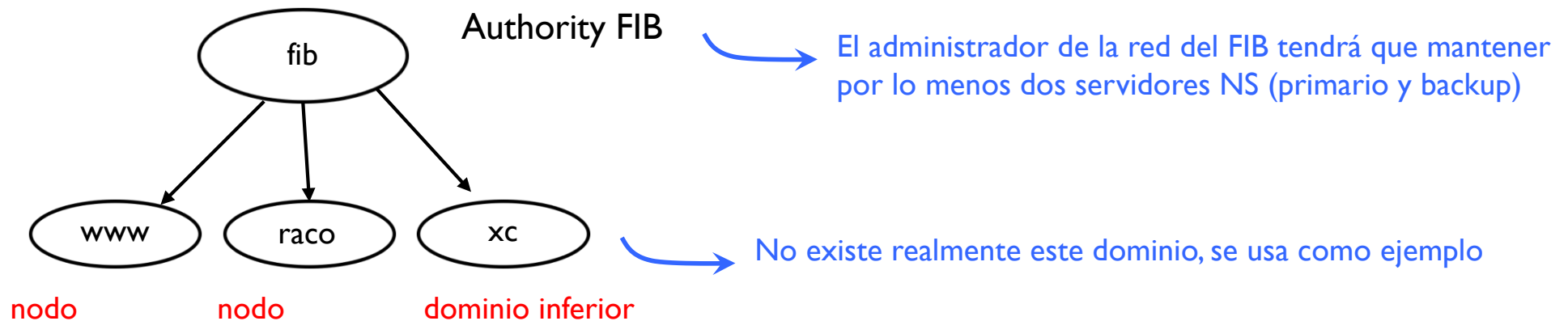
Tema 5 – DNS ejemplo

- ▶ www.fib.upc.edu
- ▶ ¿cuál es el dominio?
fib.upc.edu
- ▶ ¿cuál es el nombre del nodo de este dominio?
www
- ▶ ¿quién es el authority de este dominio?

Tema 5 – DNS ejemplo

- ▶ www.fib.upc.edu
- ▶ ¿cuál es el dominio?
fib.upc.edu
- ▶ ¿cuál es el nombre del nodo de este dominio?
www
- ▶ ¿quién es el authority de este dominio?
el servidor de nombres (NS) de la FIB

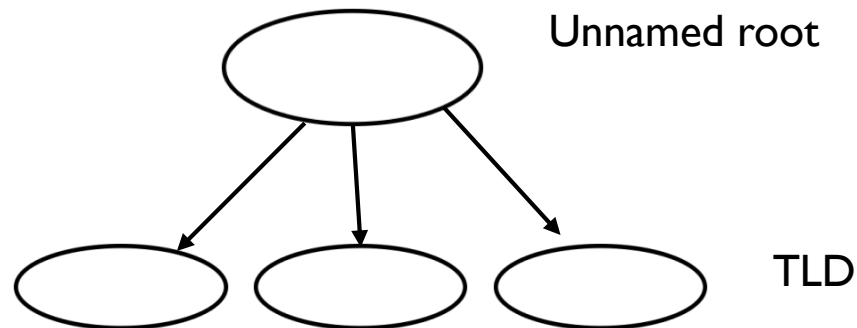
Tema 5 – DNS ejemplo



- ▶ Los NS del dominio FIB tendrán que mantener los RR del propio dominio
- ▶ Por un lado las asociaciones nombre de los nodos - @IP
 - ▶ El nodo www (servidor de paginas web de la UPC) y su @IP 147.83.53.43
 - ▶ El nodo raco (servidor del campus virtual) y su @IP 147.83.41.15
- ▶ Por el otro los nombre y @IP de las authorities del nivel inferior
 - ▶ El NS del subdominio xc y su @IP 147.83.55.59

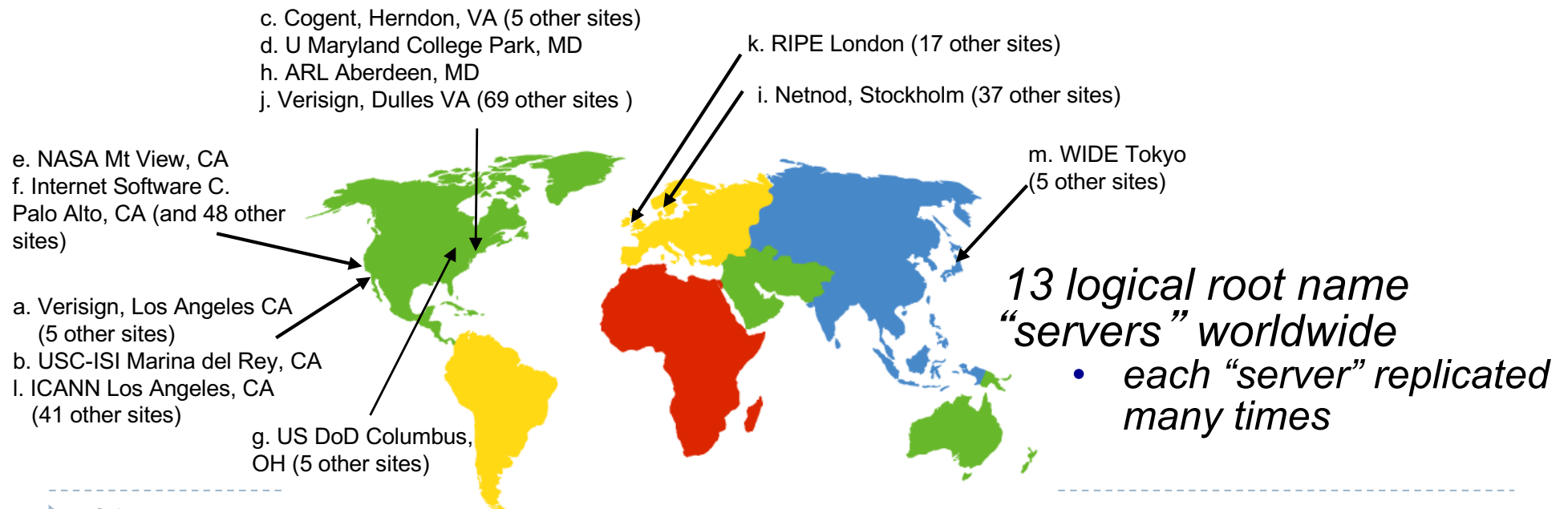
Tema 5 – DNS

- ▶ Los Name Servers (NS) de la authority de root se llaman Root Servers
- ▶ Hay 13 Root Servers (RS) primarios en el mundo
 - ▶ Se llaman A, B, C, D, ..., M
 - ▶ Hay varios backups
 - ▶ Tienes las @IP y los nombres de los NS del nivel TLD
 - ▶ <http://www.root-servers.org>



Tema 5 – DNS

- ▶ Los Name Servers (NS) de la authority de root se llaman Root Servers
- ▶ Hay 13 Root Servers (RS) primarios en el mundo
 - ▶ Se llaman A, B, C, D, ..., M
 - ▶ Hay varios backups
 - ▶ Tienes las @IP y los nombres de los NS del nivel TLD
 - ▶ <http://www.root-servers.org>



Tema 5 – DNS

- ▶ <http://www.root-servers.org>

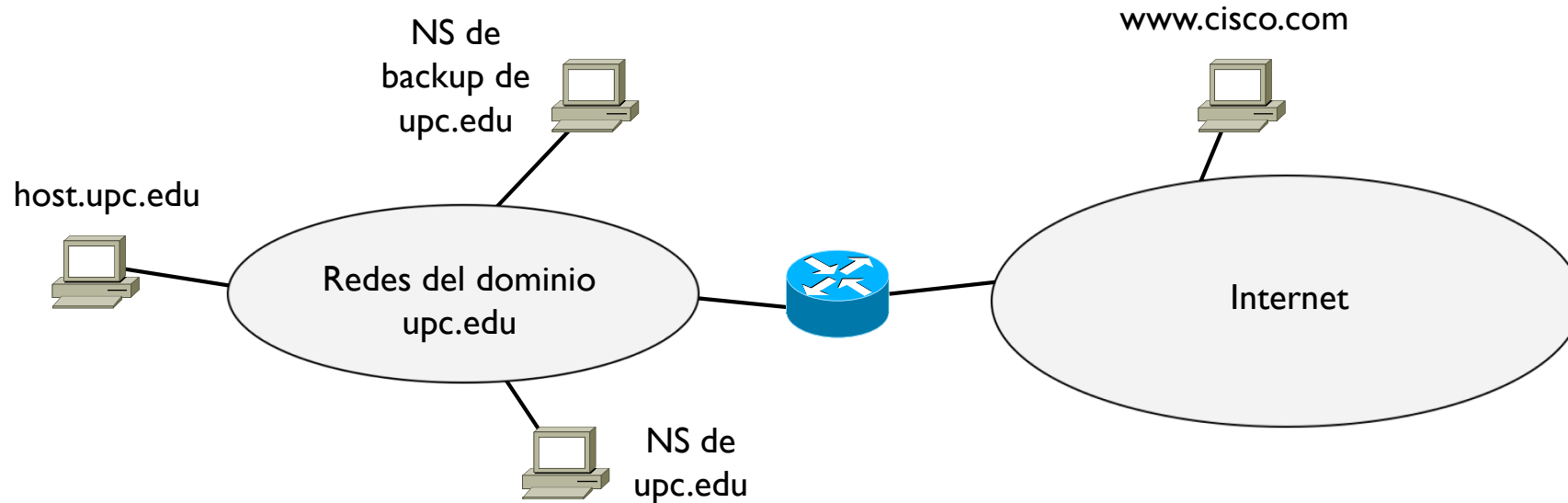
Hostname	IP Addresses	Manager
a.root-servers.net	198.41.0.4, 2001:503:ba3e::2:30	VeriSign, Inc.
b.root-servers.net	192.228.79.201, 2001:500:84::b	University of Southern California (ISI)
c.root-servers.net	192.33.4.12, 2001:500:2::c	Cogent Communications
d.root-servers.net	199.7.91.13, 2001:500:2d::d	University of Maryland
e.root-servers.net	192.203.230.10	NASA (Ames Research Center)
f.root-servers.net	192.5.5.241, 2001:500:2f::f	Internet Systems Consortium, Inc.
g.root-servers.net	192.112.36.4	US Department of Defense (NIC)
h.root-servers.net	198.97.190.53, 2001:500:1::53	US Army (Research Lab)
i.root-servers.net	192.36.148.17, 2001:7fe::53	Netnod
j.root-servers.net	192.58.128.30, 2001:503:c27::2:30	VeriSign, Inc.
k.root-servers.net	193.0.14.129, 2001:7fd::1	RIPE NCC
l.root-servers.net	199.7.83.42, 2001:500:3::42	ICANN
m.root-servers.net	202.12.27.33, 2001:dc3::35	WIDE Project



Tema 5 – DNS funcionamiento

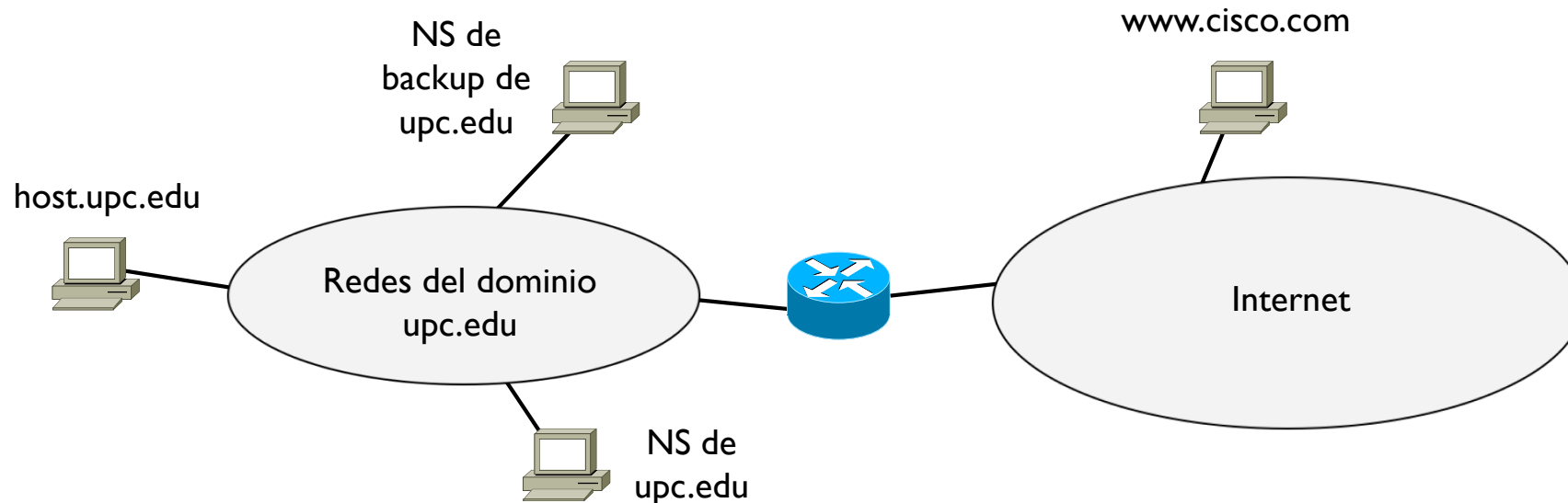
- ▶ Arquitectura cliente – servidor
- ▶ Servidor UDP puerto 53 (puerto conocido de TCP/IP)
 - ▶ Comunicación no fiable
- ▶ Nota: DNS usa TCP cuando se hacen backups entre el NS primario y el de backup

Tema 5 – DNS ejemplo



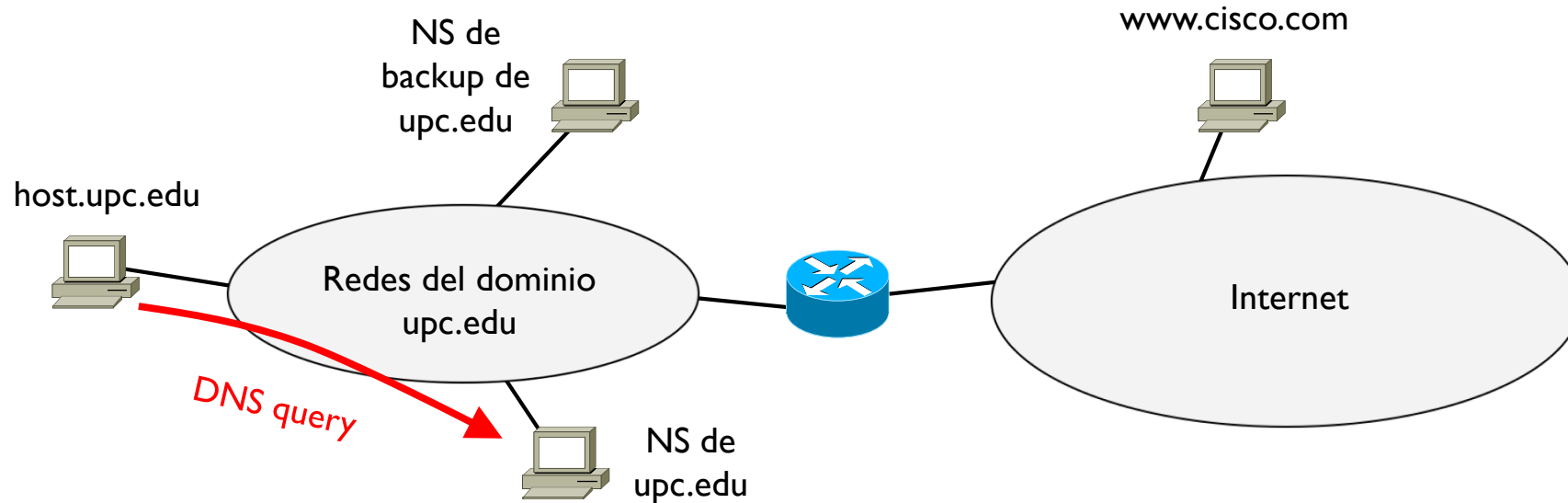
- ▶ El host host.upc.edu quiere ver la página web de www.cisco.com
 - ▶ www.cisco.com es un nombre mientras que las redes funcionan con @IP
 - ▶ El host tiene que encontrar la @IP de www.cisco.com
 - ▶ Se dice que se tiene que hacer una resolución de un nombre

Tema 5 – DNS ejemplo



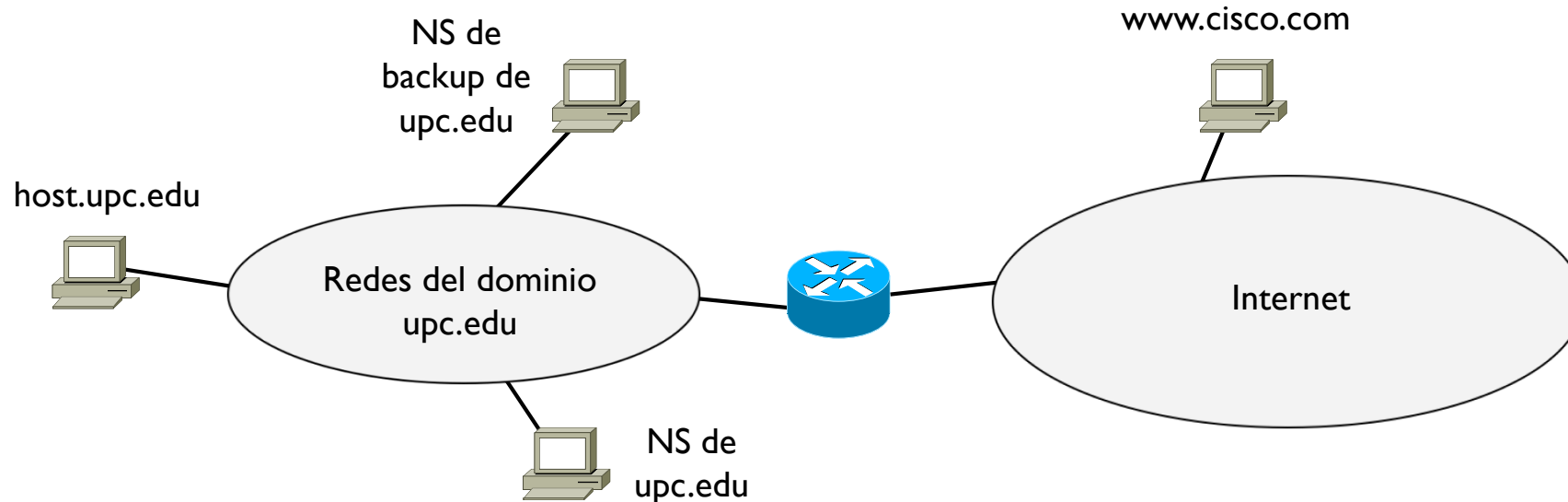
- ▶ El host comprueba si tiene una resolución ya disponible en su memoria cache del DNS
 - ▶ En efecto, cada vez que se hace una resolución DNS, los hosts mantienen esta resolución durante un cierto tiempo en una memoria llamada cache
 - ▶ Este tiempo es programable

Tema 5 – DNS ejemplo



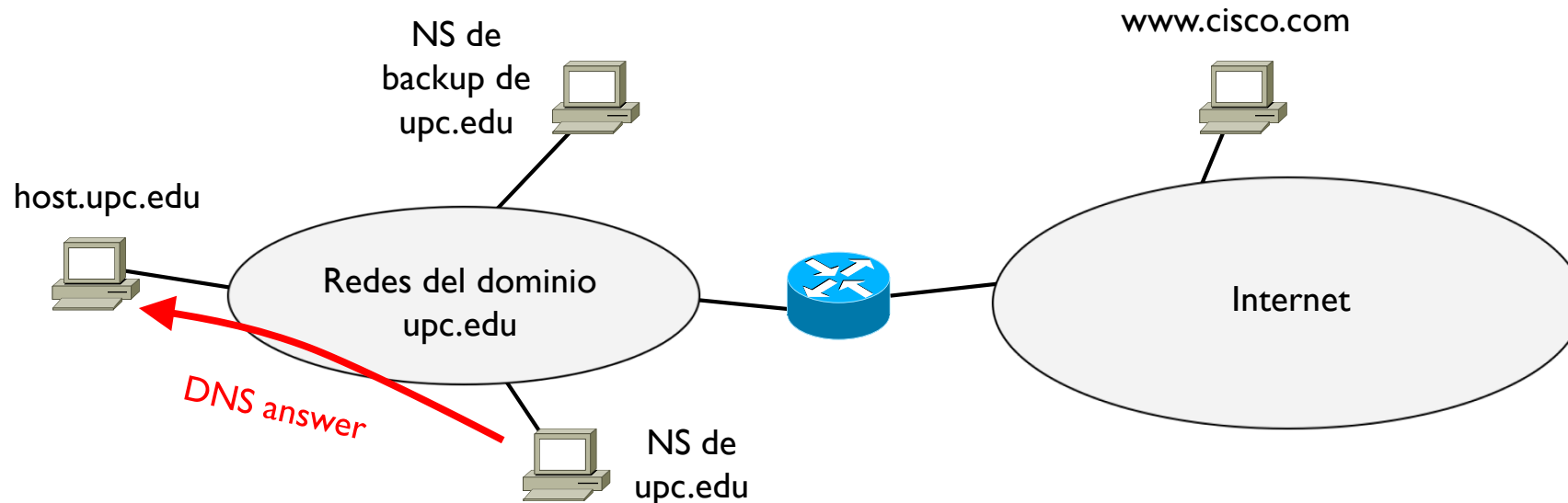
- ▶ Si no tiene esta resolución, contacta el authority (el NS primario) de su dominio
 - ▶ Envía al NS una petición de resolución (DNS query) con el nombre a resolver, es decir www.cisco.com
 - ▶ Se usa el protocolo UDP con puerto destino 53
 - ▶ El puerto origen es un número efímero

Tema 5 – DNS ejemplo



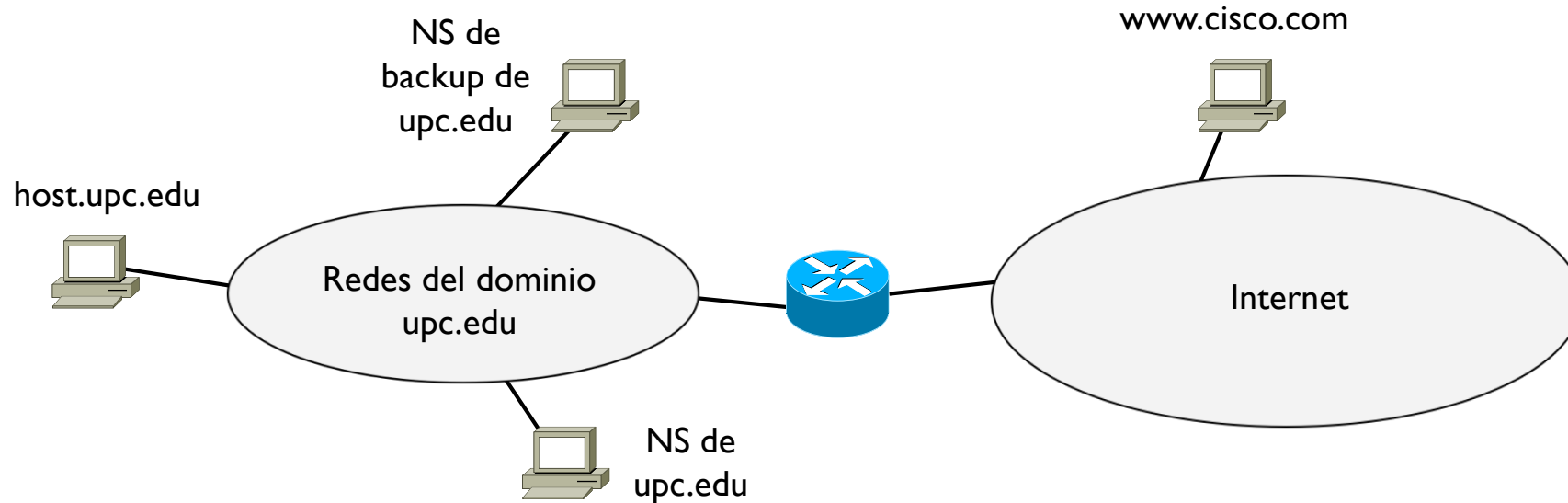
- ▶ El NS comprueba si ya tiene resuelto el nombre **www.cisco.com**
 - ▶ El NS tiene una base de datos con algunas resoluciones ya configuradas (estáticas, es decir están configuradas manualmente y no se borran con el tiempo)
 - ▶ A parte, el NS se guarda todas las resoluciones hechas de forma dinámica durante generalmente 2 días (en una memoria llamada **cached RR**)
 - ▶ Es decir podría ser que otro host del dominio upc.edu haya pedido una misma resolución en el pasado y por lo tanto el NS podría tenerla guardada en su memoria

Tema 5 – DNS ejemplo



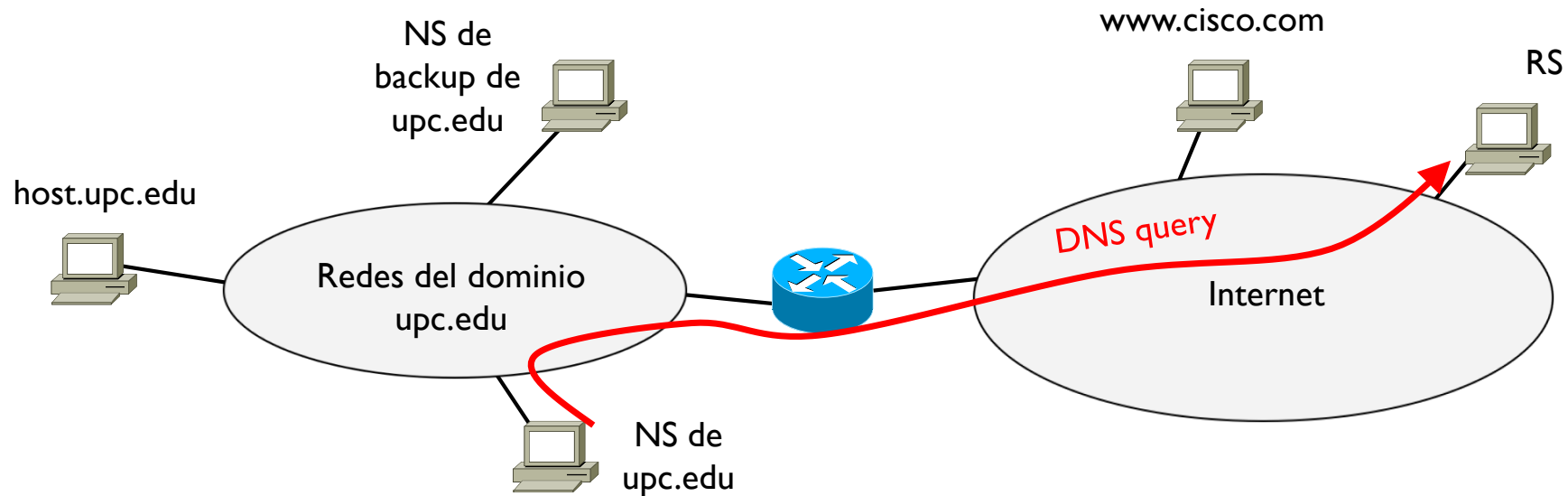
- ▶ Si ya tiene esta resolución, el NS contesta con un DNS answer al host
 - ▶ En la respuesta se indica la @IP de www.cisco.com y también el nombre y @IP del authority que ha proporcionado esta resolución

