Computer Networks

Grado en Ingeniería Informática

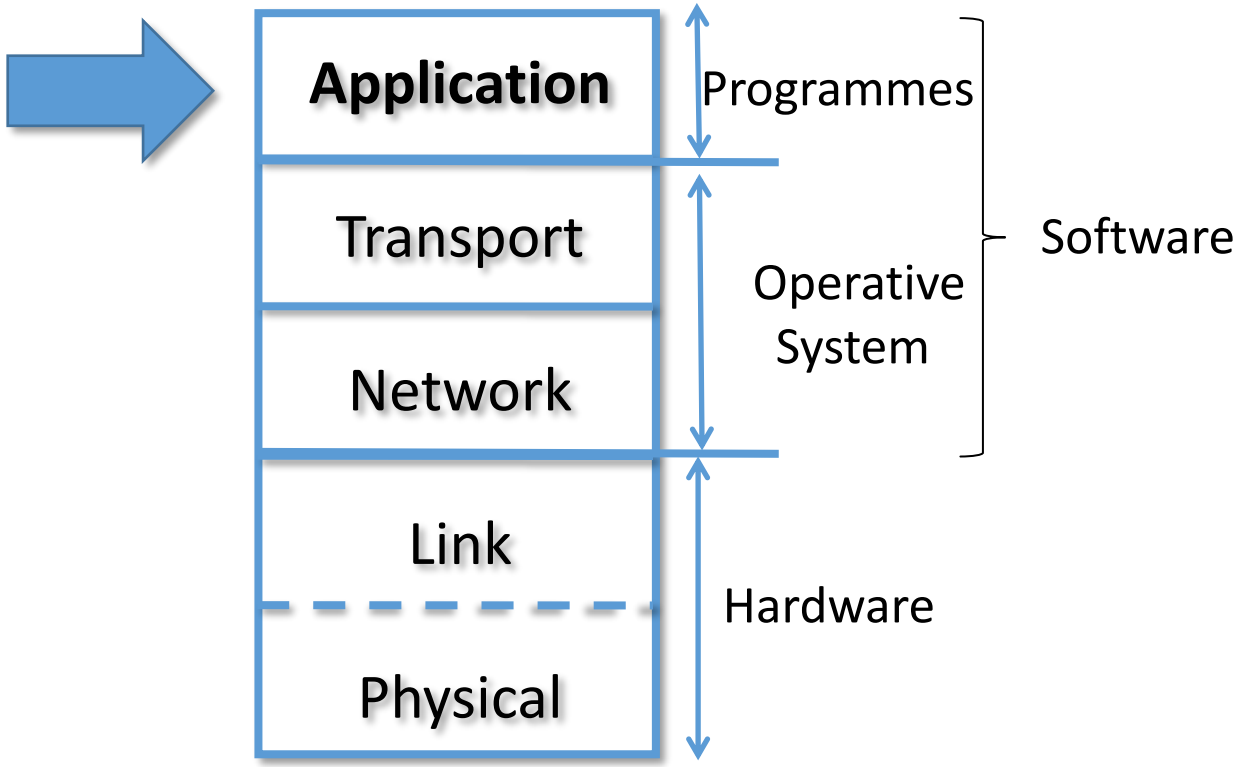
Departamento de
Tecnología Electrónica

Contents

Computer Networks
Lesson 2

The Application Layer

Departamento de
Tecnología Electrónica


Application
Transport
Network
Link
Physical
Programmes
Operative System
Software
Hardware

Lesson 2: The Application Layer

Objectives

- To understand the application layer of the TCP/IP model and the OSI model
- To know some basic protocols of this layer