Tema 5 – DNS ejemplo



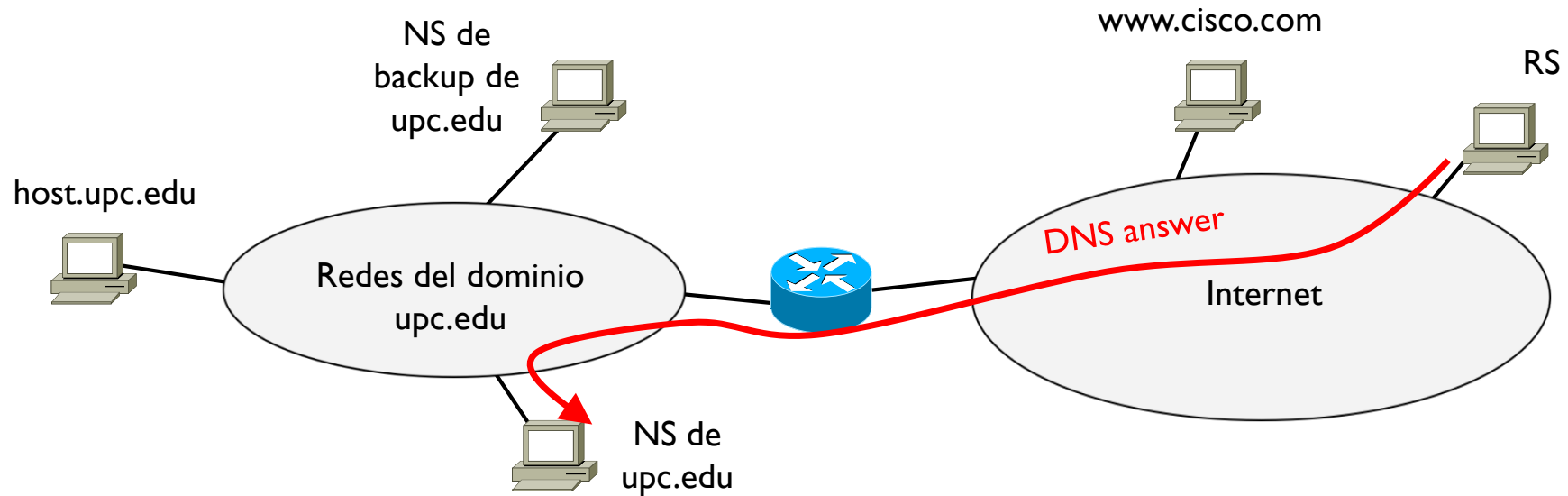
- ▶ Si no la tiene, el NS debe buscar la resolución en Internet
- ▶ Suponiendo que el NS no sabe nada del nombre www.cisco.com, el NS debe empezar la resolución a partir de un RS
- ▶ Un NS debe por lo tanto tener configurado en su base de datos el nombre y @IP de por lo menos un RS

Tema 5 – DNS ejemplo



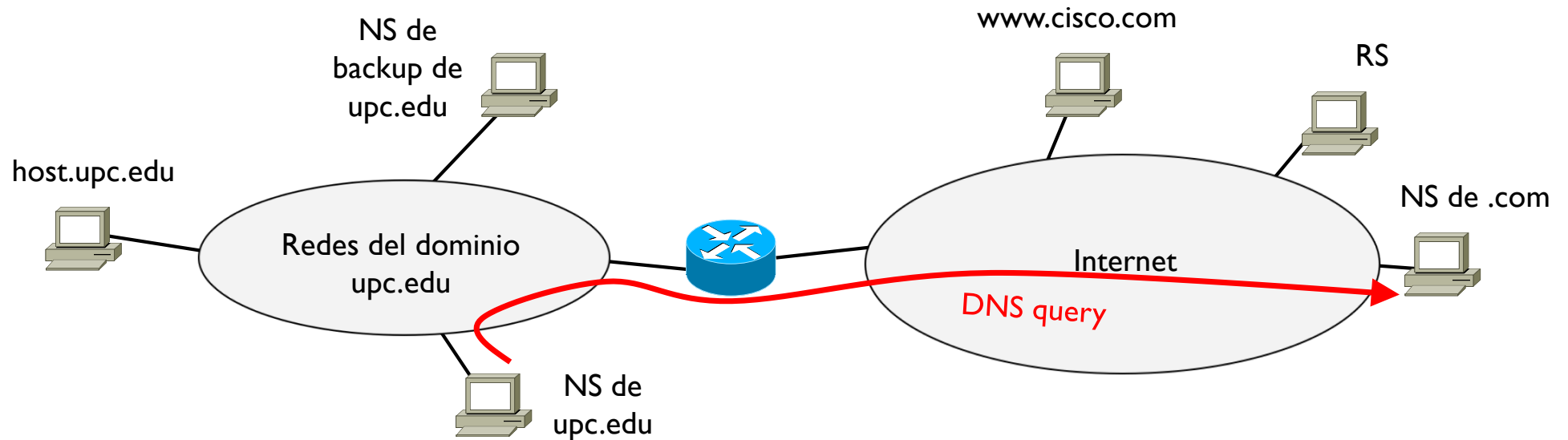
- ▶ El NS de upc.edu pide a un RS la @IP de un NS del dominio .com
 - ▶ Es decir la resolución siempre es a partir de la raíz y se recorre el árbol hasta la hoja

Tema 5 – DNS ejemplo



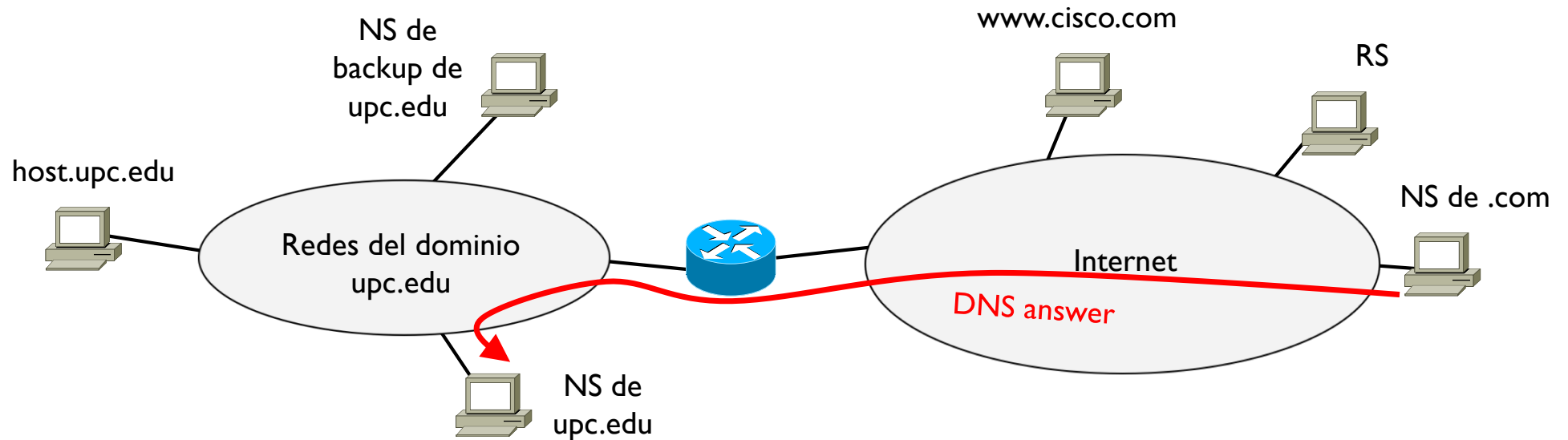
- El RS contesta con la @IP de un NS del dominio .com

Tema 5 – DNS ejemplo



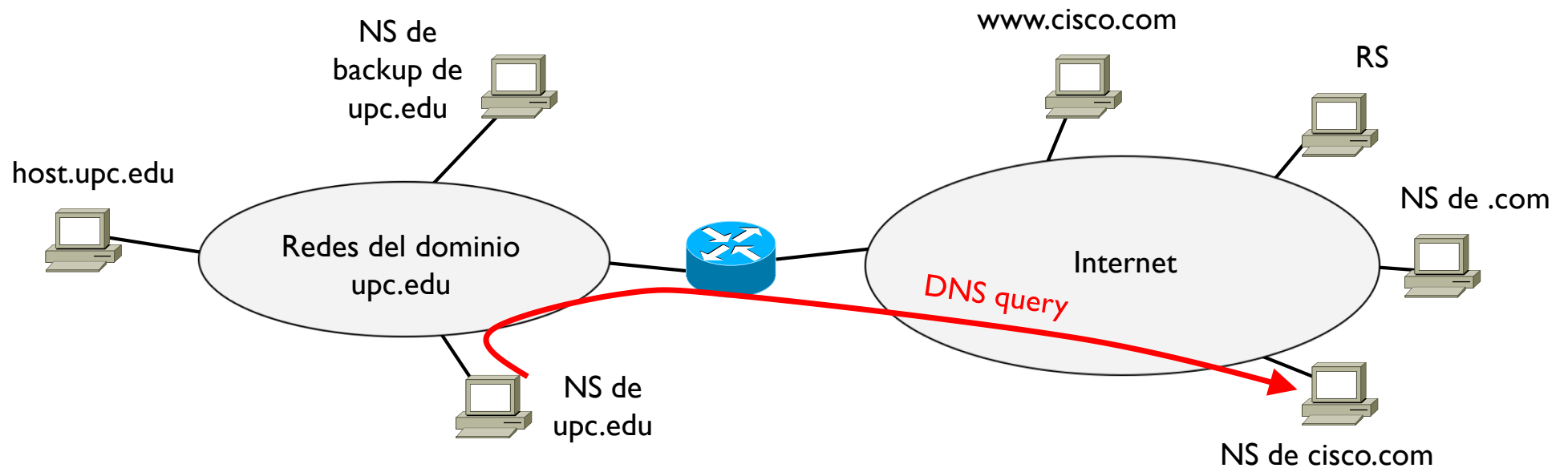
- ▶ Con esta información, el NS de upc.edu ahora puede pedir al NS del dominio .com la @IP de un NS del dominio inferior cisco.com
- ▶ Es decir se ha bajado de la authority unnamed root a una authority del nivel TLD

Tema 5 – DNS ejemplo



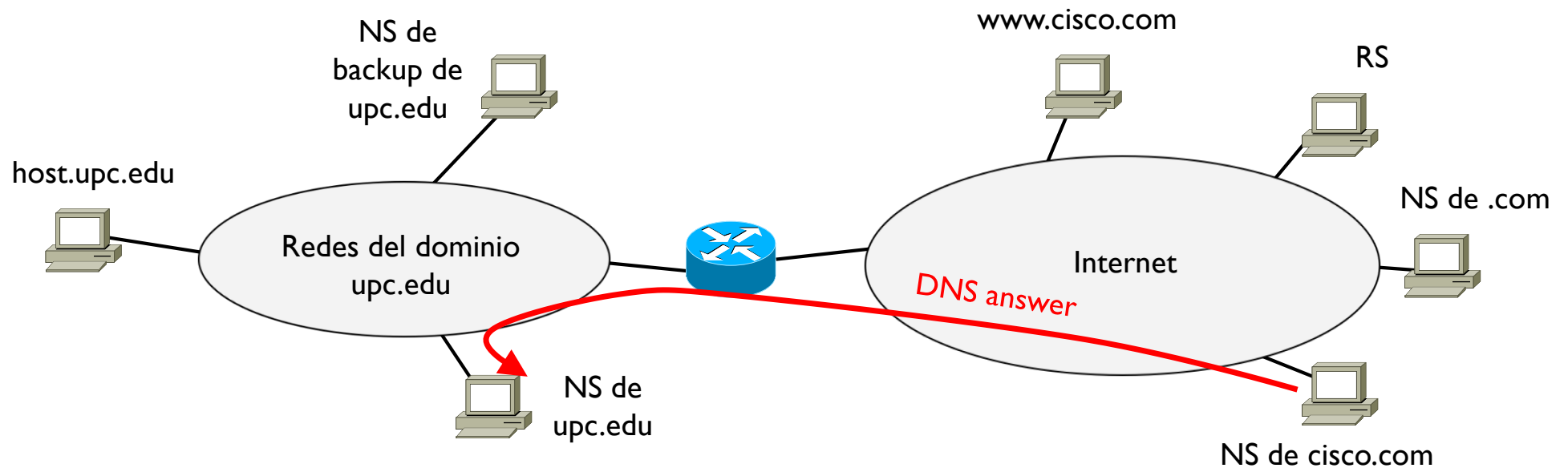
- El NS del dominio .com contesta con la @IP del NS del subdominio cisco.com

Tema 5 – DNS ejemplo



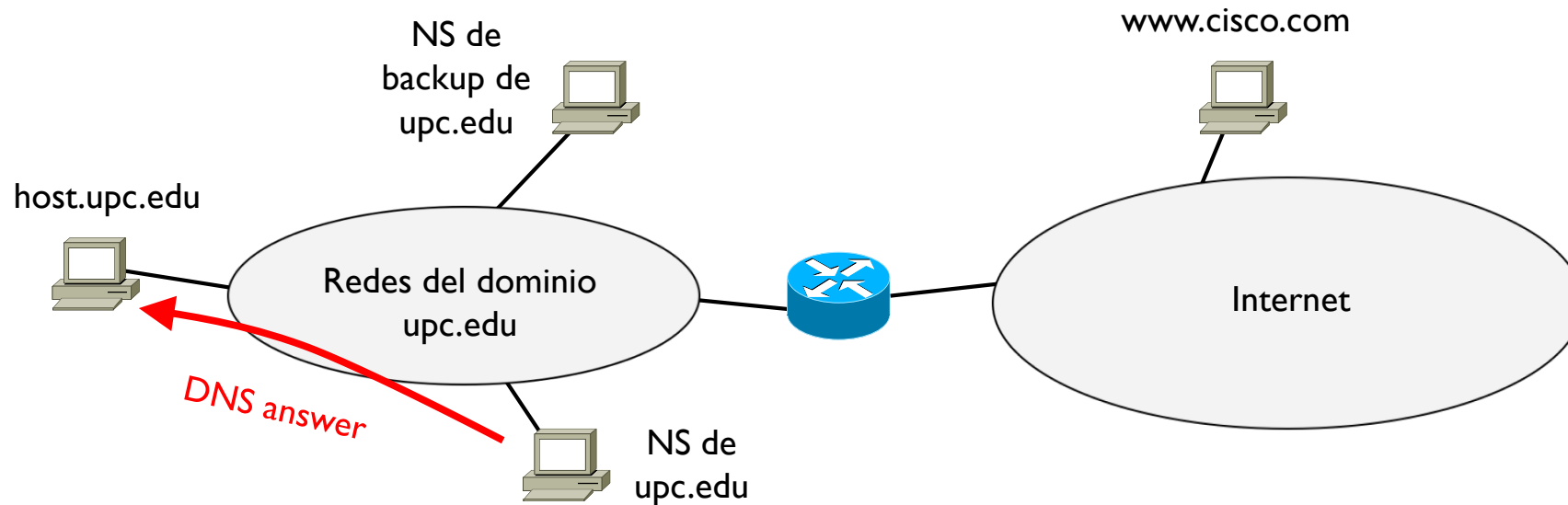
- ▶ Con esta información, el NS de upc.edu ahora puede pedir al NS del dominio cisco.com la @IP de uno de sus nodos, en concreto el www (servidor de paginas web)
 - ▶ Es decir ya se ha llegado a la authority que conoce lo que se estaba buscando

Tema 5 – DNS ejemplo



- ▶ El NS de `cisco.com` proporciona la @IP del servidor de paginas web de su dominio, es decir la @IP de `www.cisco.com`

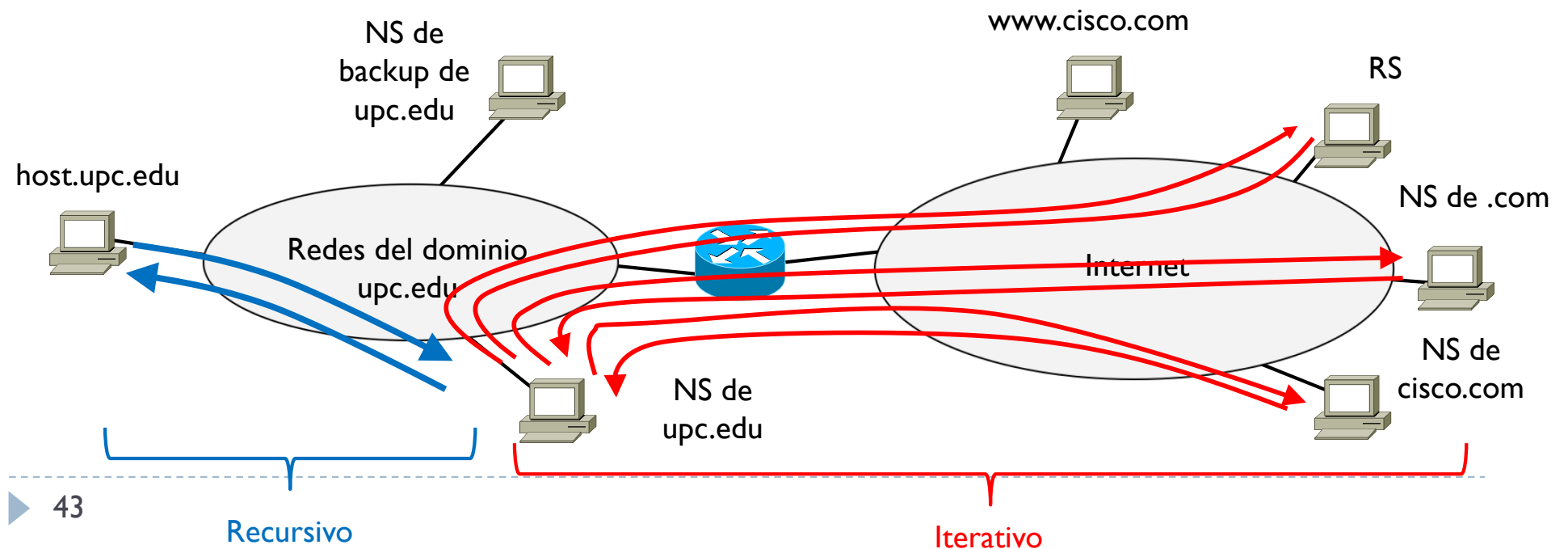
Tema 5 – DNS ejemplo



- ▶ Ahora el NS de upc.edu tiene lo que le había pedido el host y finalmente le puede contestar

Tema 5 – DNS detalles

- ▶ Una resolución puede ser
 - ▶ **Recursiva**: cuando un host pide a otro que resuelva un nombre
 - ▶ **Iterativa**: cuando un host puede resolver un nombre recorriendo el árbol jerárquico DNS
 - ▶ Es decir, generalmente un servidor NS usa una resolución iterativa mientras que un host cliente usa la recursiva (ya que necesita el NS para resolver un nombre)



Tema 5 – DNS detalles

- ▶ Un mismo nombre puede estar asociado a diferentes @IP (es decir diferentes nodos)
 - ▶ Por ejemplo no existe un único servidor www.google.com
 - ▶ En estos casos, los NS generalmente proporcionan la @IP del nodo más próximo al que ha pedido la resolución)
- ▶ Un mismo host puede tener diferentes nombres
 - ▶ Se llaman alias
 - ▶ Se puede pedir que un NS proporcione todos los alias de un nombre
 - ▶ Es una resolución que se dice de tipo CNAME
- ▶ También existe la resolución inversa
 - ▶ Conocida la @IP, se pide por el nombre
 - ▶ Es una resolución que se dice de tipo PTR

Tema 5 – DNS RR

RR format: (`name`, `value`, `type`, `ttl`)

Tipo A

- **name** es el hostname
- **value** es la @IP

Tipo CNAME

- **name** es un alias
- **value** es el nombre canonico
p.e., **www.ibm.com** es un alias de **servereast.backup2.ibm.com**

Tipo NS

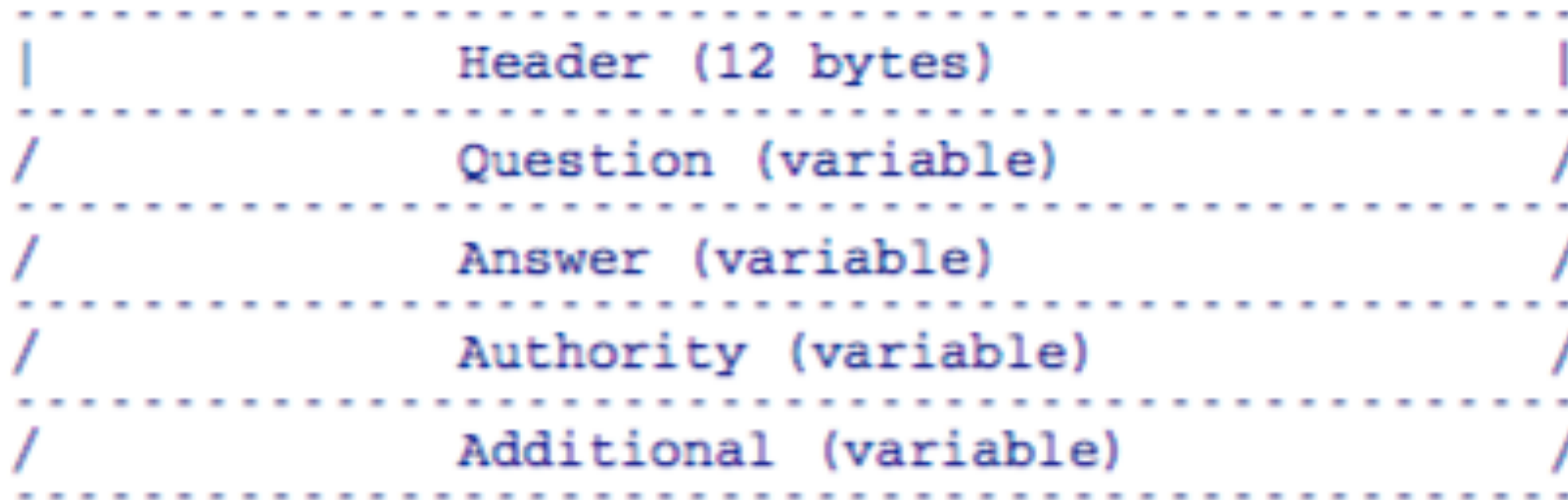
- **name** es el dominio
(p.e., foo.com)
- **value** es el nombre del
autoridad DSN del dominio

Tipo MX

- **name** es el dominio
(p.e., foo.com)
- **value** es el nombre del
servidor de correos de este
dominio

Tema 5 – Formato DNS

- ▶ Todos los mensajes DNS tienen el mismo formato

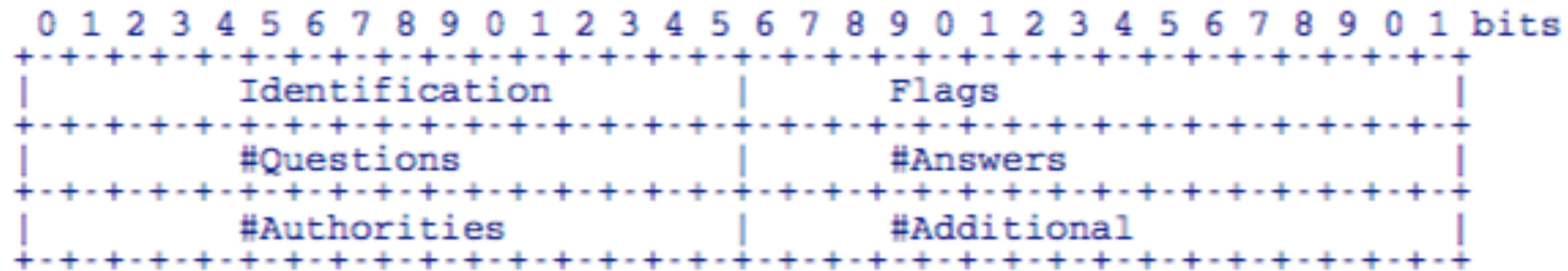


- ▶ Header: cabecera, indica el tipo de mensaje que se está enviando
 - ▶ Question: lo que se quiere resolver
 - ▶ Answer: la respuesta
 - ▶ Authority: Nombres de las authorities que han proporcionado la respuesta
 - ▶ Additional: información adicional, típicamente la @IP de las authorities
-



Tema 5 – Formato DNS

► Header

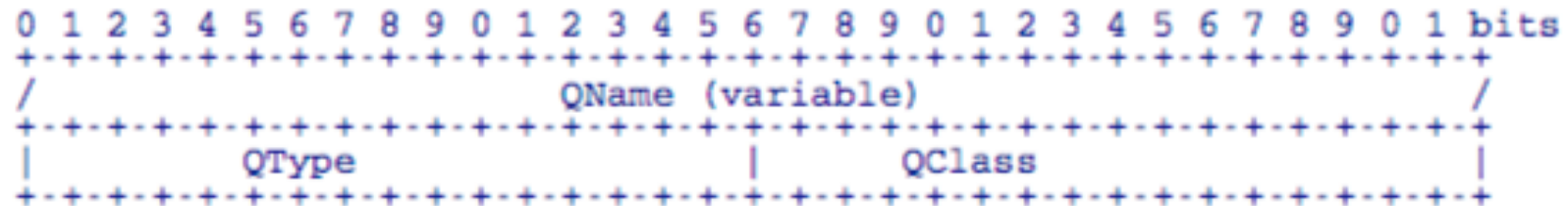


- Tamaño fijo de 12 bytes
- Identification: un número aleatorio de 16 bits que sirve para relacionar una pregunta con una respuesta
- Flags: bits que indican información sobre el tipo de mensaje
 - Flag QR: 0 si pregunta, 1 si respuesta
 - Flag AA: 1 si es una respuesta de una authority
 - Flag RD: 1 indica que se pide una resolución recursiva
- El resto de campos indican el número de preguntas, respuestas, authorities y additional hay en el resto del mensaje



Tema 5 – Formato DNS

▶ Question

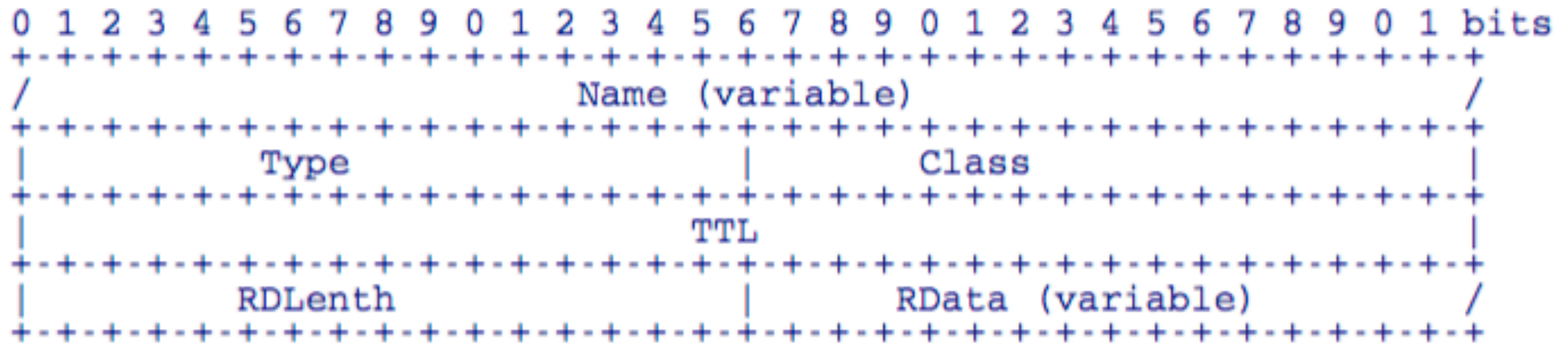


- ▶ QName: indica el nombre que se quiere resolver
- ▶ QType: indica el tipo de pregunta
 - ▶ A: Para conocer una dirección @IP a partir de un nombre
 - ▶ NS: Para conocer el nombre de un Name Server de un domain
 - ▶ PTR: Para conocer un nombre a partir de una @IP (inverso de A)
 - ▶ MX: Para conocer el nombre del server de email de un domain
 - ▶ CNAME: Para conocer todos los alias de un nombre
- ▶ QClass: indica el tipo de direccionamiento que se usa (I para IPv4)



Tema 5 – Formato DNS

- ▶ Answer, authority y additional



- ▶ Name, Type y Class como en Question
- ▶ TTL (Time To Live): número de segundos que hay guardar la resolución en la cached RR
- ▶ RDLenth: tamaño de la resolución en bytes
- ▶ RData: la resolución
 - ▶ @IP si es de tipo A
 - ▶ Un nombre si es de tipo NS, MX o PTR o varios nombres si es CNAME



Tema 5 – Protocolos de nivel aplicación

- ▶ Introducción
- ▶ Protocolos del nivel aplicaciones
 - ▶ DNS
 - ▶ **Email**
 - ▶ Web
- ▶ Otro
 - ▶ HTML
 - ▶ XML
 - ▶ Charsets

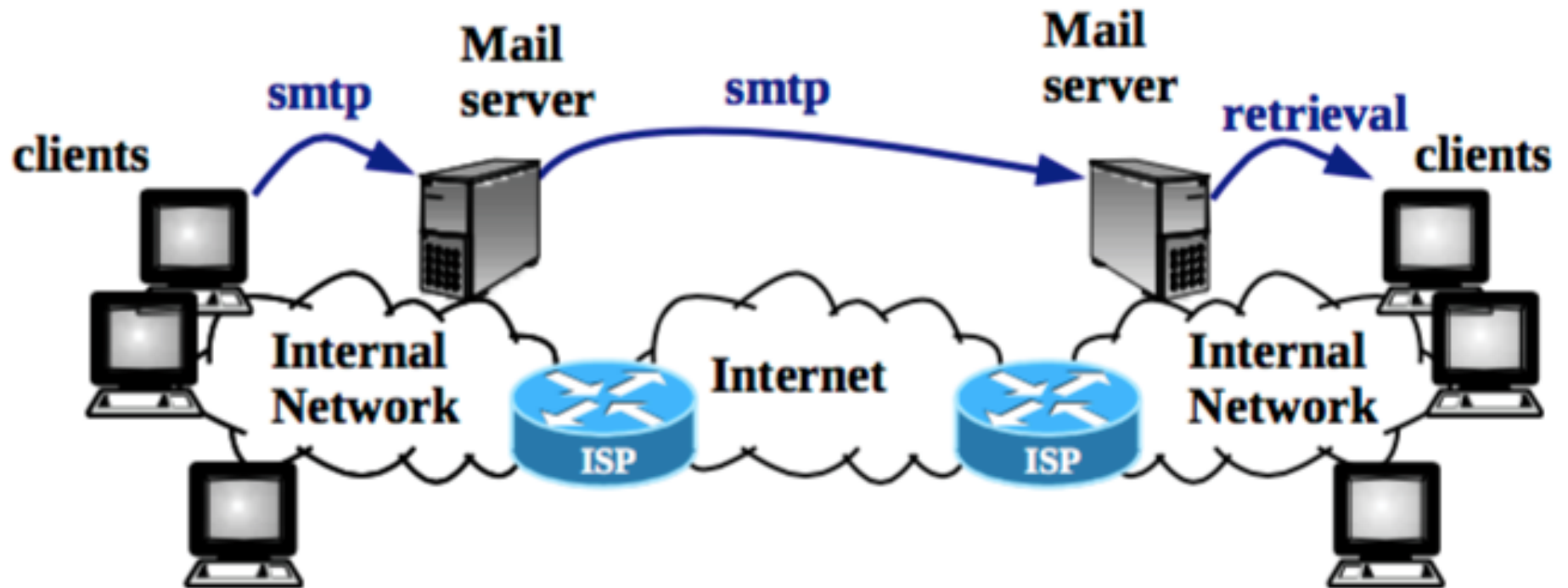
Tema 5 – Email

- ▶ Electronic mail, correo electrónico
- ▶ Una de las primeras aplicaciones que se desarrollaron cuando se creó Internet (ARPAnet entonces)
- ▶ Basado en el paradigma cliente – servidor
- ▶ Hay dos componentes básicos
 - ▶ Protocolo para el envío de los correos: SMTP
 - ▶ Protocolo para recuperar correos (retrieval): POP, IMAP, HTTP



Tema 5 – Email

► Modelo

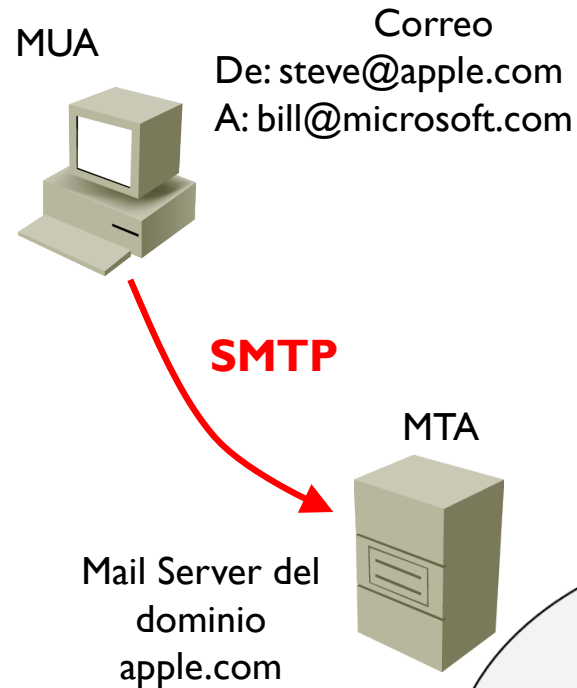


Tema 5 – SMTP

- ▶ Simple Mail Transfer Protocol
- ▶ RFC 821 (original), RFC 5321 (última versión)
- ▶ TCP, puerto 25
 - ▶ Transmisión fiable
- ▶ Definiciones
 - ▶ MUA: Mail User Agent, p.e. Mail, Outlook, Thunderbird, etc.
 - ▶ MTA: Mail Transfer Agent, p.e. Sendmail, Postfix, etc.
- ▶ El formato de un correo es Nombre@dominio
 - ▶ Nombre: el nombre del usuario, debe tener una cuenta en el servidor de correo del dominio
 - ▶ Dominio: el dominio del sistema donde se encuentra el servidor de correos y el usuario



Tema 5 – SMTP funcionamiento

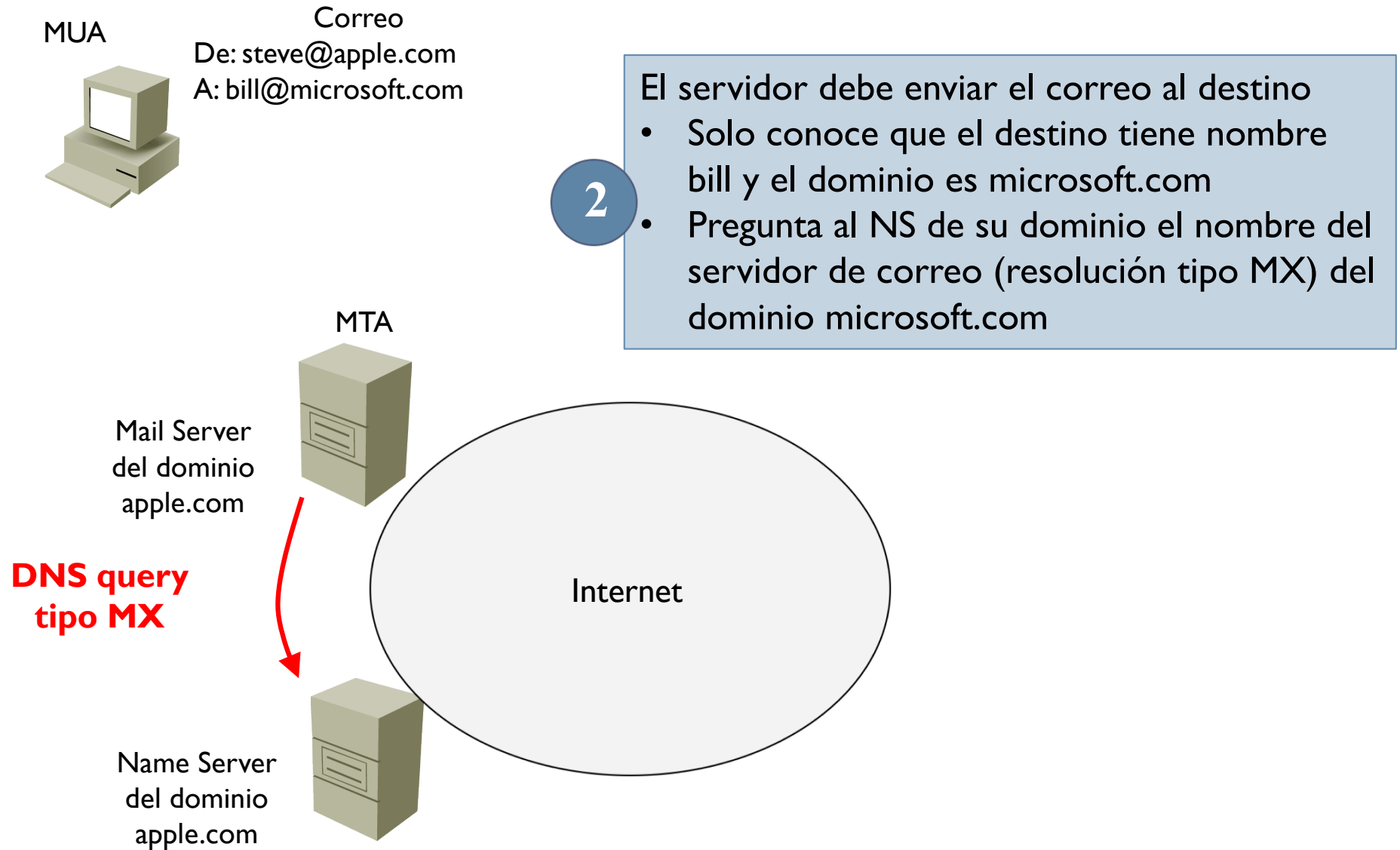


1

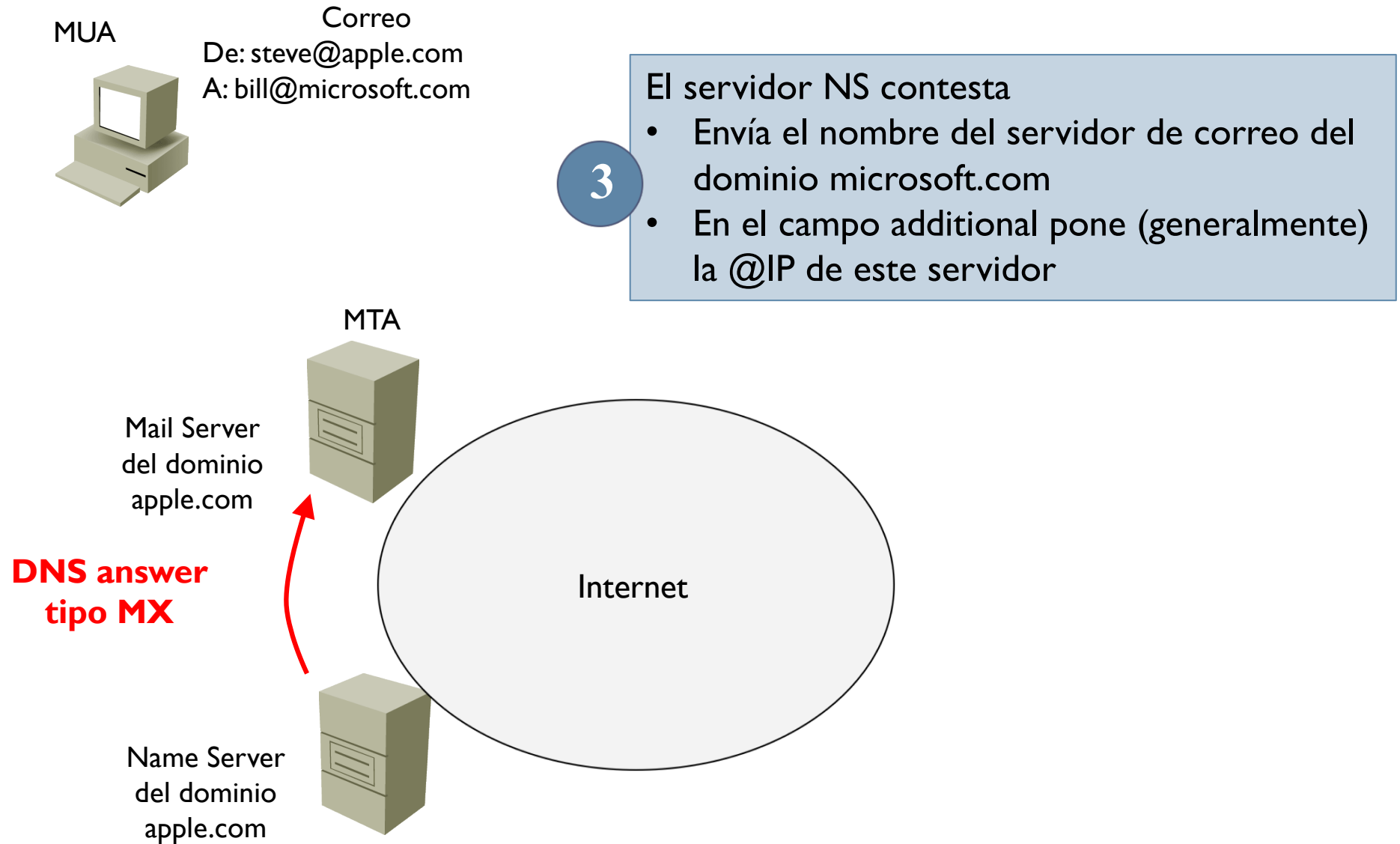
Steve quiere enviar un correo a Bill

- El cliente MUA envía el correo al servidor de correos de su dominio
- Los correos se guardan en un buzón (mailbox) de salida
- Se usa el protocolo SMTP

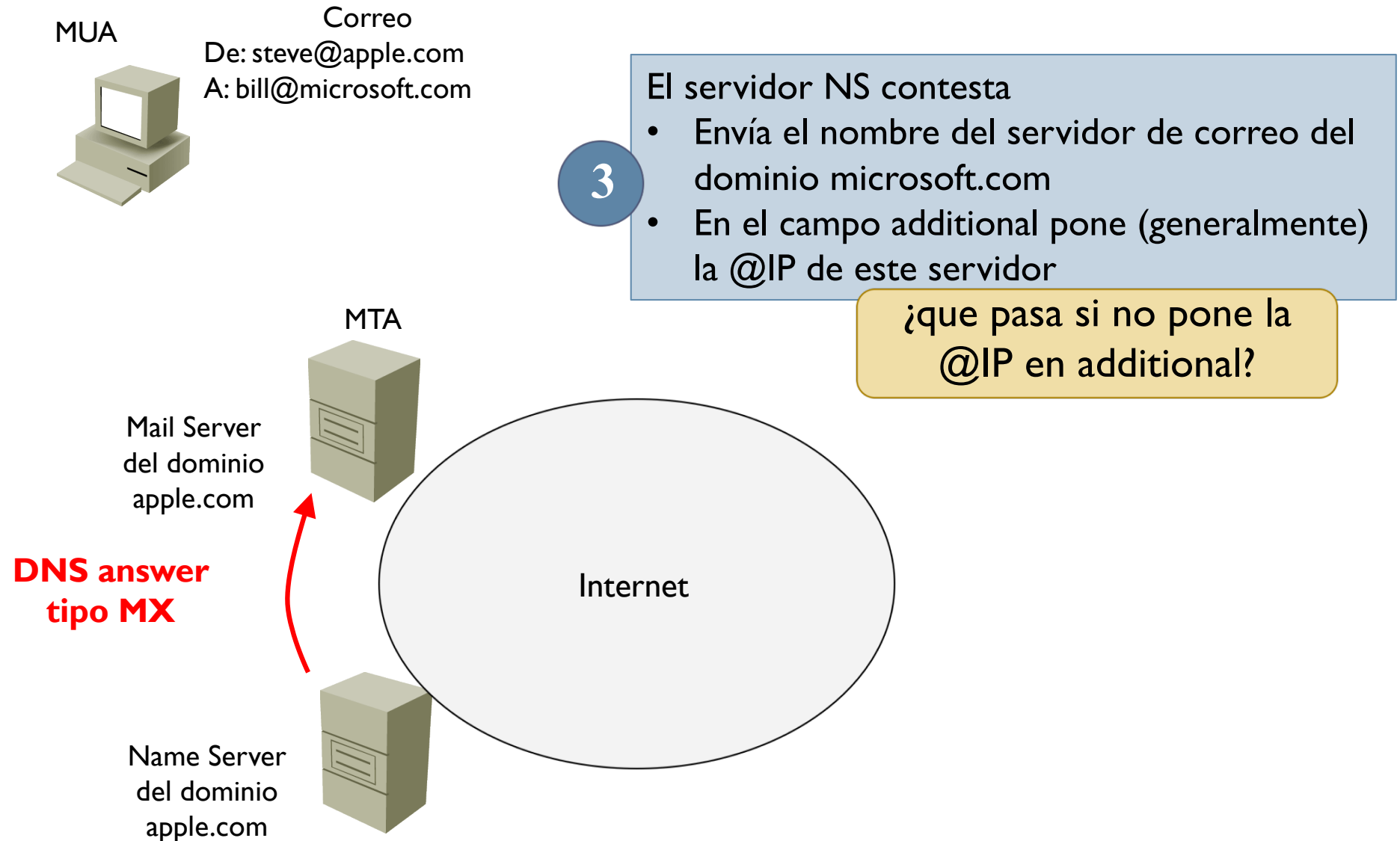
Tema 5 – SMTP funcionamiento



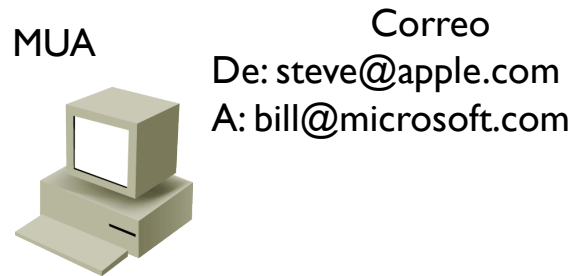
Tema 5 – SMTP funcionamiento



Tema 5 – SMTP funcionamiento



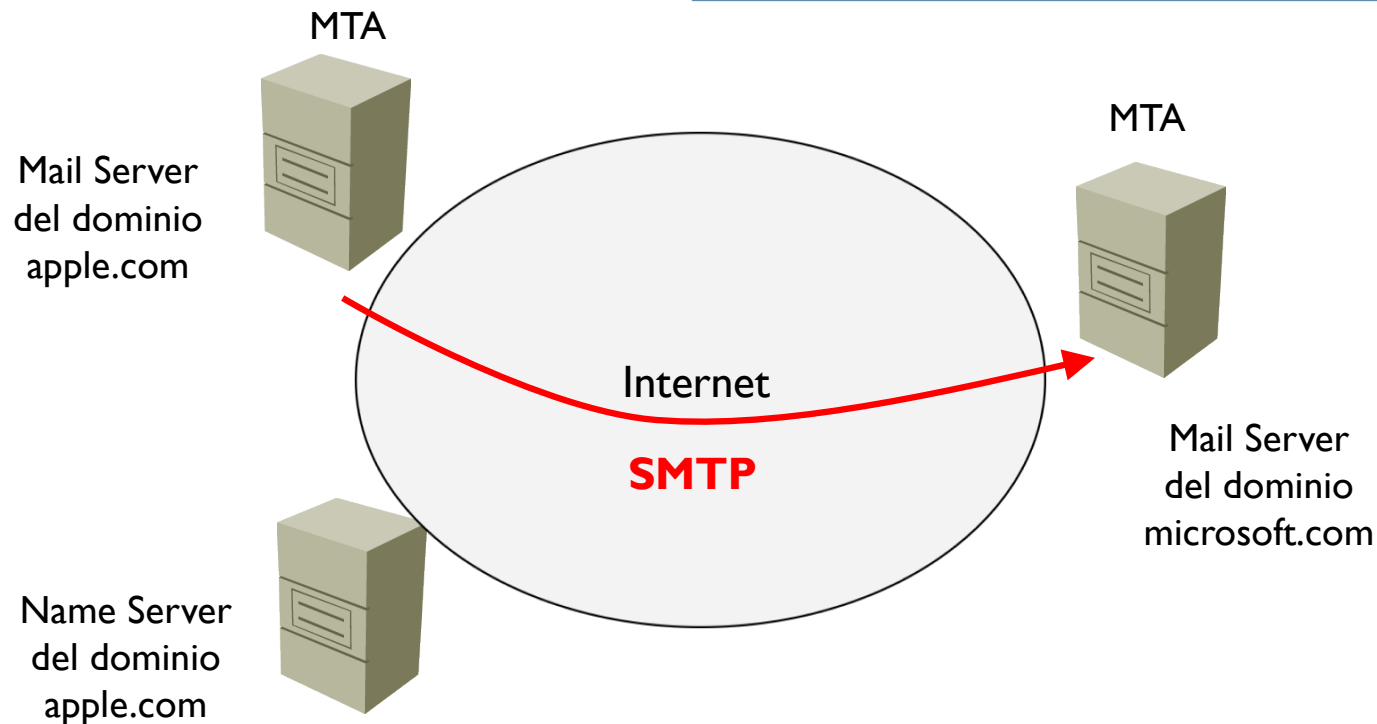
Tema 5 – SMTP funcionamiento



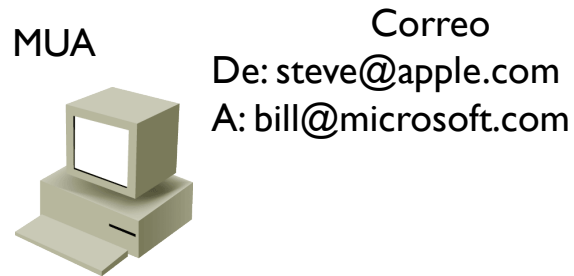
4

El servidor de correo

- Envía el correo al servidor de correo de microsoft.com usando como destino la @IP que le ha dado el NS
- Se usa el protocolo SMTP



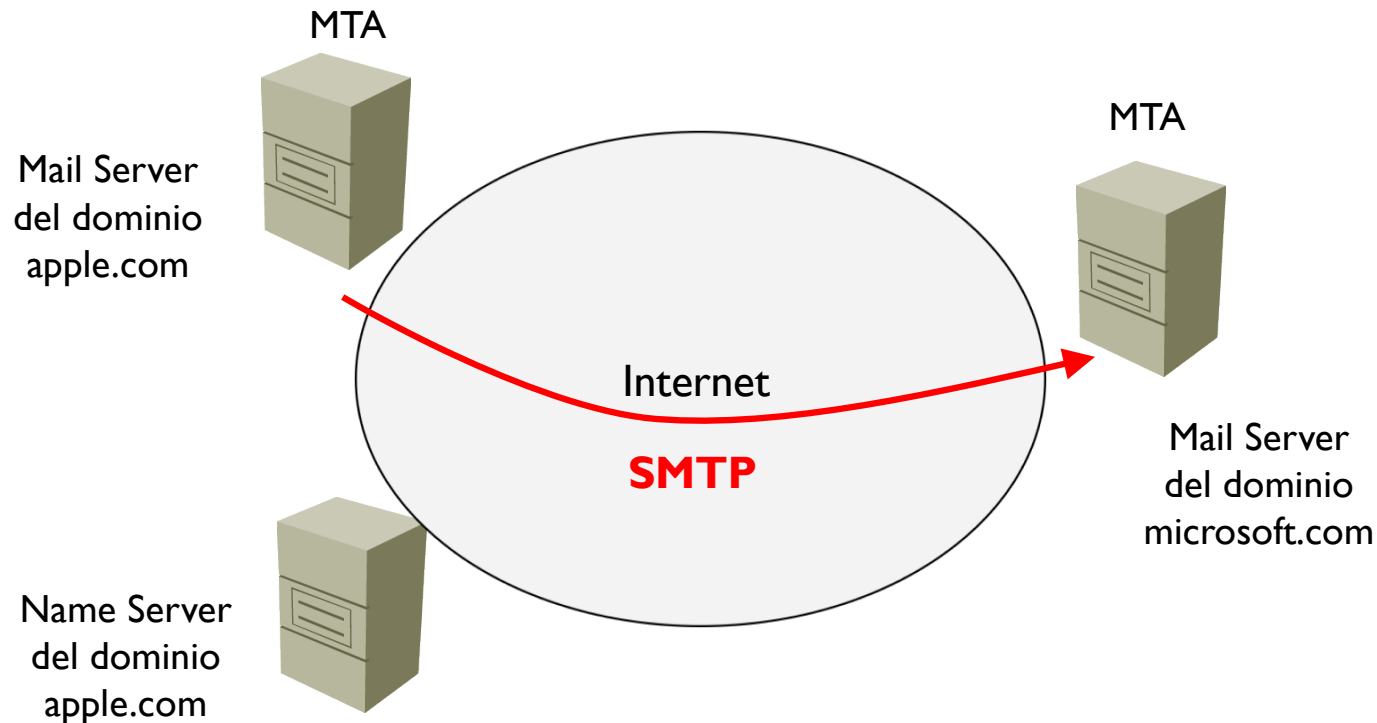
Tema 5 – SMTP funcionamiento



5

En el servidor

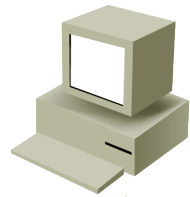
- Cada usuario tiene su buzón (mailbox) donde se almacenan los correos entrantes a la espera del retrieval



Tema 5 – SMTP funcionamiento

MUA

Correo
De: steve@apple.com
A: bill@microsoft.com



SMTP

MTA



Mail Server
del dominio
apple.com

Name Server
del dominio
apple.com



Internet

SMTP

MTA

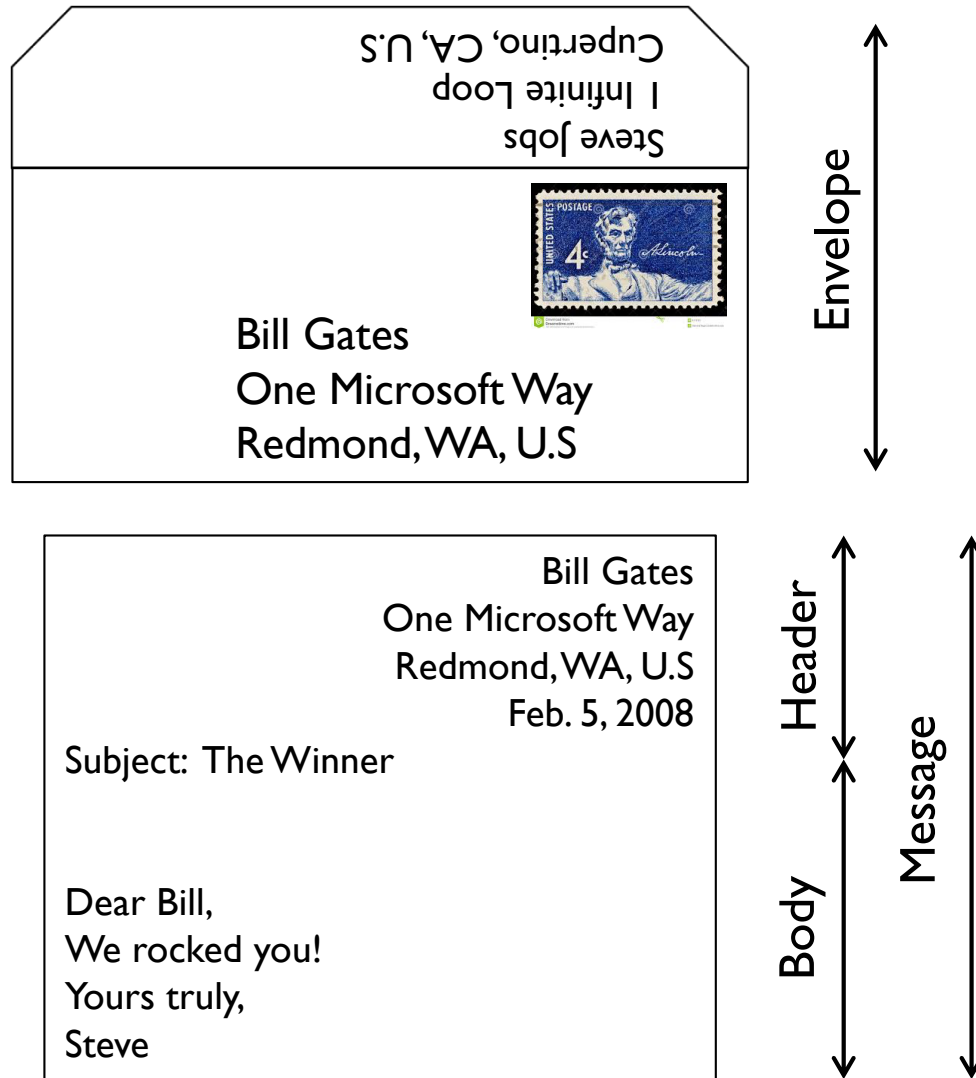


Mail Server
del dominio
microsoft.com

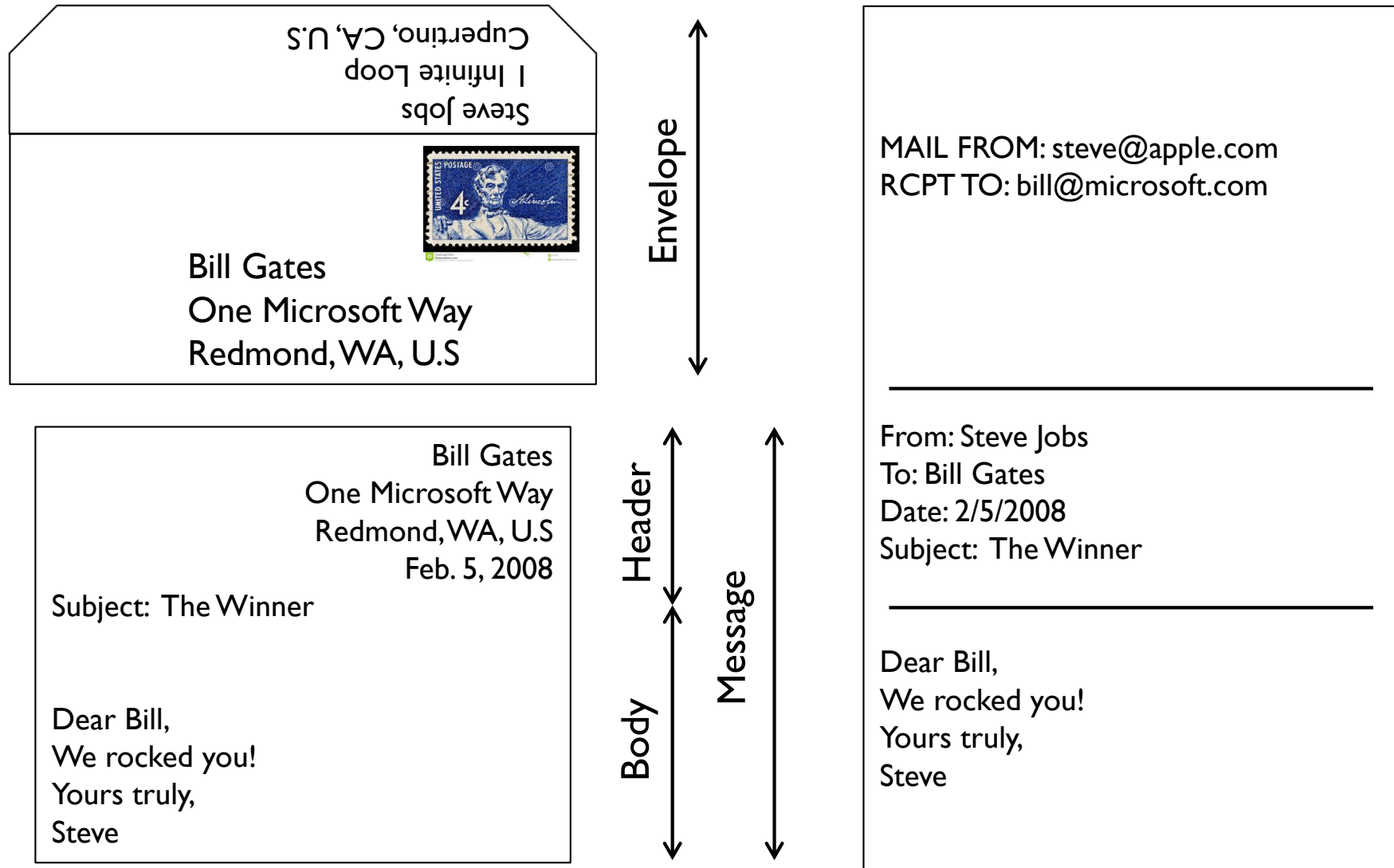
Resumen

- SMTP se ocupa de ir del MUA al MTA local (del origen) y luego al MTU destino
- Bill deberá luego bajarse los correos guardados en su servidor de correo para poder leer el correo de Steve
- Esta última ya no es tarea de SMTP

Tema 5 – El protocolo SMTP

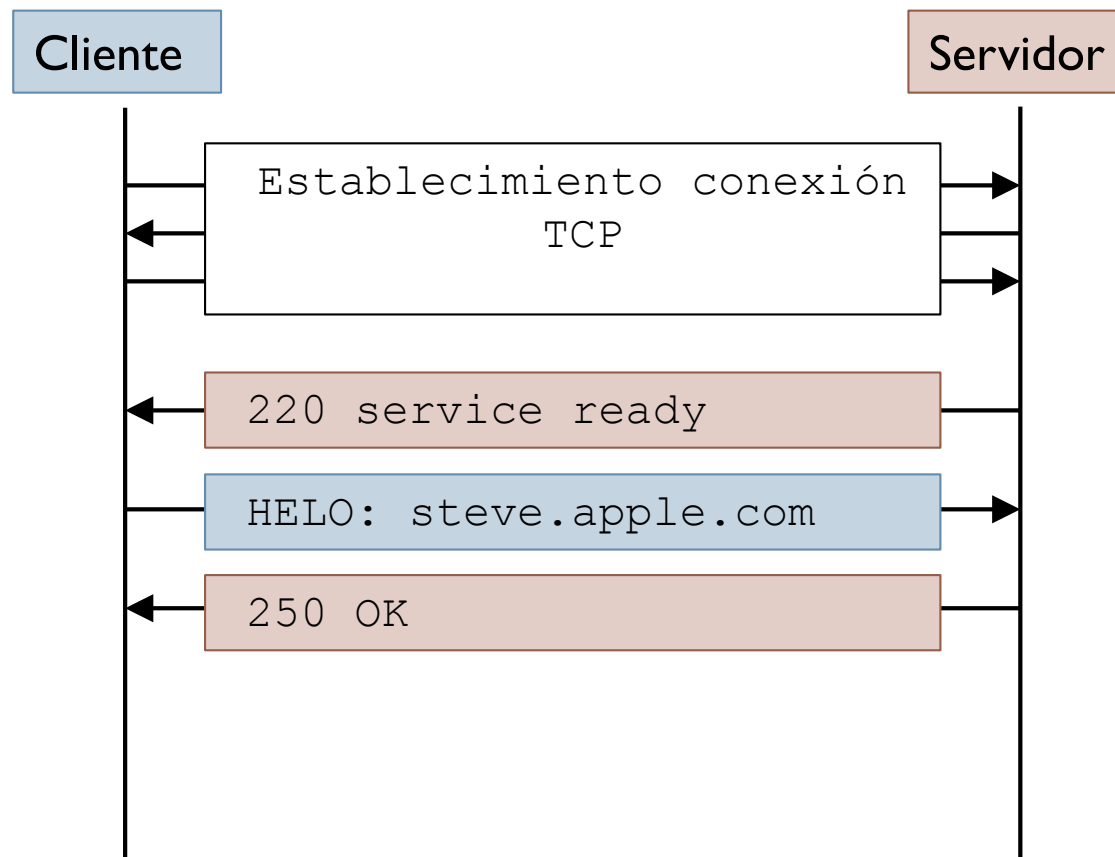


Tema 5 – El protocolo SMTP



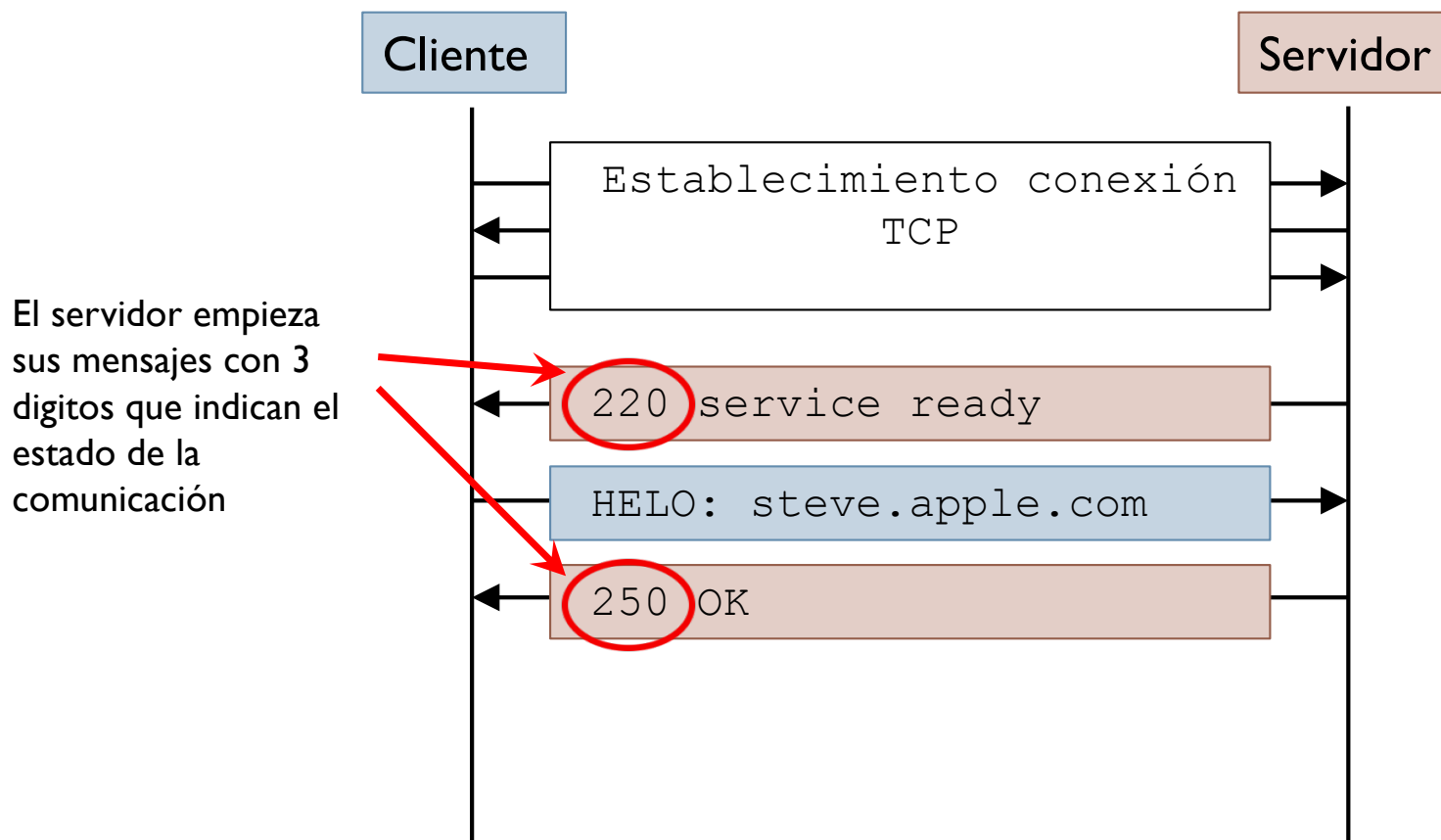
Tema 5 – El protocolo SMTP

- ▶ Diseñado para que sea muy simple (pocos comandos)
- ▶ Basado en texto ASCII



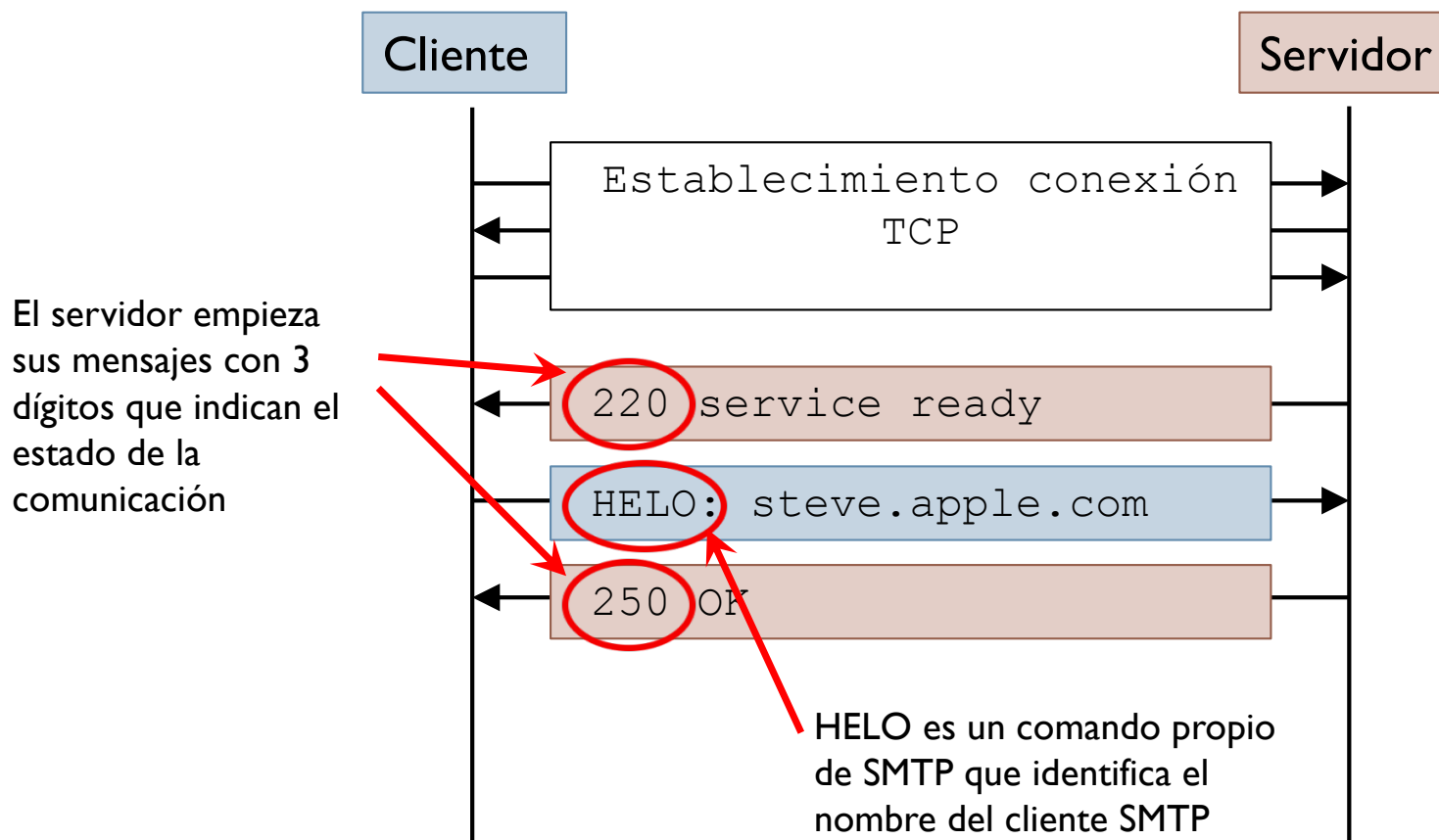
Tema 5 – El protocolo SMTP

- ▶ Diseñado para que sea muy simple (pocos comandos)
- ▶ Basado en texto ASCII

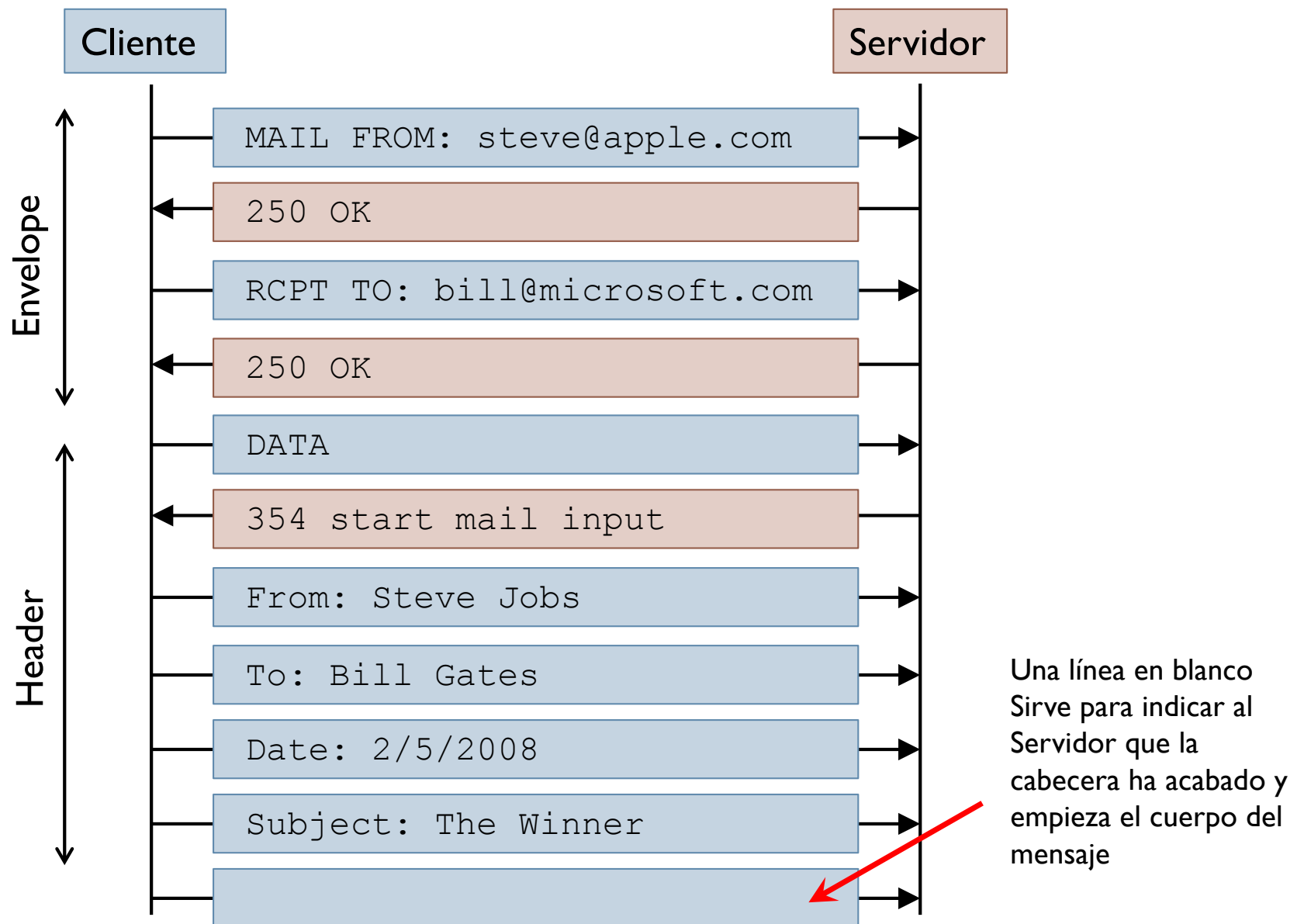


Tema 5 – El protocolo SMTP

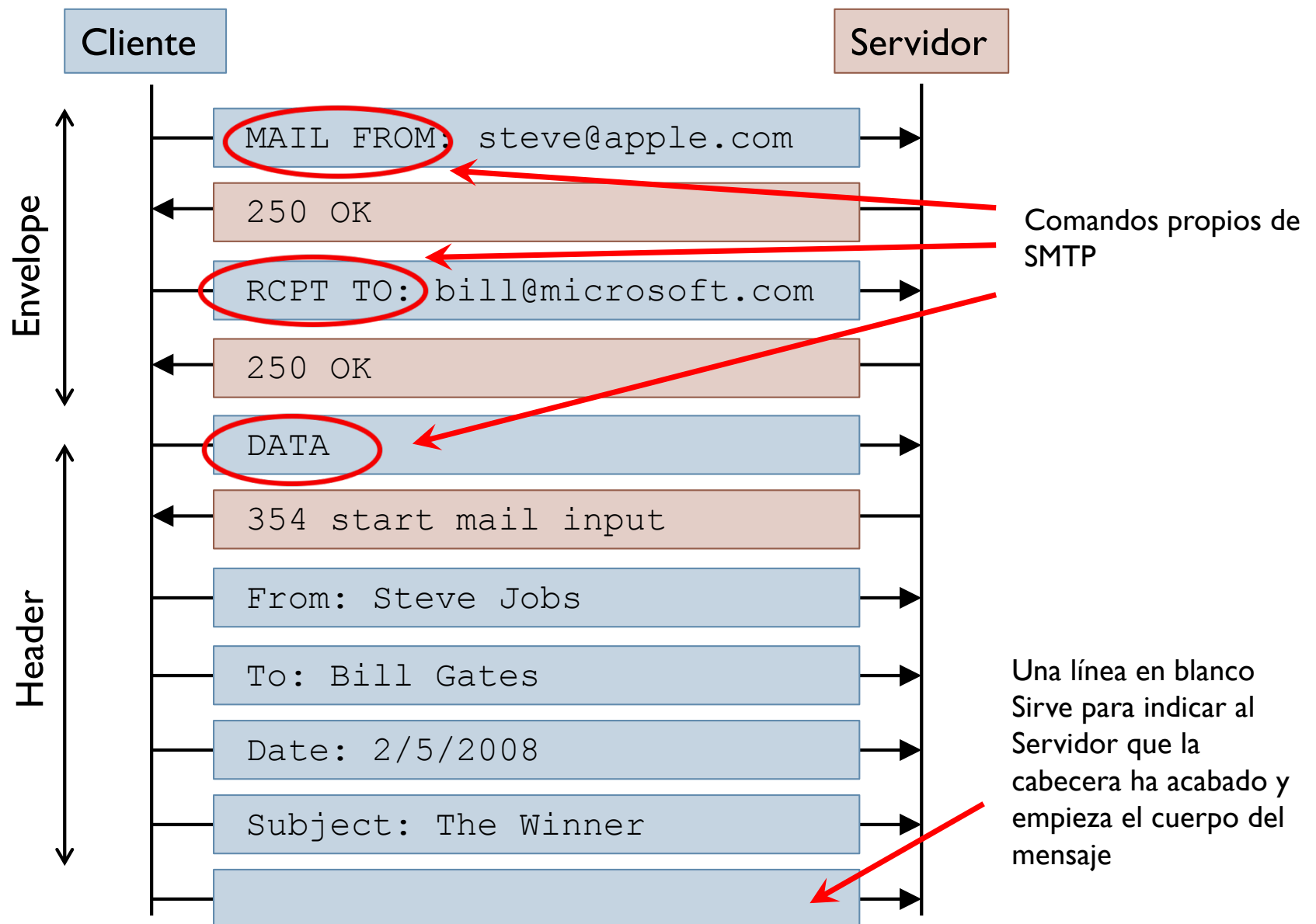
- ▶ Diseñado para que sea muy simple (pocos comandos)
- ▶ Basado en texto ASCII



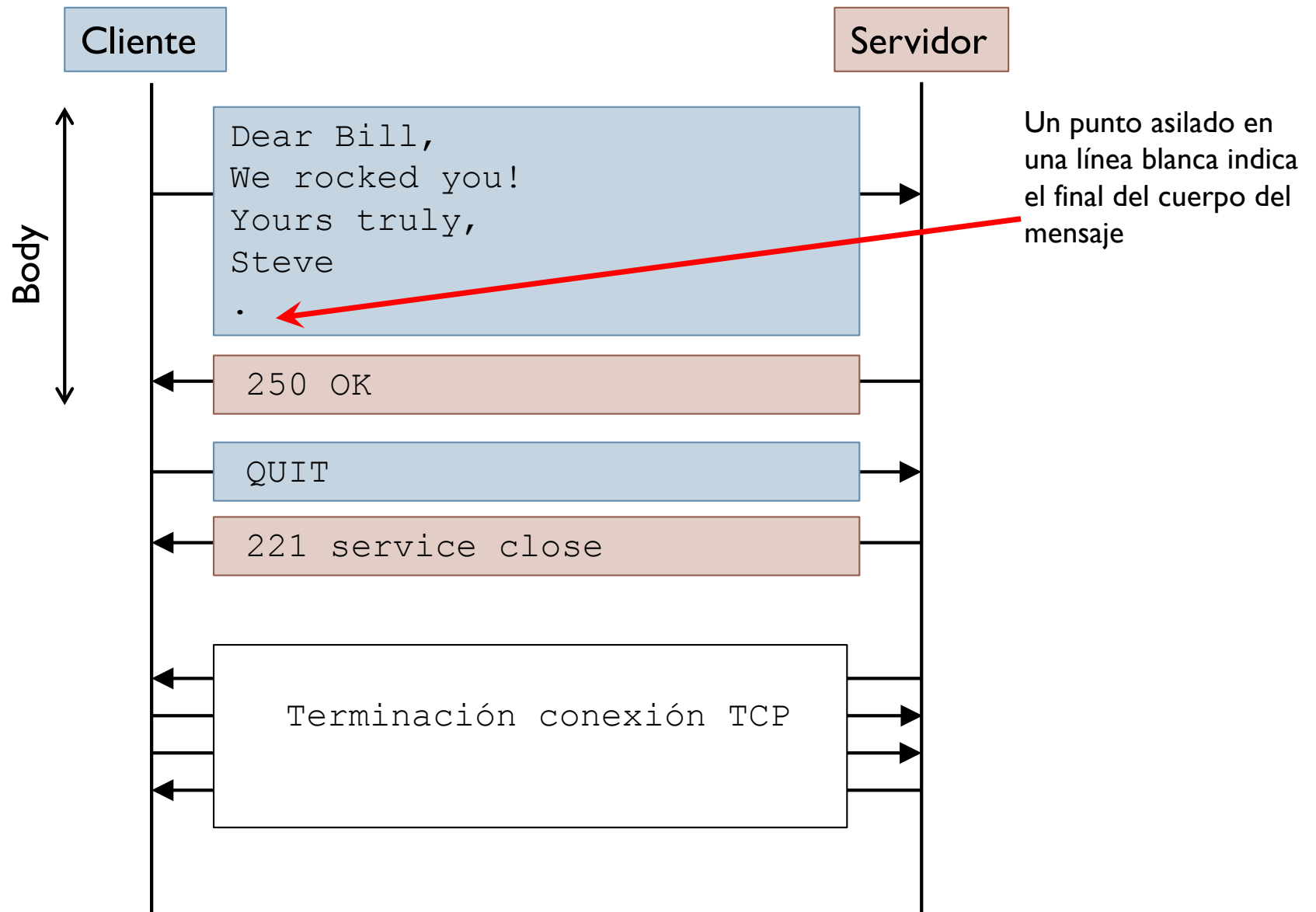
Tema 5 – El protocolo SMTP



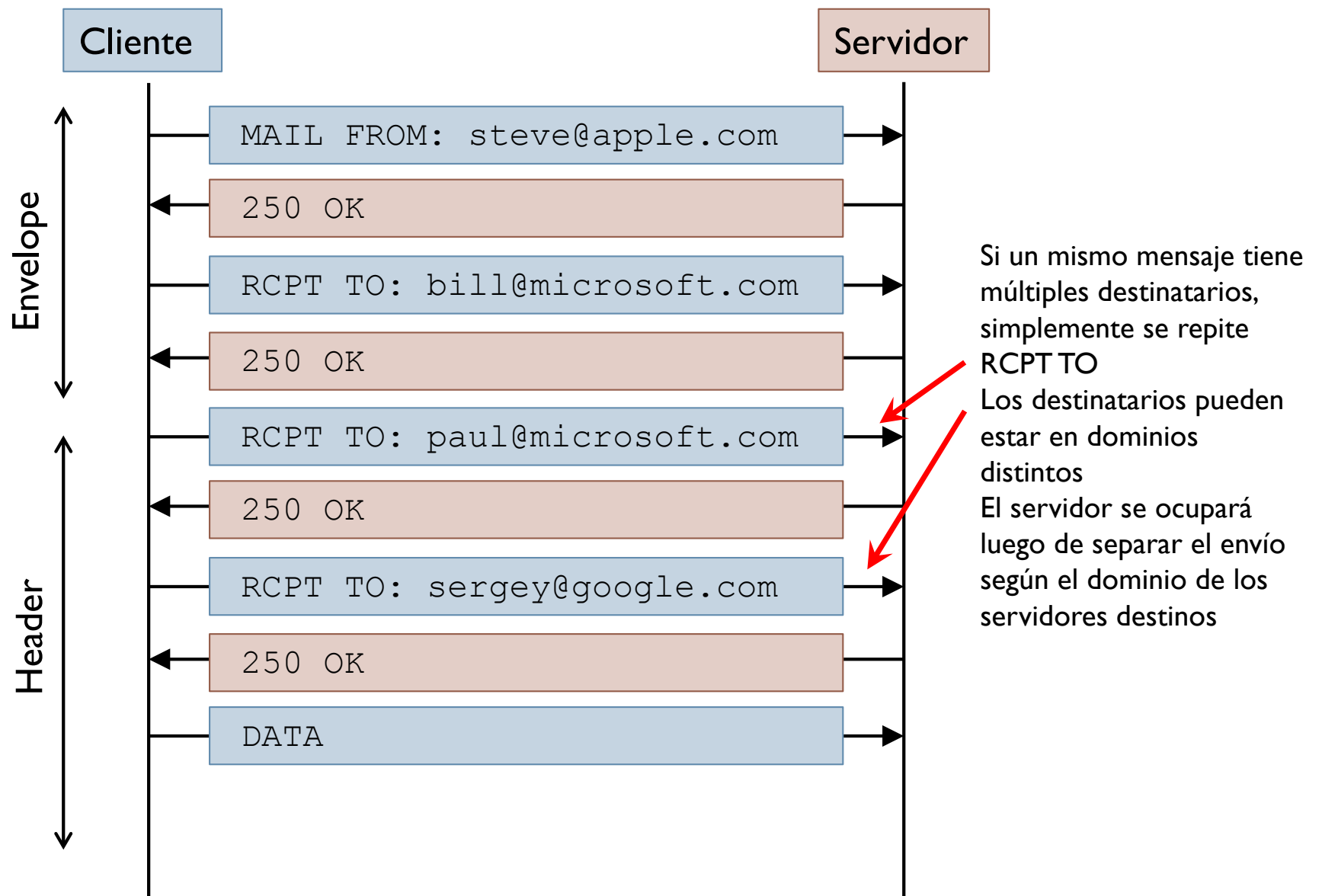
Tema 5 – El protocolo SMTP



Tema 5 – El protocolo SMTP



Tema 5 – El protocolo SMTP

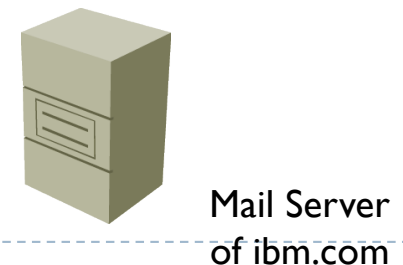
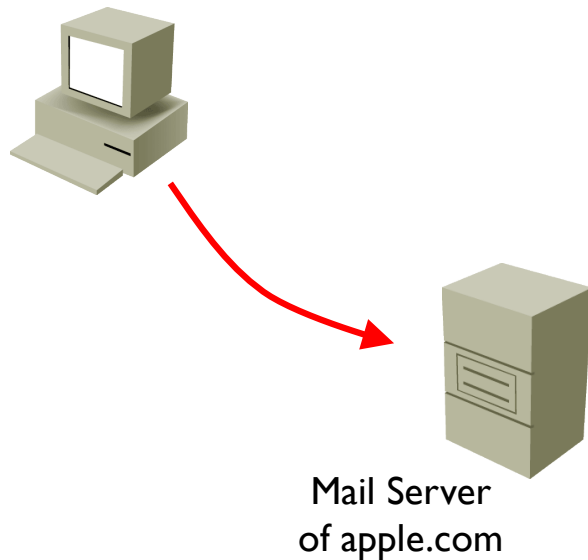


Tema 5 – El protocolo SMTP

- ▶ Se envía un mismo correo

From: `steve@apple.com`

To: `bill@microsoft.com`, `sergey@google.com`,
`paul@microsoft.com`, `lauren@ibm.com`

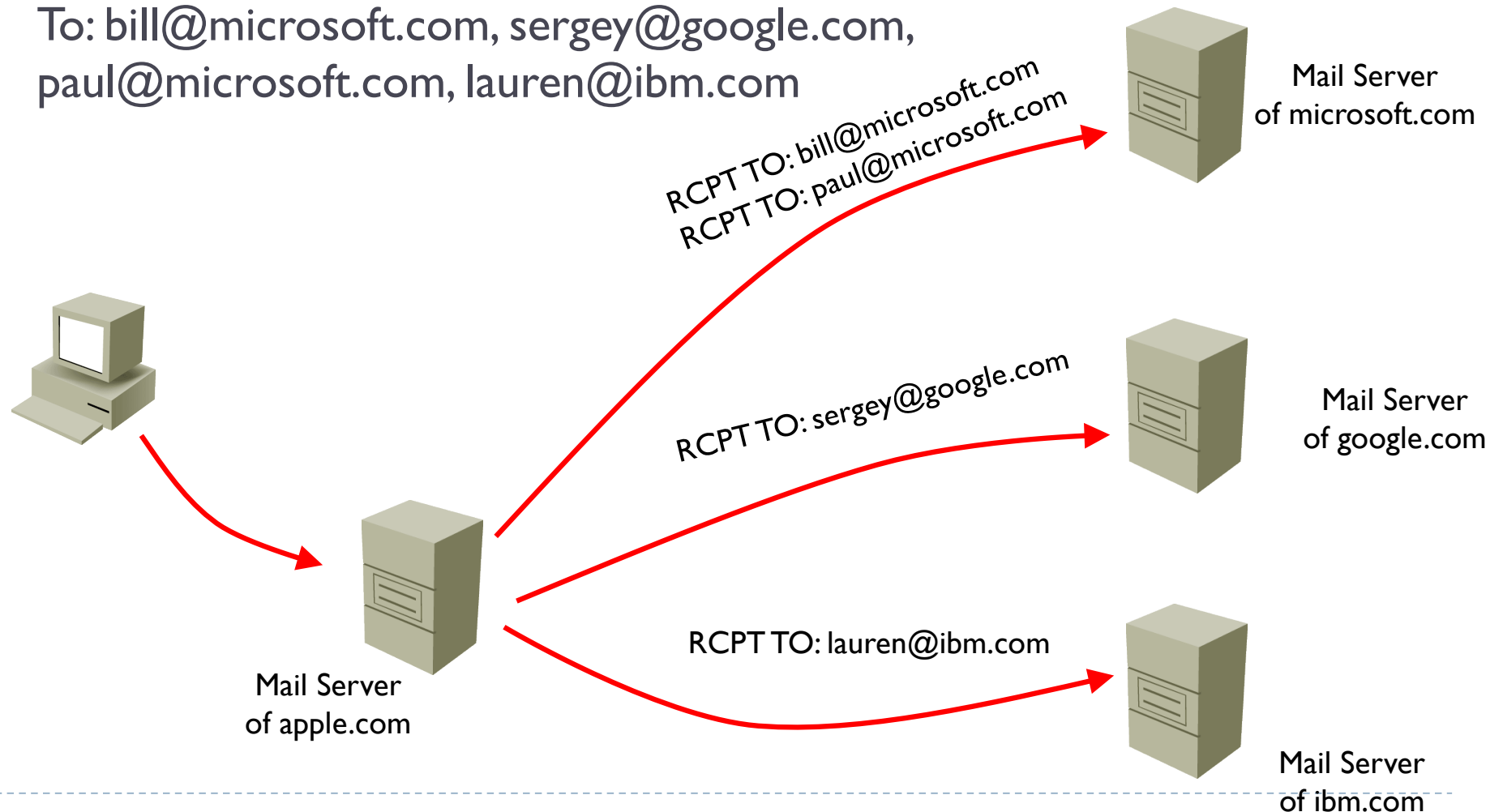


Tema 5 – El protocolo SMTP

- ▶ Se envía un mismo correo

From: `steve@apple.com`

To: `bill@microsoft.com`, `sergey@google.com`,
`paul@microsoft.com`, `lauren@ibm.com`



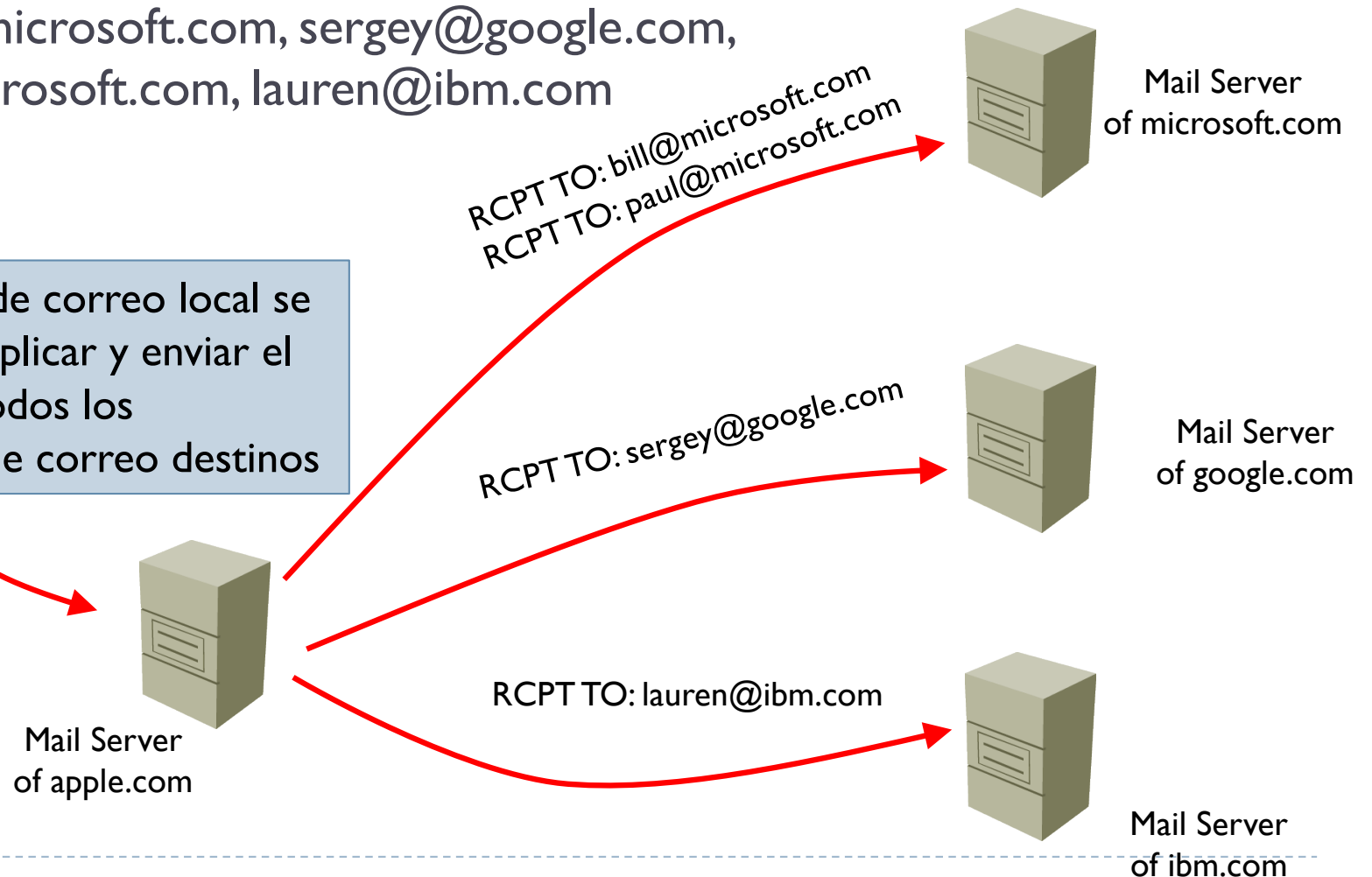
Tema 5 – El protocolo SMTP

- ▶ Se envía un mismo correo

From: `steve@apple.com`

To: `bill@microsoft.com`, `sergey@google.com`,
`paul@microsoft.com`, `lauren@ibm.com`

El servidor de correo local se ocupa de duplicar y enviar el mensaje a todos los servidores de correo destinos



Tema 5 – El protocolo SMTP

- ▶ Se envía un mismo correo

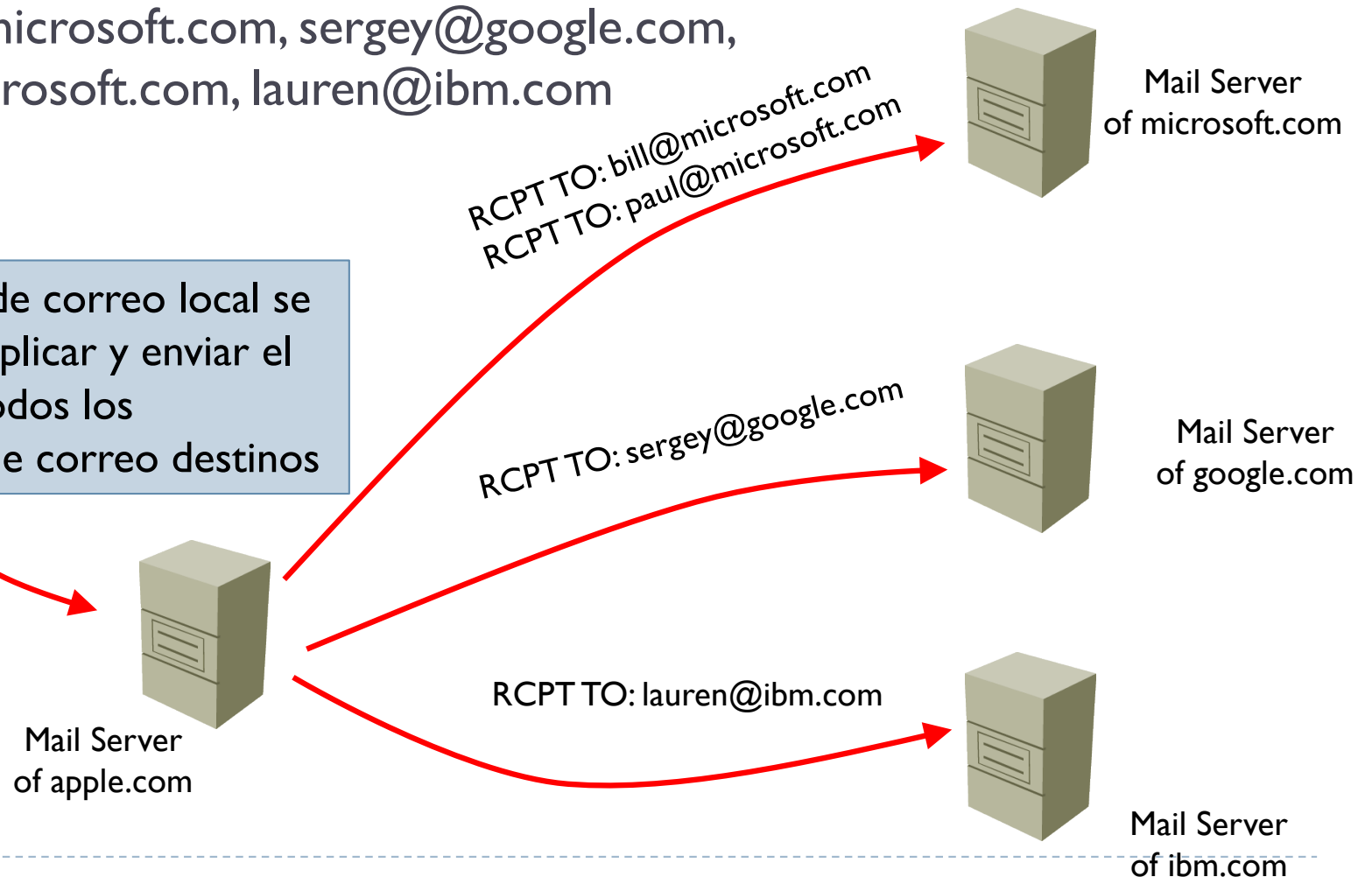
From: `steve@apple.com`

To: `bill@microsoft.com`, `sergey@google.com`,
`paul@microsoft.com`, `lauren@ibm.com`

El sobre (la comunicación entre servidores) cambia según el destino

La cabecera y el contenido del mensaje es el mismo para todos

El servidor de correo local se ocupa de duplicar y enviar el mensaje a todos los servidores de correo destinos



Tema 5 – El protocolo SMTP

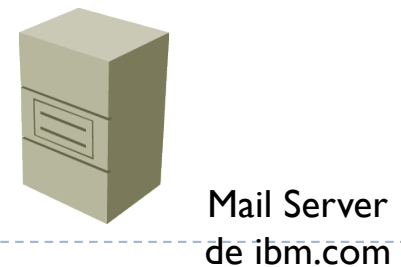
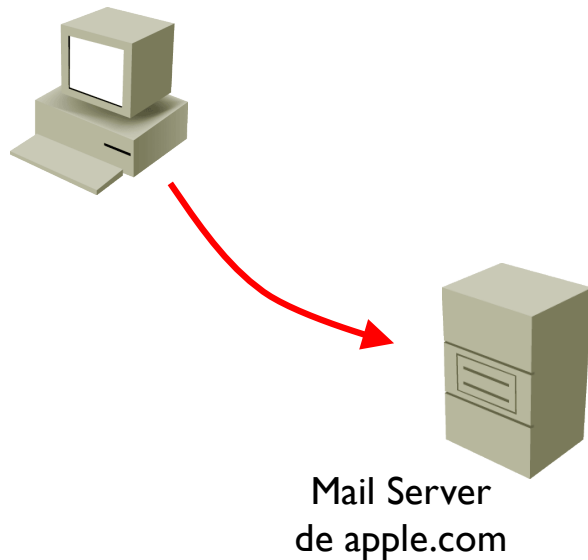
- ▶ Se envía un mismo correo

From: steve@apple.com

To: bill@microsoft.com, sergey@google.com

Cc: paul@microsoft.com

Bcc: lauren@ibm.com



Tema 5 – El protocolo SMTP

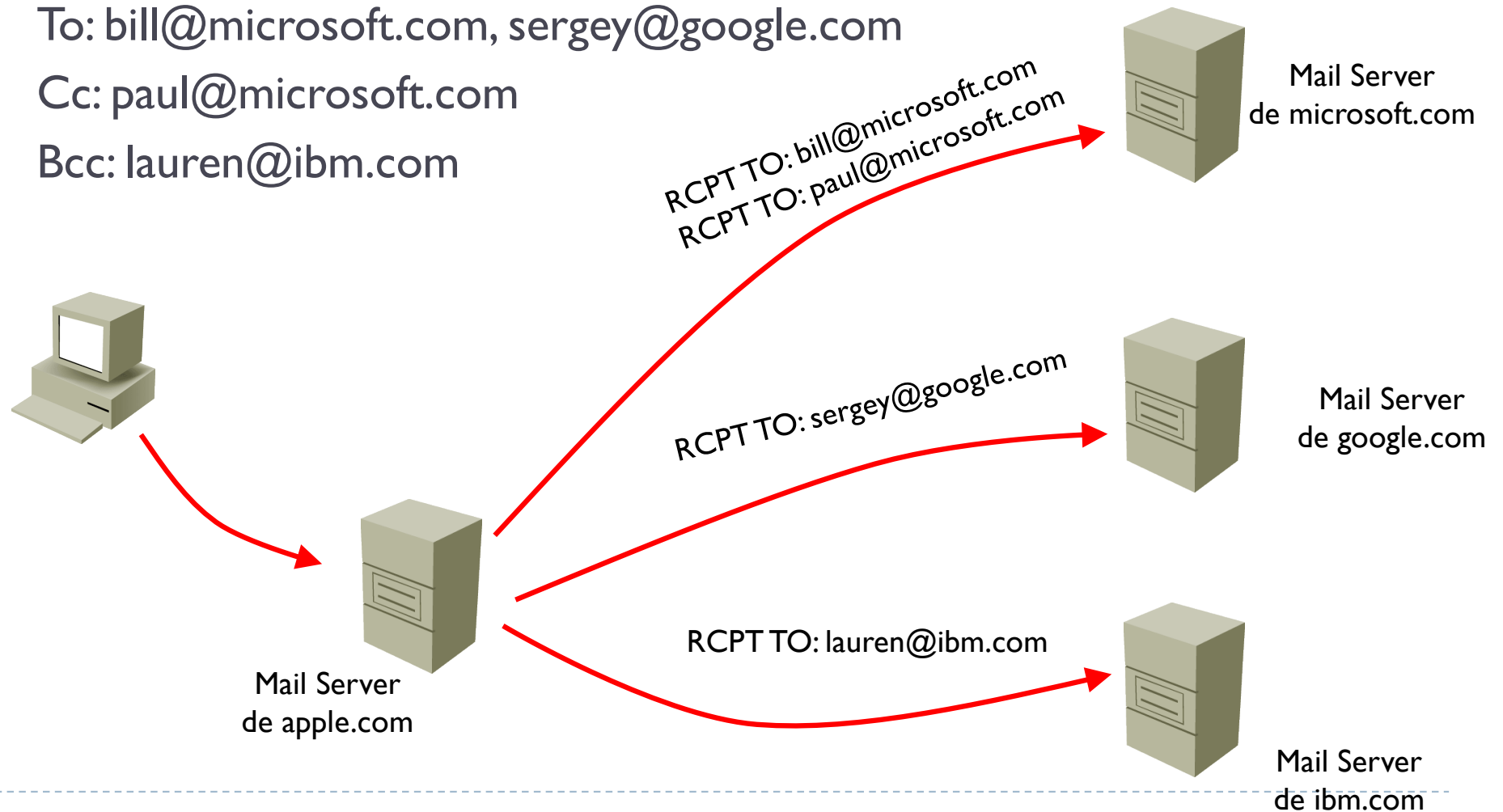
- ▶ Se envía un mismo correo

From: steve@apple.com

To: bill@microsoft.com, sergey@google.com

Cc: paul@microsoft.com

Bcc: lauren@ibm.com



Tema 5 – El protocolo SMTP

- ▶ Se envía un mismo correo

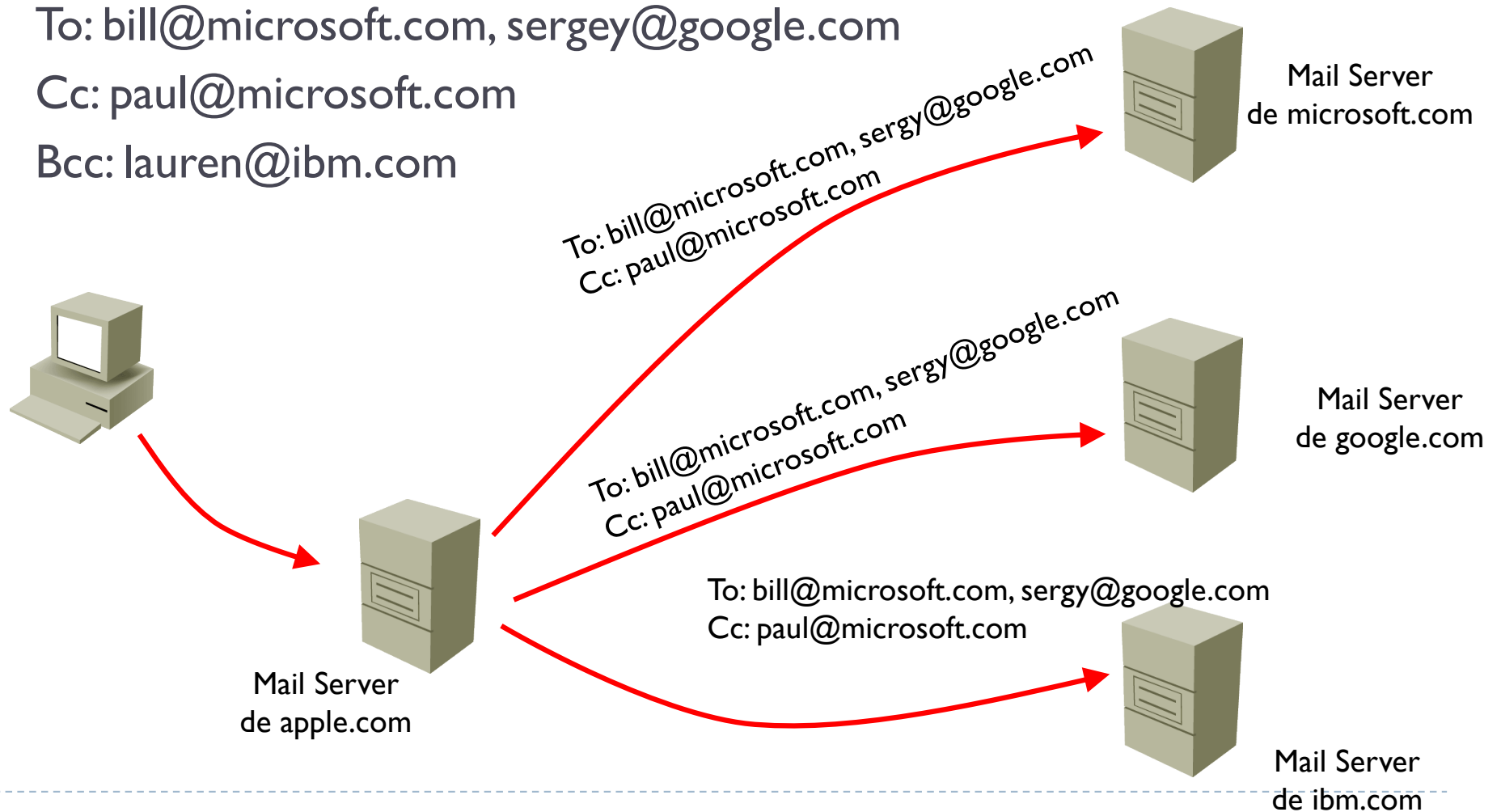
From: steve@apple.com

To: bill@microsoft.com, sergy@google.com

Cc: paul@microsoft.com

Bcc: lauren@ibm.com

Se elimina el destino Bcc de la cabecera



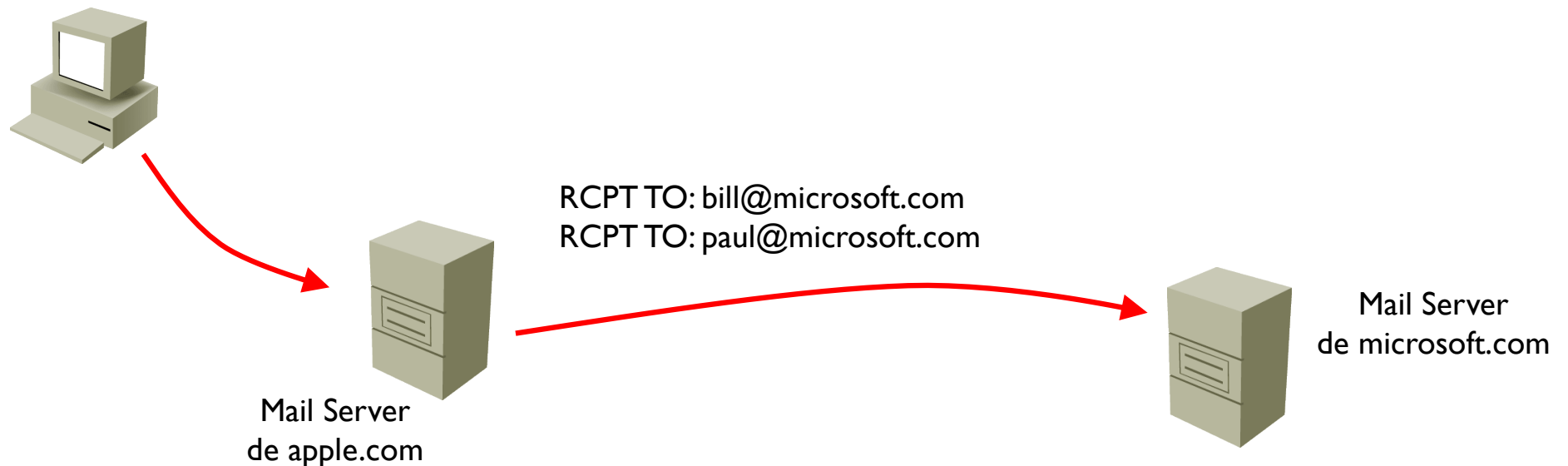
Tema 5 – El protocolo SMTP

- ▶ Se envía un mismo correo

From: steve@apple.com

Bcc: bill@microsoft.com, paul@microsoft.com

En este caso, la cabecera no contiene ningún destino. El servidor usa RCPT para saber a quien enviar el mensaje.



Tema 5 – Formatos SMTP

- ▶ Un mensaje email puede usar dos formatos distintos
 - ▶ Internet Message Format (IMF)
 - ▶ Multipurpose Internet Mail Extensions (MIME)



Tema 5 – Formatos SMTP

- ▶ Internet Message Format (IMF)
 - ▶ RFC 882 (original), RFC 5322 (última versión) actualizada en RFC 6854
 - ▶ Mensajes exclusivamente en formato texto ASCII
 - ▶ Ejemplo (extraído del RFC 5322)

```
From: John Doe <jdoe@machine.example>  
To: Mary Smith mary@example.net  
Subject: Saying Hello  
Date: Fri, 21 Nov 1997 09:55:06 -0600  
Message ID: <1234@local.machine.example>
```

```
This is a message just to say hello.  
So, "Hello".
```



Tema 5 – Formatos SMTP

- ▶ Multipurpose Internet Mail Extensions (MIME)
 - ▶ RFC 2045, 2046, 2049
 - ▶ Para incluir datos non-ASCII (p.e. audio, video, fichero)
 - ▶ El mensaje se compone
 - ▶ De una cabecera que incluye la definición de una “palabra” que hace de frontera entre tipos diferentes
 - ▶ De un cuerpo que puede contener datos de diferente formato



Tema 5 – Formatos SMTP

From: John Doe <jdoe@machine.example>
To: Mary Smith mary@example.net
Subject: Saying Hello
Date: Fri, 21 Nov 1997 09:55:06 -0600
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=separador

Indica que la palabra “separador”
separa formatos distintos de
contenido

--separador
Content-Type: text/plain

This is the body of the message.

--separador
Content-Type: audio/basic
Content-Transfer-Encoding: base64

PGh0bWw+CiAgPGhlYWQ+CiAgPC9oZWFKPgogIDxib2R5PgogICAgPHA+VGhpcyBpcyB0aGUg
Ym9keSBvZiB0aGUgbWVzc2FnZS48L3A+CiAgPC9ib2R5Pgo8L2h0bWw+Cg==

--separador
Content-Type: image/jpeg
Content-Transfer-Encoding: base64

CiAgPC9oZWFKPgogIDxib2R5PgogICAgPHA+VGhpcyBpcyB0aGUgYm9keSBvZiB0aGUgbWVzc2FnZS48L3A

--separador--

Tema 5 – Formatos SMTP

From: John Doe <jdoe@machine.example>
To: Mary Smith mary@example.net
Subject: Saying Hello
Date: Fri, 21 Nov 1997 09:55:06 -0600
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=separador

Indica que la palabra “separador”
separa formatos distintos de
contenido

--separador
Content-Type: text/plain

Primer tipo, es texto ASCII

This is the body of the message.

Aquí el texto codificado en ASCII

--separador
Content-Type: audio/basic
Content-Transfer-Encoding: base64

PGh0bWw+CiAgPGhlYWQ+CiAgPC9oZWFKPgogIDxib2R5PgogICAgPHA+VGhpYBpcyB0aGUg
Ym9keSBvZiB0aGUgbWVzc2FnZS48L3A+CiAgPC9ib2R5Pgo8L2h0bWw+Cg==

--separador
Content-Type: image/jpeg
Content-Transfer-Encoding: base64

CiAgPC9oZWFKPgogIDxib2R5PgogICAgPHA+VGhpYBpcyB0aGUgYm9keSBvZiB0aGUgbWVzc2FnZS48L3A

--separador--



Tema 5 – Formatos SMTP

From: John Doe <jdoe@machine.example>
To: Mary Smith mary@example.net
Subject: Saying Hello
Date: Fri, 21 Nov 1997 09:55:06 -0600
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=separador

Indica que la palabra “separador”
separa formatos distintos de
contenido

--separador
Content-Type: text/plain

Primer tipo, es texto ASCII

This is the body of the message.

Aquí el texto codificado en ASCII

--separador
Content-Type: audio/basic
Content-Transfer-Encoding: base64

Se acaba el formato anterior y aquí empieza uno nuevo
El nuevo es un audio codificado como base64

PGh0bWw+CiAgPGhlYWQ+CiAgPC9oZWFKPgogIDxib2R5PgogICAgPHA+VGhpcyBpcyB0aGUg
Ym9keSBvZiB0aGUgbWVzc2FnZS48L3A+CiAgPC9ib2R5Pgo8L2h0bWw+Cg==

--separador
Content-Type: image/jpeg
Content-Transfer-Encoding: base64

CiAgPC9oZWFKPgogIDxib2R5PgogICAgPHA+VGhpcyBpcyB0aGUgYm9keSBvZiB0aGUgbWVzc2FnZS48L3A

--separador--



Tema 5 – Formatos SMTP

From: John Doe <jdoe@machine.example>
To: Mary Smith mary@example.net
Subject: Saying Hello
Date: Fri, 21 Nov 1997 09:55:06 -0600
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=separador

Indica que la palabra “separador”
separa formatos distintos de
contenido

--separador
Content-Type: text/plain

Primer tipo, es texto ASCII

This is the body of the message.

Aquí el texto codificado en ASCII

--separador
Content-Type: audio/basic
Content-Transfer-Encoding: base64

Se acaba el formato anterior y aquí empieza uno nuevo
El nuevo es un audio codificado como base64

PGh0bWw+CiAgPGhlYWQ+CiAgPC9oZWFKPgogIDxib2R5PgogICAgPHA+VGhpYBpcyB0aGUg
Ym9keSBvZiB0aGUgbWVzc2FnZS48L3A+CiAgPC9ib2R5Pgo8L2h0bWw+Cg==

--separador
Content-Type: image/jpeg
Content-Transfer-Encoding: base64

Se acaba el formato anterior y aquí empieza uno nuevo
El nuevo es una imagen codificado como base64

CiAgPC9oZWFKPgogIDxib2R5PgogICAgPHA+VGhpYBpcyB0aGUgYm9keSBvZiB0aGUgbWVzc2FnZS48L3A

--separador--



Tema 5 – Formatos SMTP

From: John Doe <jdoe@machine.example>
To: Mary Smith mary@example.net
Subject: Saying Hello
Date: Fri, 21 Nov 1997 09:55:06 -0600
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=separador

Indica que la palabra “separador”
separa formatos distintos de
contenido

--separador
Content-Type: text/plain

Primer tipo, es texto ASCII

This is the body of the message.

Aquí el texto codificado en ASCII

--separador
Content-Type: audio/basic
Content-Transfer-Encoding: base64

Se acaba el formato anterior y aquí empieza uno nuevo
El nuevo es un audio codificado como base64

PGh0bWw+CiAgPGhlYWQ+CiAgPC9oZWFKPgogIDxib2R5PgogICAgPHA+VGhpYBpcyB0aGUg
Ym9keSBvZiB0aGUgbWVzc2FnZS48L3A+CiAgPC9ib2R5Pgo8L2h0bWw+Cg==

--separador
Content-Type: image/jpeg
Content-Transfer-Encoding: base64

Se acaba el formato anterior y aquí empieza uno nuevo
El nuevo es una imagen codificado como base64

CiAgPC9oZWFKPgogIDxib2R5PgogICAgPHA+VGhpYBpcyB0aGUgYm9keSBvZiB0aGUgbWVzc2FnZS48L3A

--separador--

Aquí se acaba el mensaje entero



Tema 5 – Tipos MIME más comunes

Type	Subtype	Description	File extensions
Application	postscript	Printable postscript document	.eps, .ps
	text	TEX document	.tex
Audio	midi	Musical Instrument Digital Interface	.midi, .mid
	realaudio	Progressive Networks sound	.ra, .ram
	wav	Microsoft sound	.wav
Image	gif	Grapical Interchange Format	.gif
	jpeg	Joint Photographic Experts Group	.jpeg, .jpg, .jpe
	png	Portable Network Graphics	.png
Model	vrml	Virtual Reality Modeling Language	.wrl
Text	html	Hypertext Markup Language	.html, .htm
	plain	Unformatted text	.txt
Video	avi	Microsoft audio video interleaved	.avi
	mpeg	Moving Picture Experts Group	.mpeg, .mpg
	quicktime	Apple QuickTime movie	.qt, .mov



Tema 5 – Extended SMTP (ESMTP)

- ▶ Hoy en día se usa la versión ESMTP del RFC 5321
- ▶ Como es retro compatible con SMTP, también esta versión se suele llamar SMTP
- ▶ Algunas diferencias
 - ▶ El primer comando usado indica la versión: HELO para SMTP y EHLO para ESMTP
 - ▶ Se puede añadir autenticación usando el comando AUTH para que el cliente se autentique antes de enviar el mensaje
 - ▶ Se puede añadir una capa de seguridad usando encriptación TLS para codificar los mensajes
 - ▶ Hay más comandos como SIZE, HELP, PIPELING, etc. definidos en el estándar ESMTP

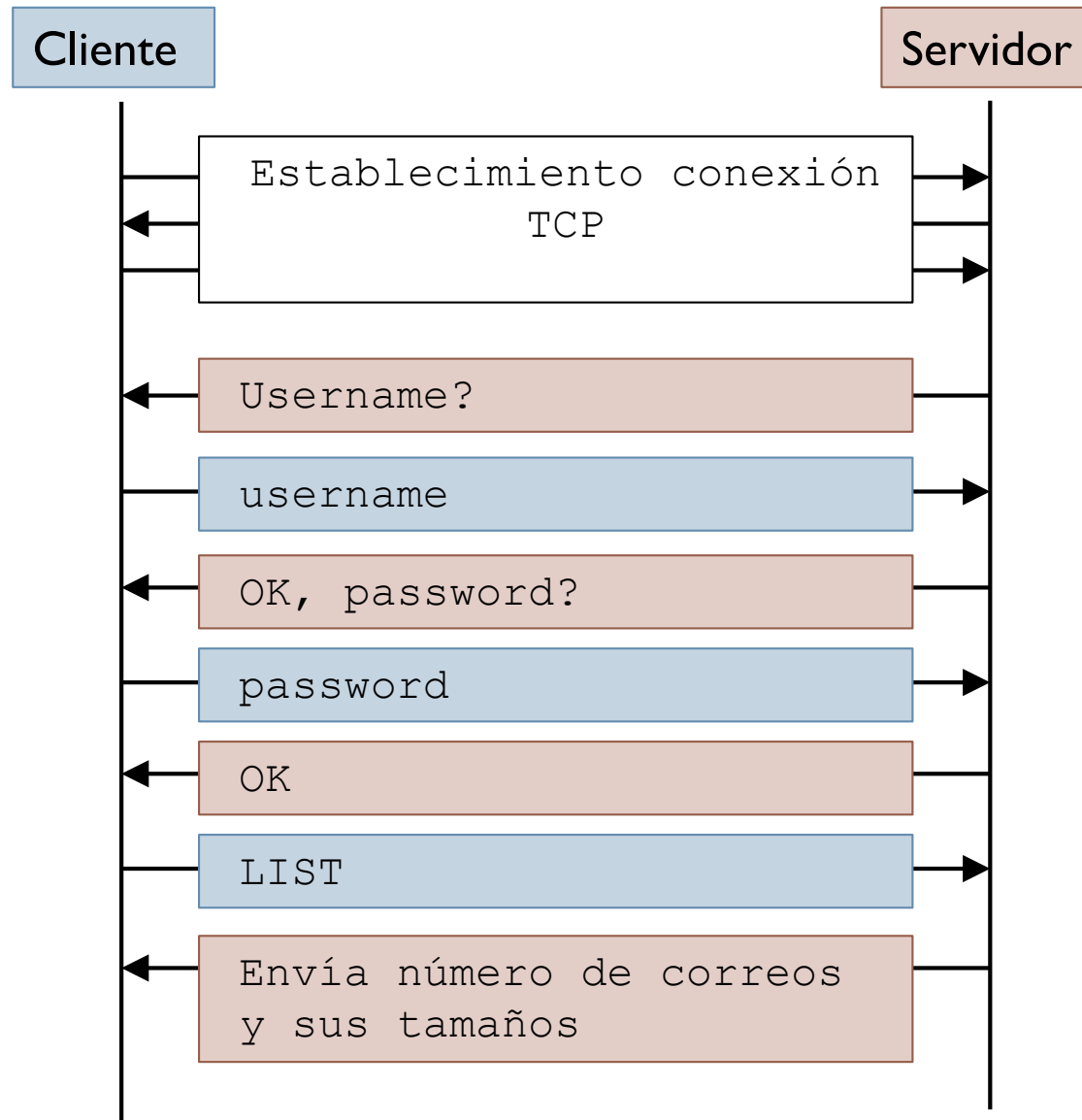


Tema 5 – POP

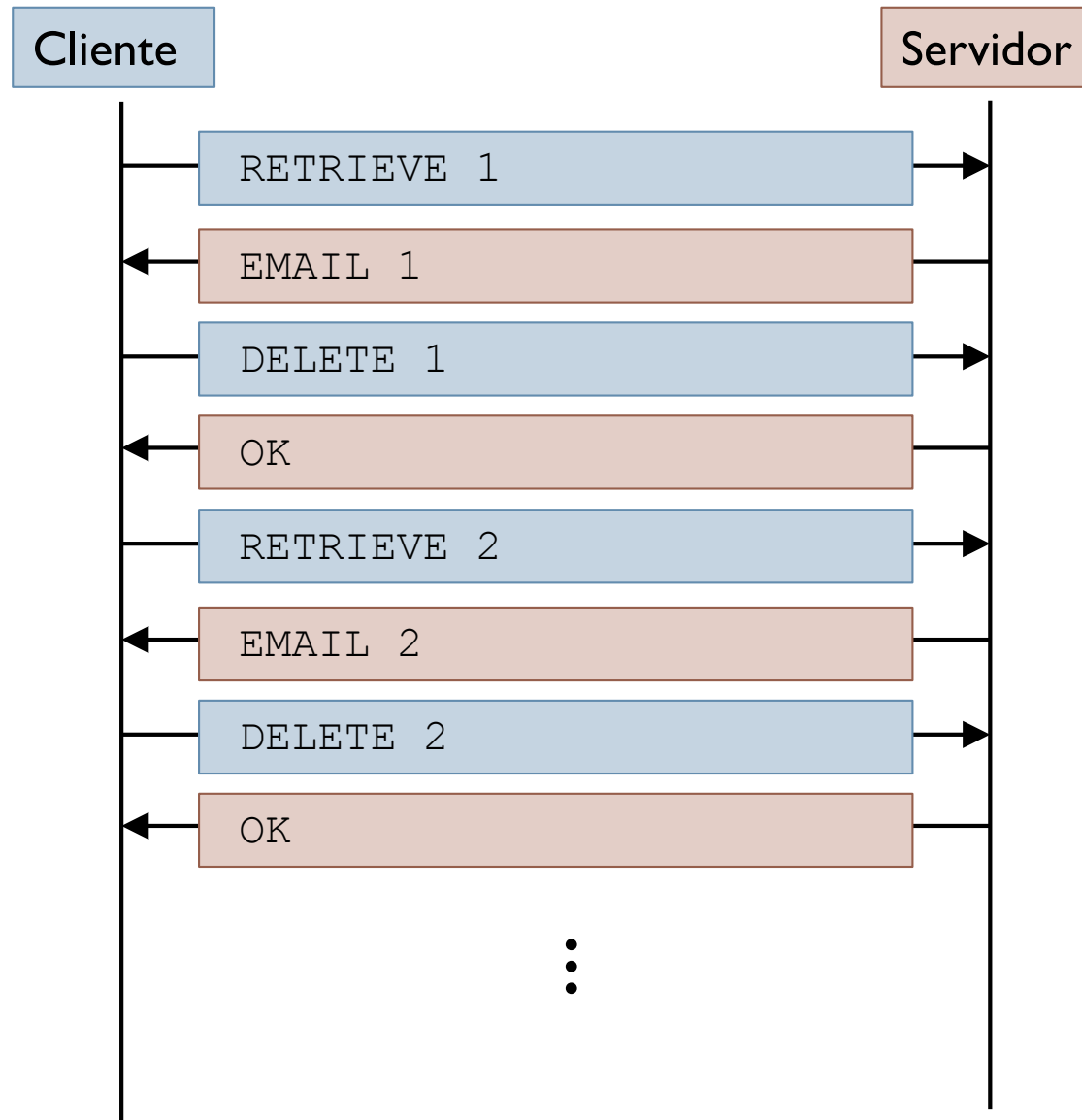
- ▶ Post Office Protocol
- ▶ RFC 918 (version 1)
- ▶ RFC 1081 (versión 3) -> actualizado en RFC 1939 y extensiones en RFC 2449 y autenticación en RFC 1734
- ▶ TCP puerto 110
- ▶ Características principales
 - ▶ El servidor guarda los mensajes de un usuario todos en un mismo sitio
 - ▶ El cliente al conectarse, baje todos los mensajes en local y estos (por defecto) se borran del servidor
 - ▶ Se puede configurar para que el servidor guarde una copia de los correos durante un cierto tiempo (que puede ser infinito)



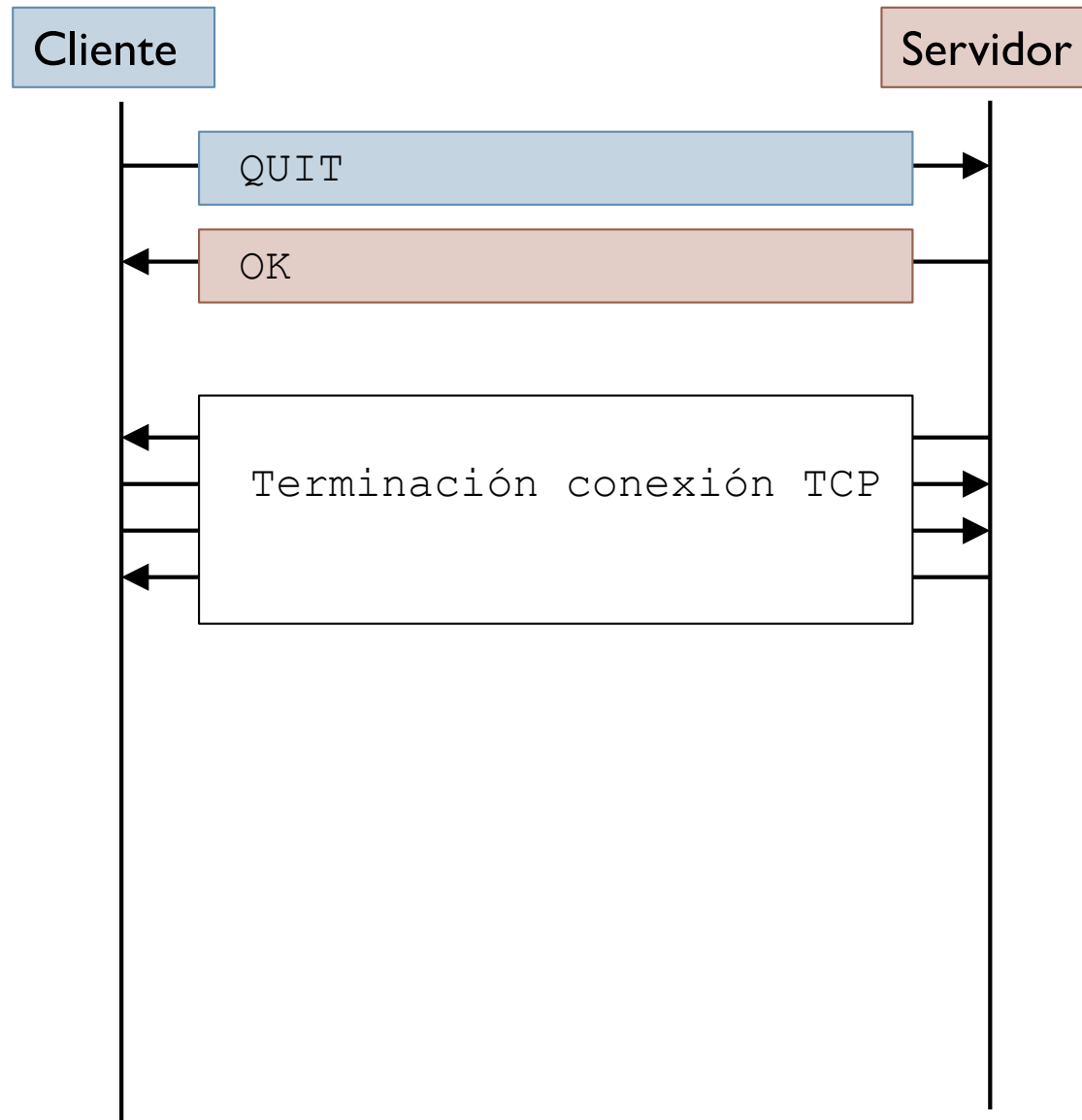
Tema 5 – POP



Tema 5 – POP



Tema 5 – POP



Tema 5 – IMAP

- ▶ Internet Message Access Protocol
- ▶ RFC 3501 (version 4)
- ▶ TCP puerto 143
- ▶ Características principales
 - ▶ Los correos se quedan guardados en el servidor hasta que el usuario no los borres explícitamente. De esta forma se puede acceder a los correos desde cualquier dispositivo ya que no se guardan en local
 - ▶ También hay la opción de crear un copia en local (de manera que no se necesita estar conectados a Internet siempre)
 - ▶ El usuario puede manejar los correos en el servidor. Por ejemplo puede crear carpetas, mover correos, bajar solo parte de un mensaje, etc.



Tema 5 – Basado en la web

- ▶ Típicamente un web server (HTTP), p.e. gmail, hotmail, yahoo, webmail, etc.
- ▶ Se accede a través de un cliente HTTP (Firefox, Safari, Chrome, etc.) directamente al propio servidor de correos
- ▶ Características principales
 - ▶ Los correos se quedan guardados en el servidor hasta que el usuario no los borres explícitamente. De esta forma se puede acceder a los correos desde cualquier navegador web ya que no se guardan en local
 - ▶ El usuario puede manejar los correos en el servidor. Por ejemplo puede crear carpetas, mover correos, bajar solo parte de un mensaje, etc.

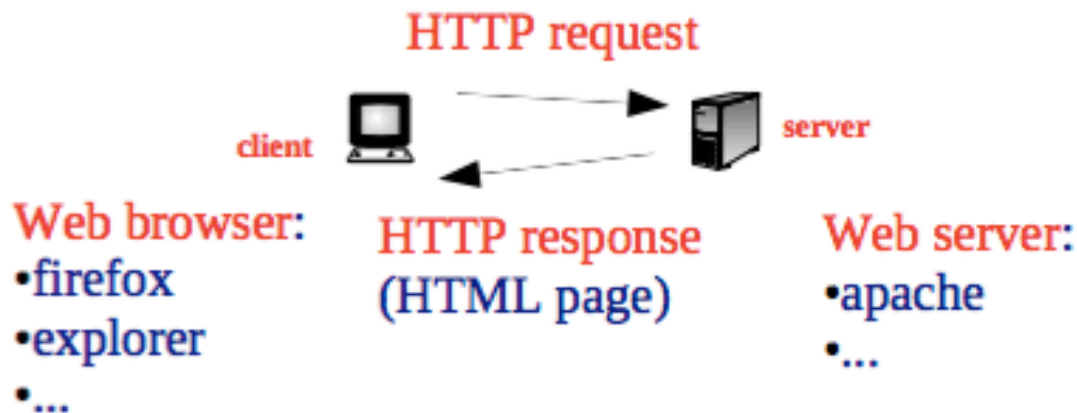


Tema 5 – Protocolos de nivel aplicación

- ▶ Introducción
- ▶ Protocolos del nivel aplicaciones
 - ▶ DNS
 - ▶ Email
 - ▶ **Web**
- ▶ Otro
 - ▶ HTML
 - ▶ XML
 - ▶ Charsets

Tema 5 – Web

- ▶ World Wide Web, www
 - ▶ Creado por Tim John Berners-Lee en 1989 y desarrollado en los 90s para proporcionar acceso simple a la información en Internet
- ▶ Componentes
 - ▶ Capa de transporte: TCP, puerto 80
 - ▶ Protocolo de aplicación: HyperText Transfer Protocol (HTTP)
 - ▶ Versión 1.0, RFC 1945 (Mayo 1996)
 - ▶ Versión 1.1, RFC 2616 (Junio 1999) → actualizado en 2014 RFC 7230-7235
 - ▶ Versión 2.0, RFC 7540 (Mayo 2015)
 - ▶ Lenguaje: HyperText Markup Language (HTML)



```
<!DOCTYPE html>
<html>
<!-- created 2010-01-01 -->
<head>
  <title>sample</title>
</head>
<body>
  <p>Voluptatem accusantium
  totam rem aperiam.</p>
</body>
</html>
```

HTML

Tema 5 – Web

- ▶ Una pagina web consiste de objetos o recursos
 - ▶ Un objeto puede ser un fichero HTML, una imagen JPEG, un applet Java, un fichero audio, etc.
 - ▶ Cada objeto es localizable a través de un URL (Uniform Resource Locator) RFC 1738

- ▶ Ejemplo de URL

`www.someschool.edu/someDept/pic.gif`

host name

path name (object)

- ▶ Típicamente hay un objeto base que es un fichero HTML que incluye luego varios otros objetos referenciados
- ▶ Ejemplo de referencia

```

```



Tema 5 – Web

- ▶ La sintaxis genérica de un URL es
 - ▶ `scheme:[//[user:password@]domain[:port]][/]path[?query][#fragment]`
 - ▶ Scheme: proposito del objeto, por ejemplo http, file, gopher, ftp, ...
 - ▶ User:password: opcional, algunos esquemas usan usuario y contraseña
 - ▶ Domain: nombre o dirección IP
 - ▶ Puerto: opcional, se puede definir un puerto diferente del por defecto
 - ▶ Path: localización interna de un objeto en el servidor
 - ▶ ?query: opcional, cadena de caracteres para localizar un objeto en una estructura compleja en el servidor
 - ▶ #fragment: opcional, identificador para localizar una posición concreta dentro de un objeto, por ejemplo el principio de una sección en un artículo



Tema 5 – Web

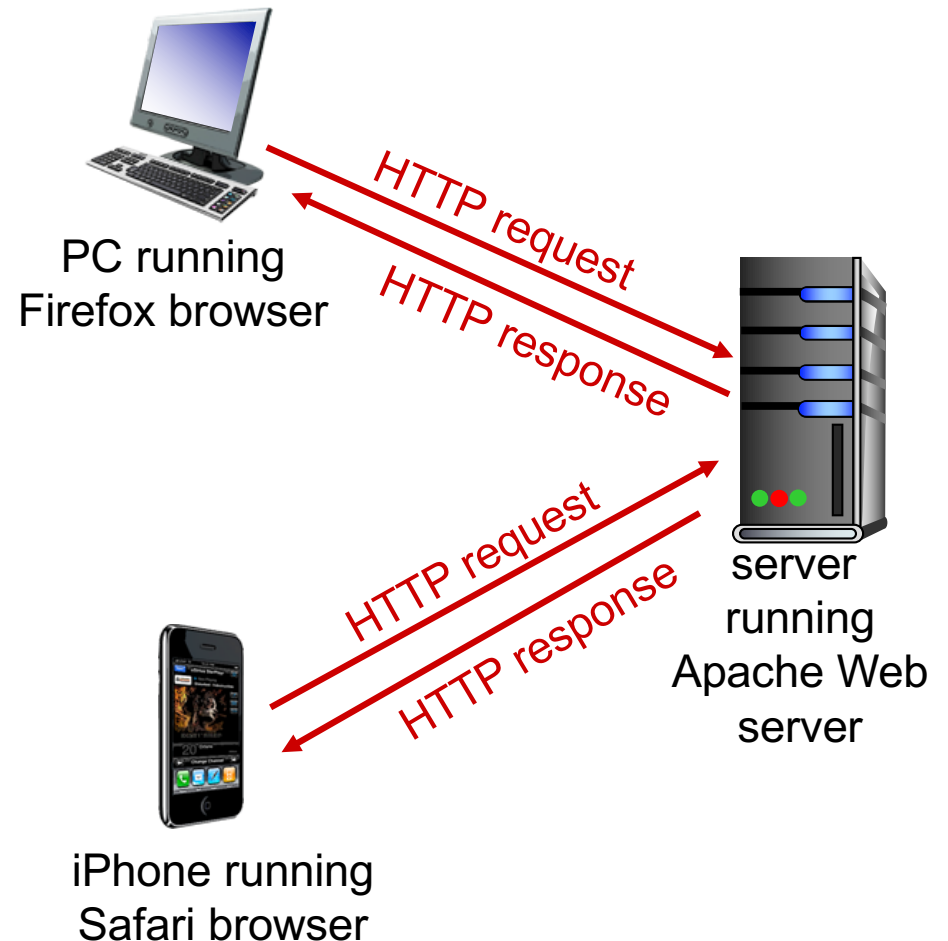
▶ Ejemplos

- ▶ <http://www.upc.edu>
- ▶ <http://tools.ietf.org/html/rfc1738>
- ▶ <http://147.83.2.135>
- ▶ <ftp://ftp.ee.lbl.gov/papers/vp-thesis/tcp.pdf>
- ▶ http://www.amazon.com/product/03879/refs9?pf_ra=ATVPD&pf_rd=07HR2

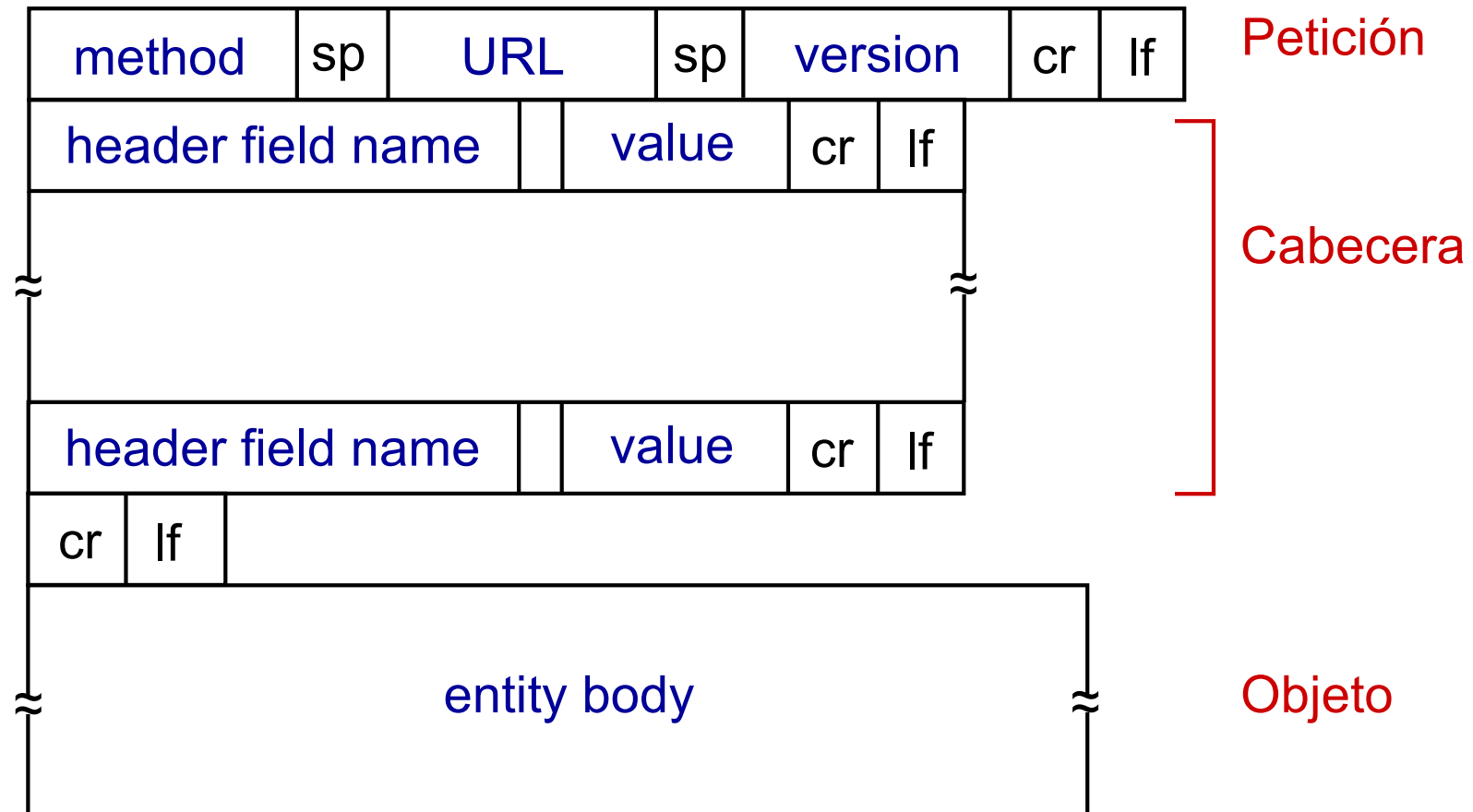


Tema 5 – Funcionamiento HTTP

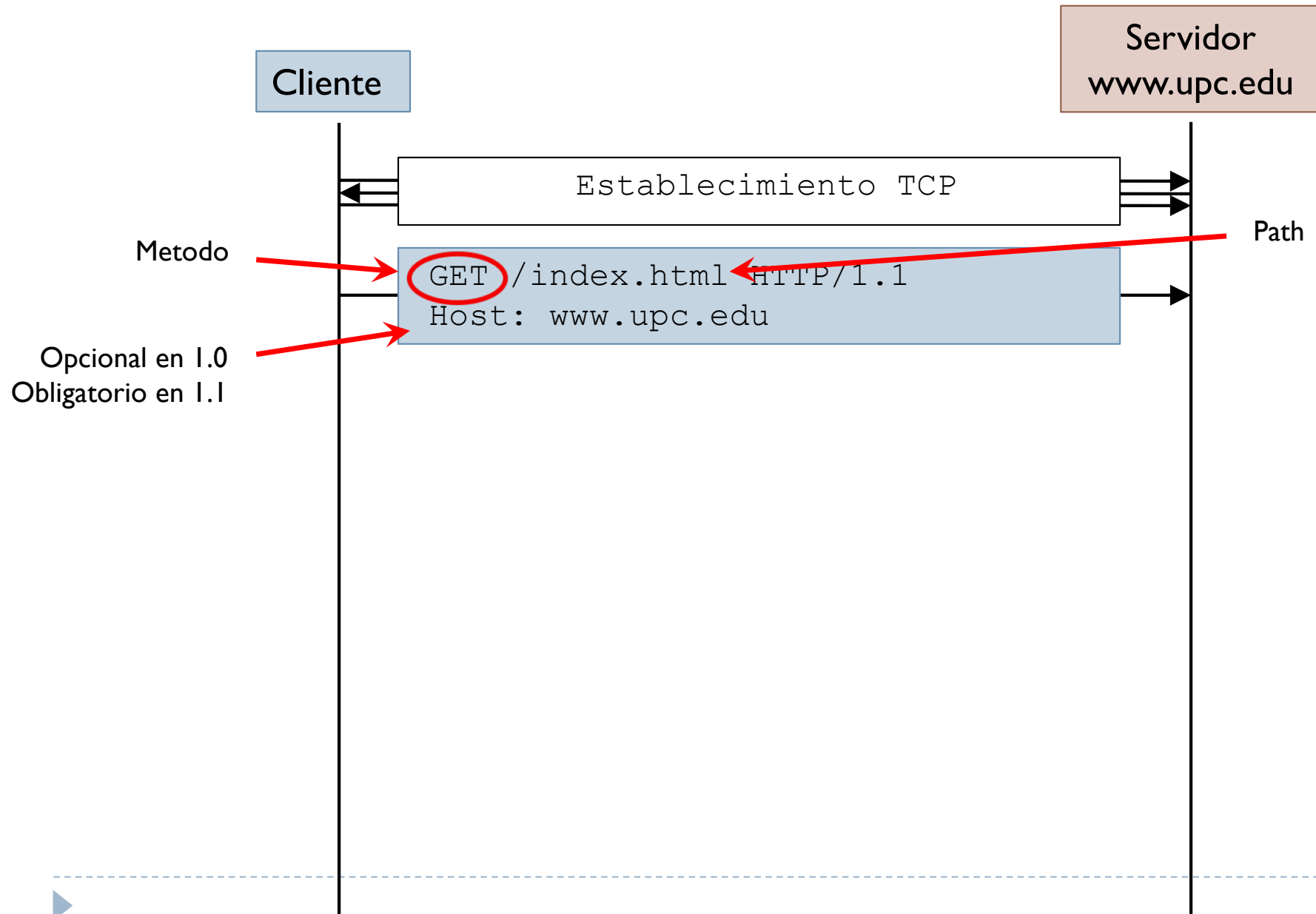
- ▶ Basado en la arquitectura cliente-servidor
 - ▶ Cliente: navegador que solicita, recibe (usando el protocolo HTTP) y muestra objetos Web
 - ▶ Servidor: servidor Web que envía (usando el protocolo HTTP) objetos en respuesta a peticiones de los clientes
- ▶ HTTP es “stateless”
 - ▶ Los servidores Web no mantienen información sobre las peticiones pasadas de los clientes
 - ▶ No mantienen estados



Tema 5 – Formato HTTP



Tema 5 – Petición HTTP



Tema 5 – Métodos HTTP

▶ Métodos

- ▶ GET: solicitar el contenido de una web
- ▶ POST: enviar datos a un servidor Web, por ejemplo el contenido de un formulario
- ▶ HEAD: para obtener solo la cabecera del objeto pero no el objeto mismo

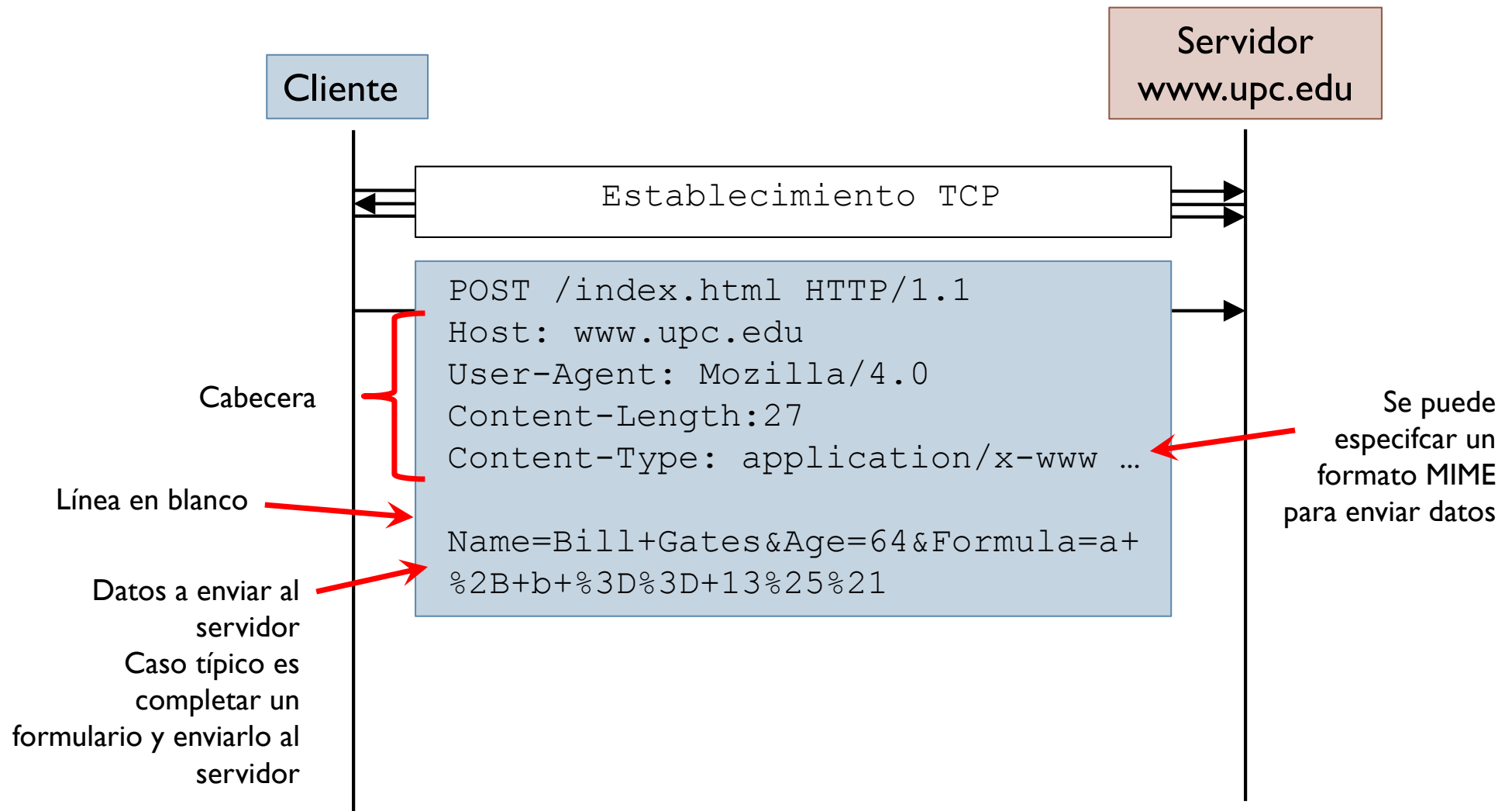
▶ HTTP 1.1 añade

- ▶ OPTIONS: para conocer los metodos aceptados por el servidor
- ▶ PUT: para insertar o substituir un determinado objeto
- ▶ DELETE: para eliminar un determinado objeto en el servidor web
- ▶ PATCH: para modificar parcialmente un objeto
- ▶ TRACE: para recibir ecos y saber si ha habido cambios
- ▶ CONNECT: generalmente usado para añadir una capa de seguridad SSL a la comunicación (HTTPS)

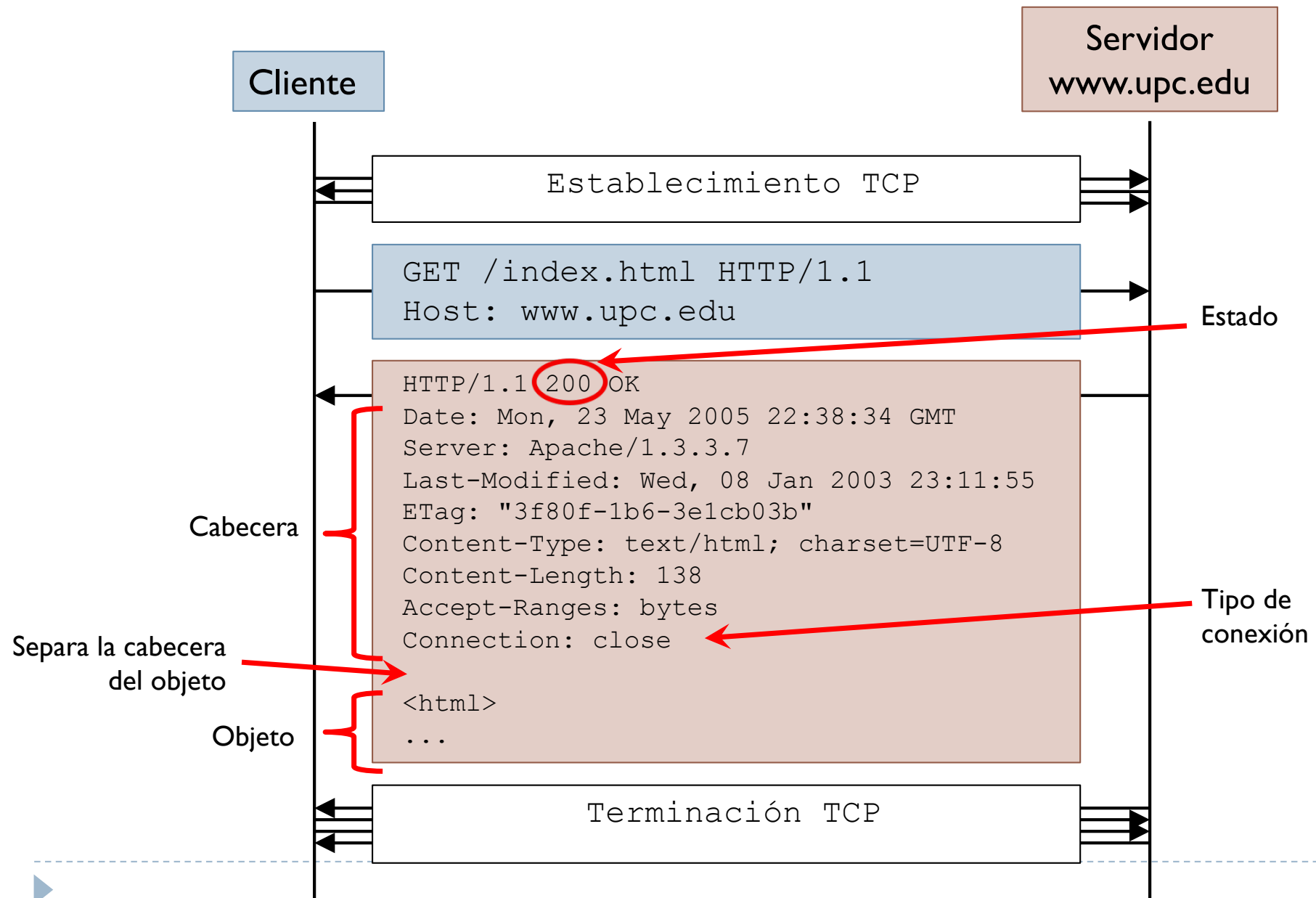


Tema 5 – Petición HTTP

► Para enviar datos



Tema 5 – Respuestas HTTP



Tema 5 – Respuestas HTTP

- ▶ El servidor contesta con una primera línea que indica el estado

200 OK

- ▶ request succeeded, requested object later in this msg

301 Moved Permanently

- ▶ requested object moved, new location specified later in this msg
(Location:)

400 Bad Request

- ▶ request msg not understood by server

404 Not Found

- ▶ requested document not found on this server

505 HTTP Version Not Supported



Tema 5 – Tipos de conexión HTTP

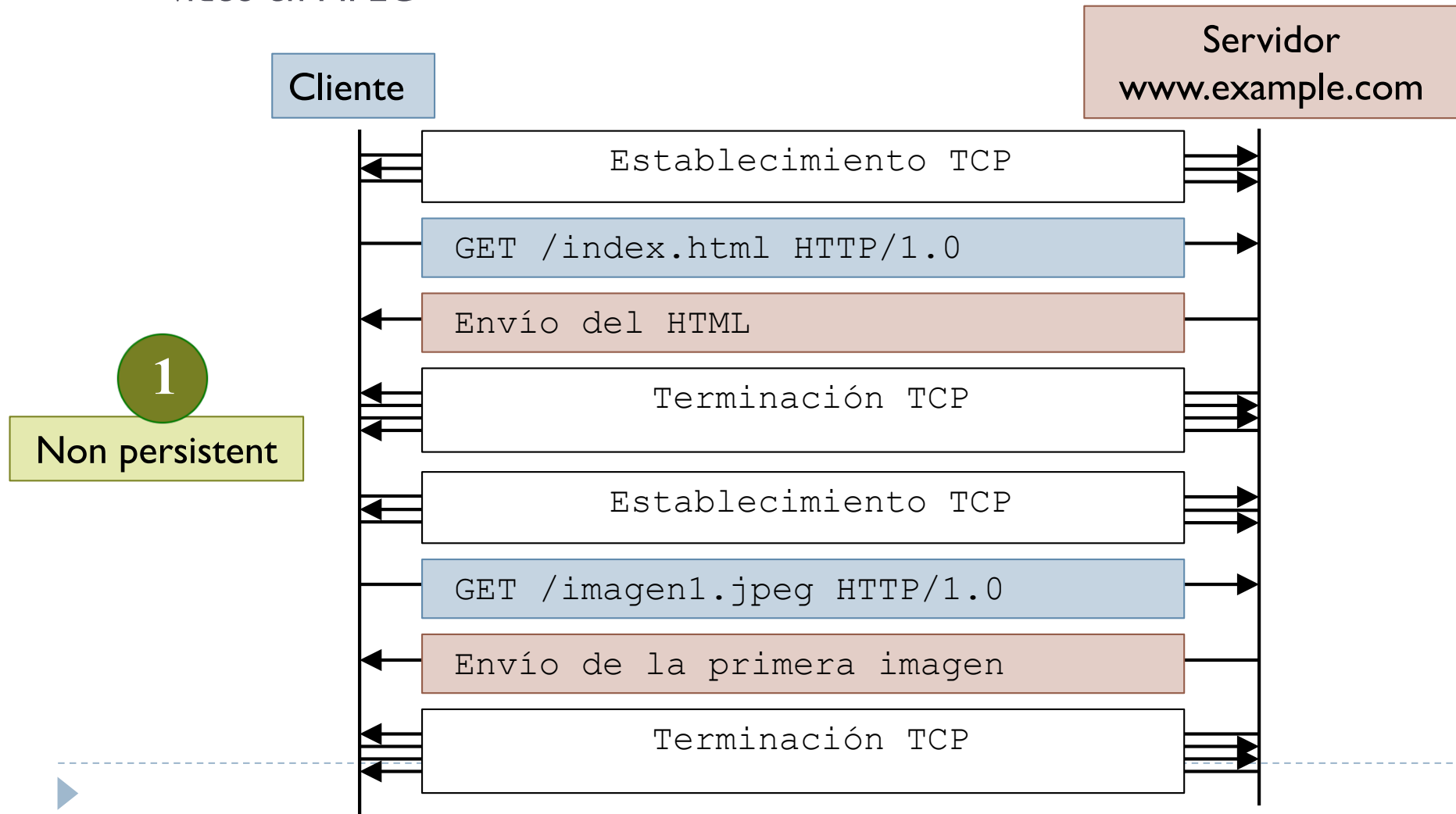
- ▶ **No persistente (por defecto en HTTP 1.0)**
 - ▶ El servidor envía el objeto solicitado y cierra la conexión
 - ▶ Si hay más objetos en una página, hay que abrir una conexión para cada uno
 - ▶ Connection: close
- ▶ **Persistente (por defecto en HTTP 1.1)**
 - ▶ El servidor mantiene la conexión abierta durante un tiempo determinado
 - ▶ Si hay más objetos, se pueden pedir uno después del otro (secuencialmente)
 - ▶ Connection: keep-alive
- ▶ **Persistente con pipeling (solo en HTTP 1.1 pero uso no recomendado)**
 - ▶ El servidor mantiene la conexión abierta durante un tiempo determinado y el cliente puede pedir nuevos objetos en el momento en que encuentra nuevas referencias en el objeto que se está bajando aunque la descarga no se haya completado (transmisiones en paralelo)
 - ▶ Si página con muchos objetos distintos, sobrecarga de GET que el servidor debe servir y contestar
 - ▶ Connection: keep-alive



Tema 5 – Tipos de conexión HTTP

► Ejemplo

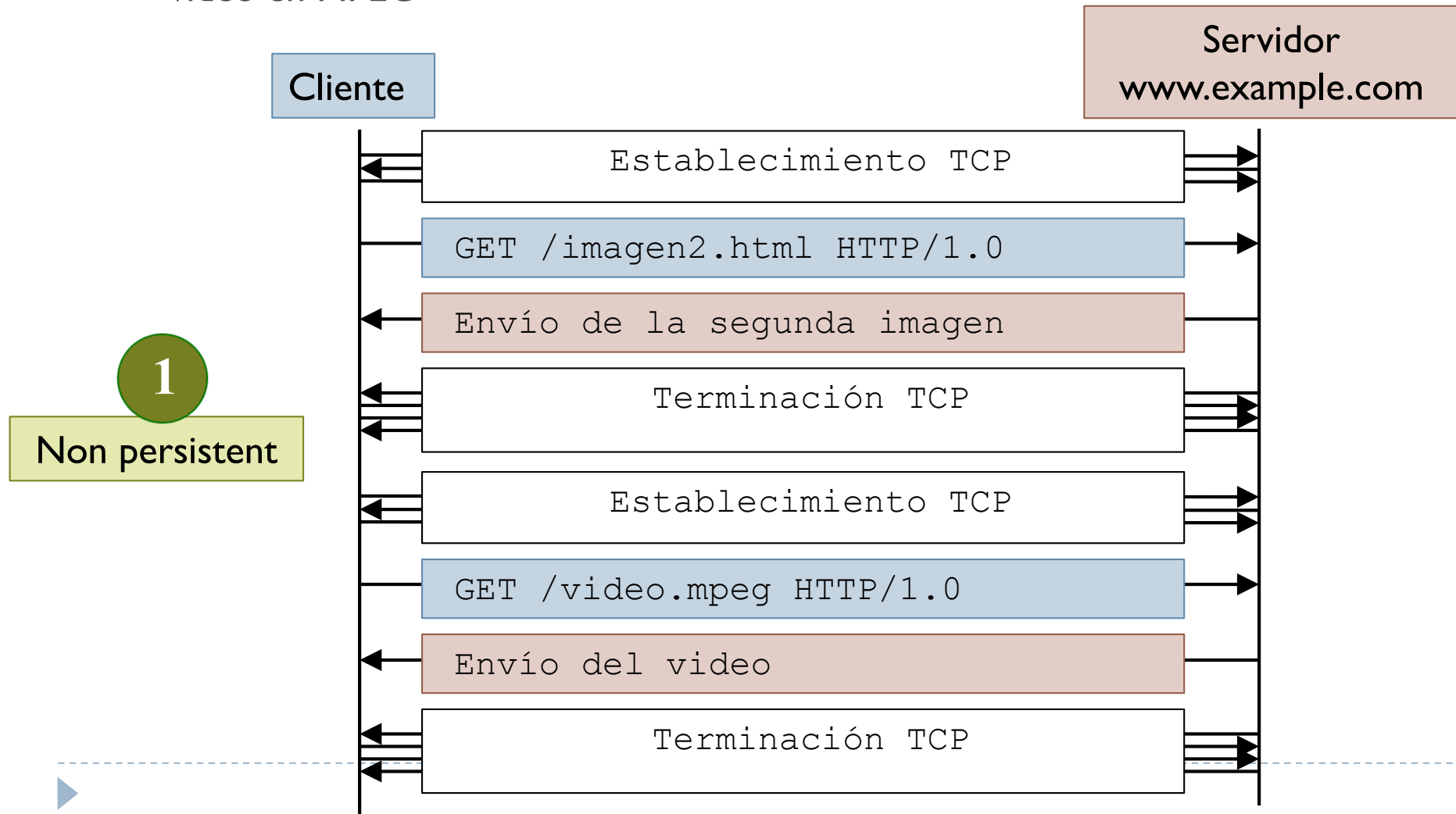
- La web `www.example.com` contiene 1 texto en HTML, 2 imágenes en JPEG y 1 video en MPEG



Tema 5 – Tipos de conexión HTTP

► Ejemplo

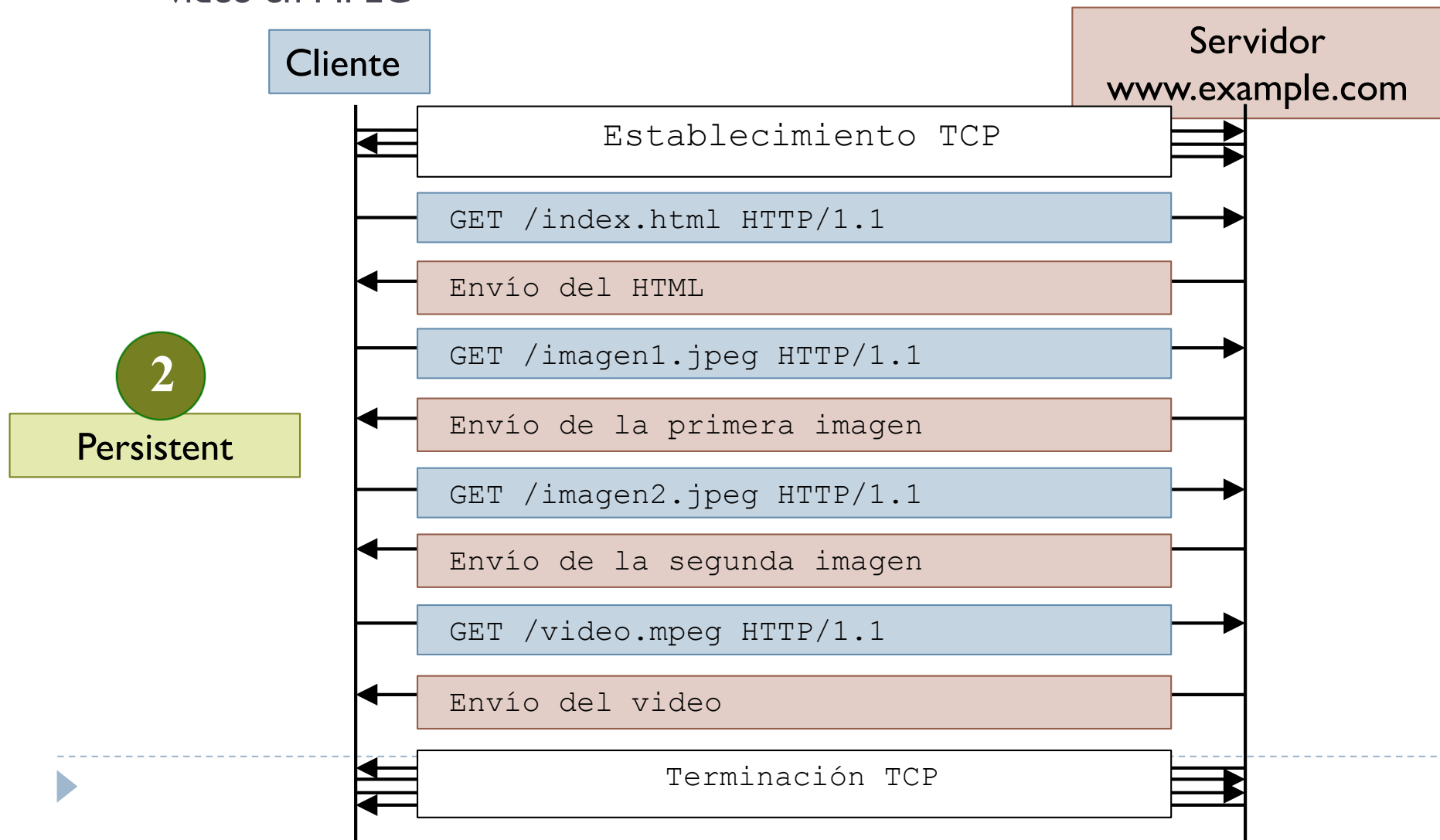
- La web `www.example.com` contiene 1 texto en HTML, 2 imágenes en JPEG y 1 video en MPEG



Tema 5 – Tipos de conexión HTTP

► Ejemplo

- La web `www.example.com` contiene 1 texto en HTML, 2 imágenes en JPEG y 1 video en MPEG



Tema 5 – HTTP caching

Objetivo: no enviar objetos si ya se han visitado previamente

- ▶ El cliente almacena las paginas web visitadas en una cache local
- ▶ Cuando el usuario quiere ver una pagina web, el cliente mira su cache local para ver si ya tiene esta página
 - ▶ Si la tiene, envía un Conditional GET al servidor
 - ▶ Este petición le dice al servidor que le envíe la pagina solo si es diferente de la que ya tiene
 - ▶ Para saber si es diferente se usan los campos Etag o Date
- ▶ Ejemplo

GET /index.html HTTP/1.1

Host: www.example.com

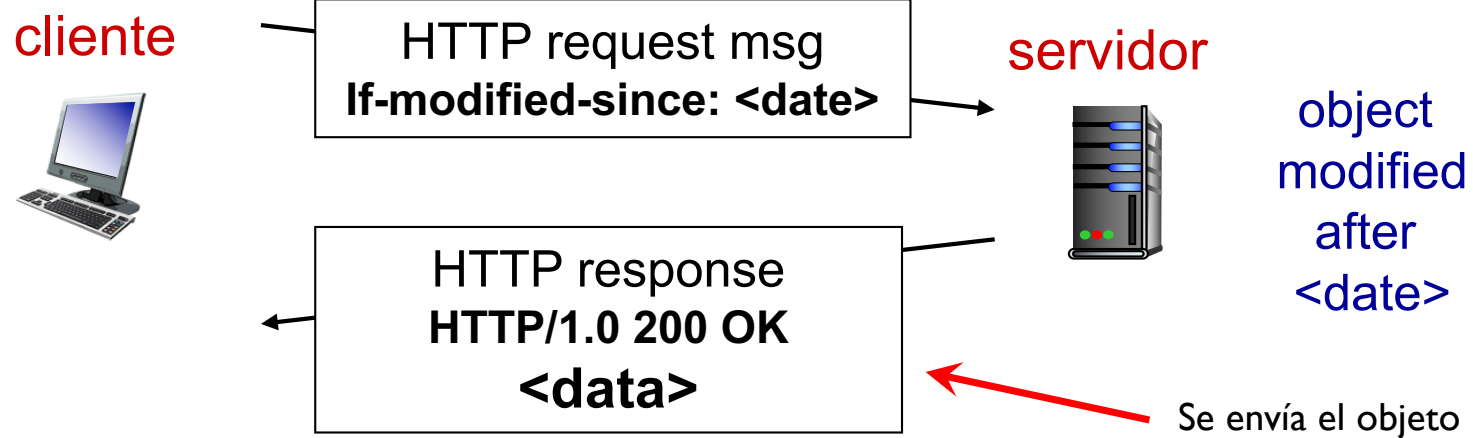
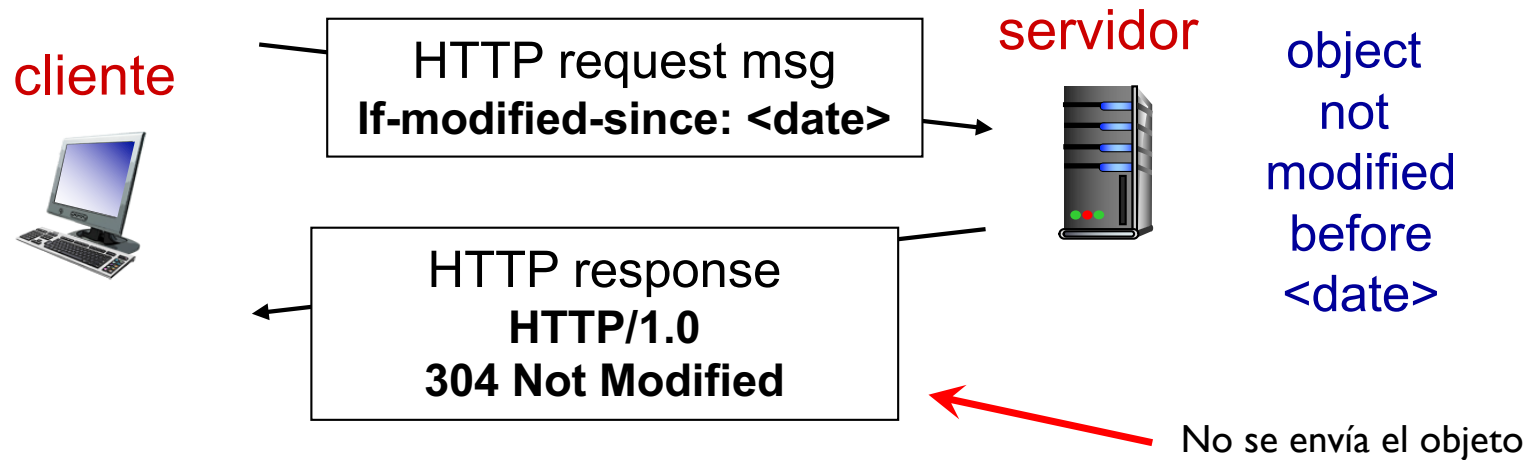
If-Modified-Since: October 21, 2002 4:57 PM

If-None-Match: "686897696a7c876b7e"

← ETag



Tema 5 – HTTP caching



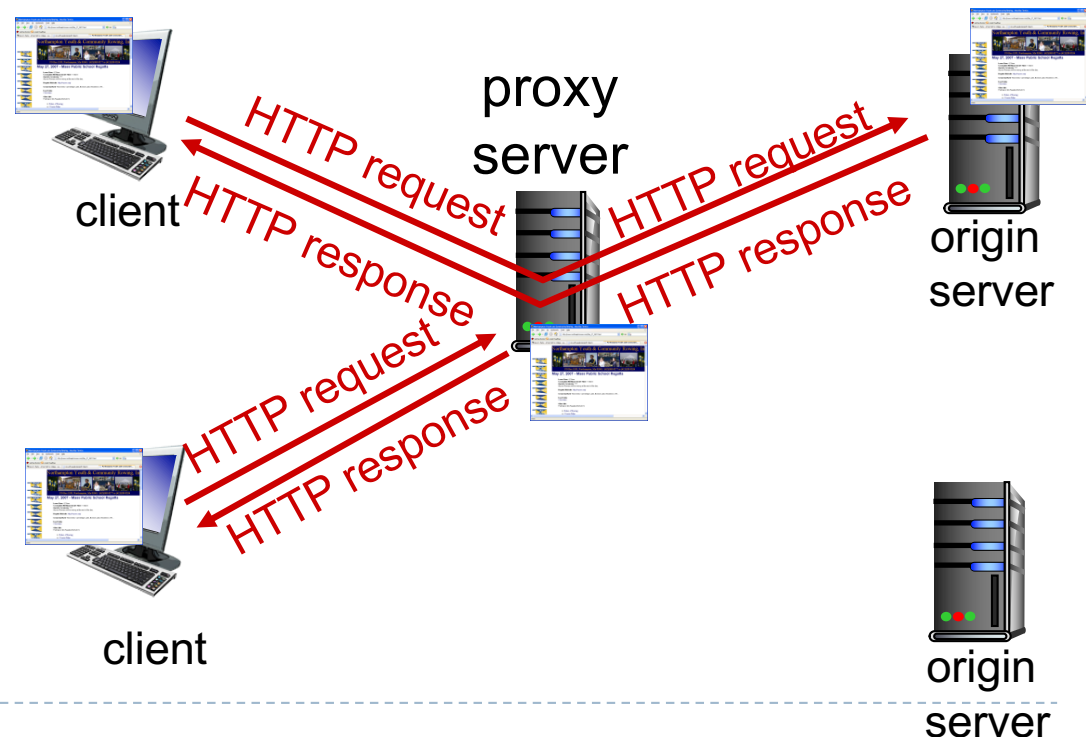
Tema 5 – HTTP proxy

Objetivo: contestar a los clientes sin que estos accedan al servidor

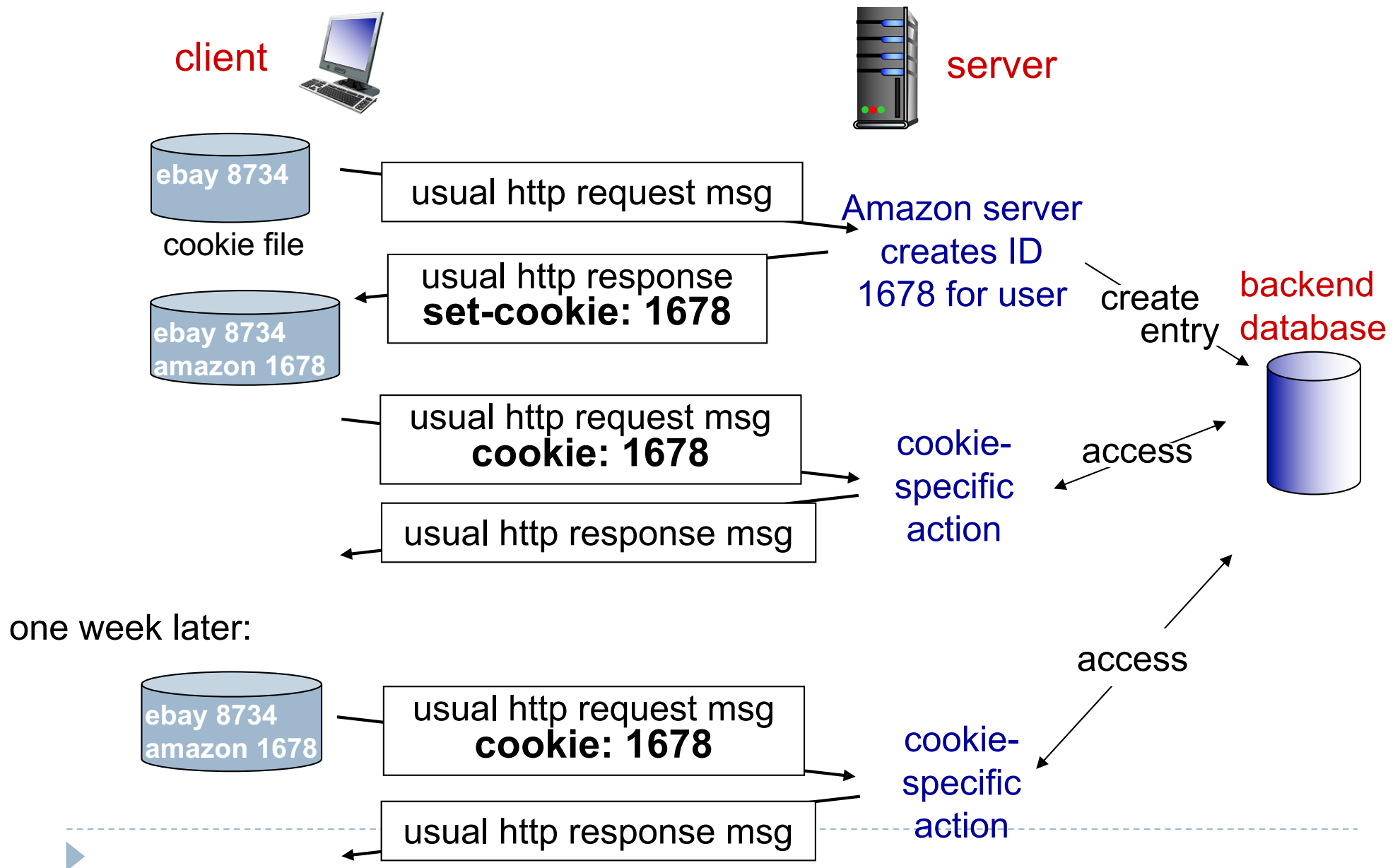
- ▶ Se usa un proxy como intermediario
- ▶ Los clientes se comunican con el proxy
- ▶ El proxy accede al servidor y almacena las paginas web solicitada y sus objetos temporáneamente en su cache local

- ▶ **Ventajas**

- ▶ Seguridad
- ▶ Rapidez
- ▶ Menor carga en los servidores
- ▶ Servidor sin acceso a Internet (@IP privada)



Tema 5 – HTTP cookies – keeping states



Tema 5 – HTTP cookies

Objetivos

- Facilitar autorización
- Mantener carritos de la compra
- Recomendar productos
- Mantener sesiones abiertas (Web e-mail)

aside

Cookies y privacidad:

- cookies permite a los sitios web aprender mucho de los usuarios
- Pueden llegar a conocer mucho de los usuario, nombre y apellido, donde viven, que hacen, por donde se mueven, que prefieren, etc.



Tema 5 – Protocolos de nivel aplicación

- ▶ **Introducción**
- ▶ **Protocolos del nivel aplicaciones**
 - ▶ DNS
 - ▶ Email
 - ▶ Web
- ▶ **Otro**
 - ▶ HTML
 - ▶ XML
 - ▶ Charsets

Tema 5 – HTML

- ▶ Hyper-Text Markup Language
- ▶ En el 1986 ISO estandariza el Standard Generalized Markup Language (SGML)
 - ▶ SGML introduce la sintaxis `<>` y se usa en documentaciones muy largas
- ▶ Tim Berners-Lee define el HTML en 1989 inspirado en el SGML
 - ▶ Su objetivo es mostrar documentos de texto formateado y con hyperlinks (enlaces a otros documentos) en navegadores web
- ▶ Basado en tags
- ▶ Ejemplo

```
<html>
<head>
  <title>Basic html document</title>
</head>
<body>
  <h1><font color="red">First Heading</font></h1>
  <p>first paragraph.</p>
</body>
</html>
```

First Heading

first paragraph.

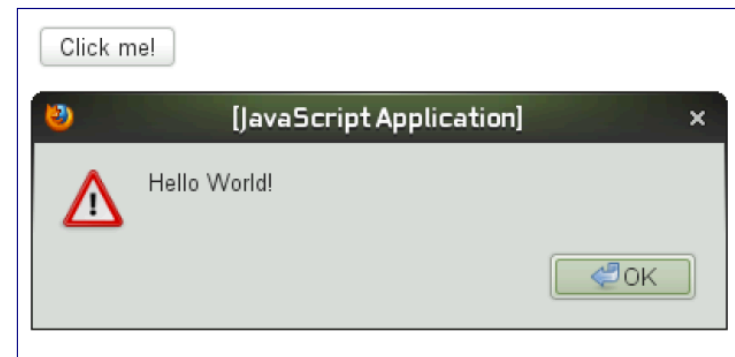
Terminology:

- element
- attribute
- text

Tema 5 – HTML

- ▶ **Forms:** el documento acepta inputs del usuario que se envían al server
- ▶ **Scripting:** permite ejecutar programas. El programa se ejecuta en la maquina del cliente cuando se carga el documento o en el momento de hacer un click a un enlace
- ▶ Ejemplo javascript

```
<html>
<head>
<script type="text/javascript">
  function displaymessage() {
    alert("Hello World!");
  }
</script>
</head>
<body>
  <form>
    <input type="button"
      value="Click me!" onclick="displaymessage()" />
  </form>
</body>
</html>
```

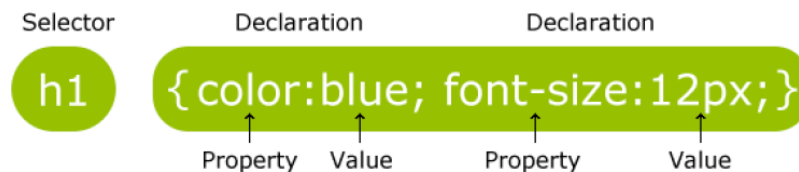


Tema 5 – HTML

► Cascading Style Sheets (CSS)

- Permite separar el contenido del documento y la forma de presentación de este, como las capas o layouts, los colores y las fuentes
- De esta forma varios documentos HTML pueden usar el mismo CSS para definir su presentación
- Si se quiere cambiar de estilo, solo se cambia el CSS y no los HTML.

► Sintaxis CSS



Source: <http://www.w3schools.com/xml/>

► Ejemplo

```
h1 {color:red; font-size:20px;}
p {margin-left:20px; color:blue; font-size:18px;}
```

fichero mystyle.css

```
<html>
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css" />
</head>
<body>
  <h1>First Heading</h1>
  <p>first paragraph.</p>
</body>
</html>
```

First Heading

first paragraph.

Tema 5 – Protocolos de nivel aplicación

- ▶ Introducción
- ▶ Protocolos del nivel aplicaciones
 - ▶ DNS
 - ▶ Email
 - ▶ Web
- ▶ Otro
 - ▶ HTML
 - ▶ XML → autoaprendizaje para las practicas no presenciales
 - ▶ Charsets

Tema 5 – Charsets



- ▶ Dificultad a la hora de crear un programa para “todo el mundo”
- ▶ Muchos idiomas, muchos alfabetos, muchas monedas, muchas maneras de indicar la fecha y la hora, etc.
 - ▶ Un mismo idioma puede tener además variantes, por ejemplo es_ES, es_CO
 - ▶ Un mismo alfabeto puede estar compartido entre varios idiomas, por ejemplo Catalán y Francés
- ▶ Se necesita que un software sea “adaptable”
 - ▶ Internationalization (i18n): adaptación de un software a diferentes idioma sin necesidad de modificar el código
 - ▶ Localization (L10n): proceso mediante el cual un producto internacionalizado se configura para una determinada región, aprovechando las opciones que la internacionalización previa de este producto ha permitido

Tema 5 – Charsets

- ▶ Típicamente se negocia el idioma y charset a usar en los protocolos de nivel aplicación
- ▶ Por ejemplo, en HTTP

El cliente indica en la cabecera los idiomas y los charsets aceptados

Accept-Language: es, ca, en-gb, en
Accept-Charset: iso-8859-15, unicode-9-0
...



Content-Language: ca
Content-Type: text/html; charset=utf-8
...

El Servidor indica en la cabecera el idioma y el charset del contenido

Tema 5 – Charsets

- ▶ **Ejemplo de charset**

- ▶ ISO 8859-1 (Latin-1): 190 caracteres + control = 256 combinaciones → 8 bits

- ▶ **Variantes**

- ▶ ISO 8859-15 extends -1 + ÿ, €
 - ▶ ISO 8859-2 (Central European)
 - ▶ ISO 8859-4 (North European)
 - ▶ ISO 8859-5 (Cyrillic)
 - ▶ ISO 8859-6 (Arabic)
 - ▶ ISO 8859-7 (Greek)
 - ▶ ISO 8859-8 (Hebrew)
 - ▶ ISO 8859-9 (Turkish, Kurdish)
 - ▶ ISO 8859-11 (Thai)

Tema 5 – Unicode

- ▶ Universal Coded Character Set
- ▶ Todos los caracteres de todos los idiomas + símbolos matemáticos + emoticonos + símbolos musicales + ... = Universal Character Set (ucs)
 - ▶ Actualmente 136,755 caracteres (versión 10.0, Junio 2017)
 - ▶ Hay 1,114,112 posiciones posibles, desde el U+0 hasta U+0FFFF
 - ▶ U+0030 es el 0, U+0041 es A, U+00C7 es Ç, U+0BB6 es 600T, etc.
- ▶ Se codifican en una secuencia de bits de acuerdo a método (algoritmo) de encoding
 - ▶ Universal Transformation Format (UTF)
 - ▶ UTF-8: la codificación dominante en la web desde 2009, 90.3% en Noviembre 2017. Compatible con ASCII

Tema 5 – UTF-8

- ▶ Codificación orientada a byte con símbolos de longitud variable, según la tabla

Number of bytes	Bits for code point	First code point	Last code point	Byte 1	Byte 2	Byte 3	Byte 4
1	7	U+0000	U+007F	0xxxxxxx			
2	11	U+0080	U+07FF	110xxxxx	10xxxxxx		
3	16	U+0800	U+FFFF	1110xxxx	10xxxxxx	10xxxxxx	
4	21	U+10000	U+10FFFF	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx

- ▶ Ejemplo
- ▶ 0 es U+0030
 - 1 byte (entre U+0000 y U+007F)
 - la codifica a usar es 0xxx xxxx → se necesitan 7 bits
 - 30 en binario es **11 0000** → Se añaden 0 delante **011 0000**

Se envía: **0011 0000**

Tema 5 – UTF-8

- ▶ Codificación orientada a byte con símbolos de longitud variable, según la tabla

Number of bytes	Bits for code point	First code point	Last code point	Byte 1	Byte 2	Byte 3	Byte 4
1	7	U+0000	U+007F	0xxxxxxx			
2	11	U+0080	U+07FF	110xxxxx	10xxxxxx		
3	16	U+0800	U+FFFF	1110xxxx	10xxxxxx	10xxxxxx	
4	21	U+10000	U+10FFFF	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx

- ▶ Ejemplo
- ▶ ¢ es U+00A2
 - 2 bytes (entre U+0080 y U+07FF)
 - la codifica a usar es 110x xxxx 10xx xxxx → se necesitan 11 bits
 - A2 en binario es 1010 0010 → 0 delante 000 1010 0010

Se envía: 1100 0010 1010 0010

Tema 5 – UTF-8

- Codificación orientada a byte con símbolos de longitud variable, según la tabla

Number of bytes	Bits for code point	First code point	Last code point	Byte 1	Byte 2	Byte 3	Byte 4
1	7	U+0000	U+007F	0xxxxxxx			
2	11	U+0080	U+07FF	110xxxxx	10xxxxxx		
3	16	U+0800	U+FFFF	1110xxxx	10xxxxxx	10xxxxxx	
4	21	U+10000	U+10FFFF	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx

- Ejemplo
- € es U+20AC
 - 3 bytes (entre U+0800 y U+FFFF)
 - la codifica a usar es 1110 xxxx 10xx xxxx 10xx xxxx → 16 bits
 - 20AC en binario es 10 0000 1010 1100 → 0010 0000 1010 1100

Se envía: 1110 0010 1000 0010 1010 1100

Xarxes de Computadors

Tema 5 – Protocolos de nivel aplicació

Tema 5. Problema 1

Un client web accedeix a la pàgina “`www.serveiweb.org/index.htm`”. Aquesta pàgina conté una imatge de capçalera incrustada, tres imatges allotjades en un server extern, un anunci allotjat en un altre server i una imatge gran allotjada en el server d'imatges.

Considera les dades següents:

- ▶ server DNS: $RTT = 10ms$; Suposa que utilitza UDP per fer les consultes al DNS
- ▶ server `serveiweb.org`: $RTT = 30ms$; conté la pàgina `index.htm` (cap en un segment de dades) i la imatge capçalera (1 segment de dades)
- ▶ server d'imatges: $RTT = 50ms$; conté tres imatges petites (1 segment/imatge) i una imatge gran (4 segments)
- ▶ server de l'anunci: $RTT = 200ms$; l'anunci (cap en 1 segment de dades)

Considera que s'utilitza HTTP persistent sense “pipelining”, el client web només obre una connexió TCP a cada server. Detalla la seqüència de transaccions i el temps de cada una. No cal tenir en compte les desconnexions de TCP. Fes un petit diagrama de temps per a cada transacció.

Calcula el temps total de descàrrega de la pàgina. Indica les suposicions que facis.



Tema 5. Problema 2

Consider that a user with email peter@bell.com wants to send an e-mail to dimitri@bell.com and sandra@hpc.es. Determine:

- a) Which messages (number, type and content) have to be sent between the host of the user and its mail server.
- b) Which messages (number, type and content) have to be sent between the host mail server and the destination mail servers.



Tema 5. Problema 2b

Consider that a user with email `dustin@facebook.com` wants to send an e-mail to `dimitri@bell.com`, a copy to `mark@facebook.com` and a blind copy to `steve@apple.com`. Determine:

- a) Which messages (number, type and content) have to be sent between the host of the user and its mail server.
- b) Which messages (number, type and content) have to be sent between the host mail server and the destination mail servers.



Tema 5. Problema 3

Dos usuarios A i B utilitzen el correu electrònic des de casa seva. A utilitza el servei de correu de BigMail.org i B el de SmallMail.org. La figura mostra la configuració de la xarxa i dels clients de correu.

- ▶ L'usuari A envia un correu electrònic a l'usuari B (b@smallmail.org). Indica la seqüència de missatges que intercanvia A amb els diferents equips per enviar el missatge des d'A fins al seu server BigMail.
- ▶ El server de correu BigMail envia el missatge al server de correu SmallMail. Indica les interaccions, entre quins equips es fan i quin protocol utilitzen.
- ▶ L'usuari B llegeix el missatge de correu del seu server. Indica les interaccions, entre quins equips es fan i quin protocol utilitzen