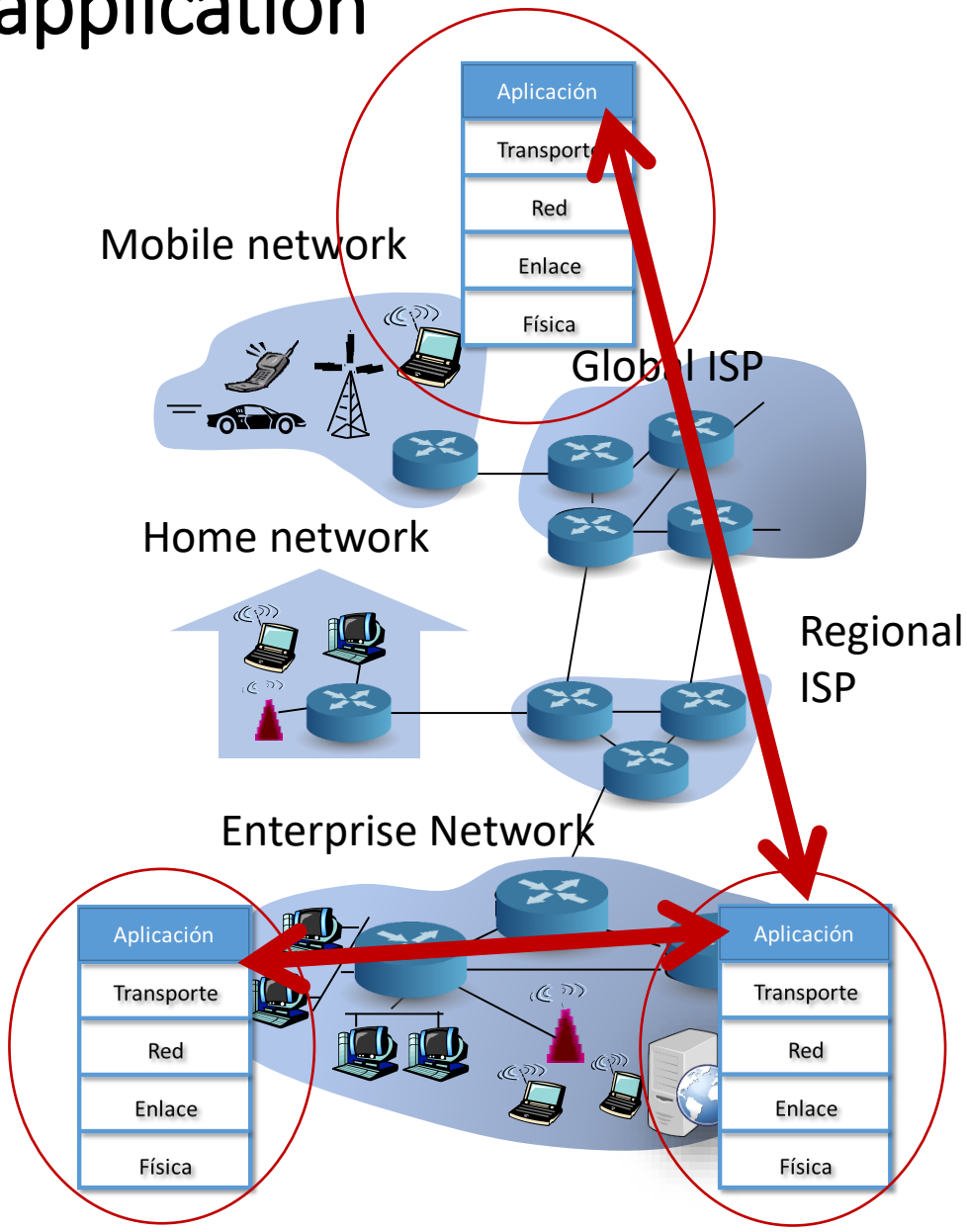
Content

1. Principles of networked applications
2. DNS
3. Web and HTTP

Some networked applications...

Creating a networked application

- Develop programs that
  - Run on (different) end systems
  - Communicate over the network
  - Ex: software of a web server that communicates with web browser software
- No programs are made for the core of the network
  - Core devices do not run user applications
  - Doing so on end systems accelerates application development and propagation time



Lesson 2: The Application Layer

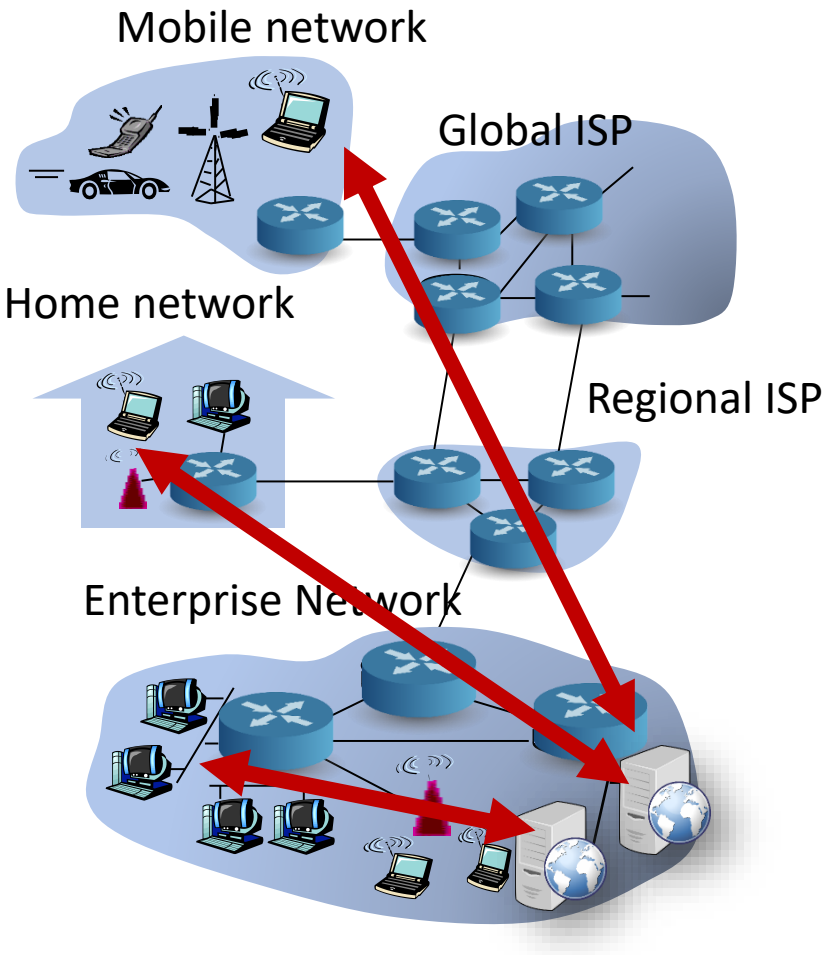
Objectives

- To understand the application layer of the TCP/IP model and the OSI model
- To know some basic protocols of this layer

Content

1. **Principles of networked applications**
2. DNS
3. Web and HTTP

Principles of networked applications Client-Server Architecture



Server:

- Always-ON Team
- Fixed IP address
- Scalability farms

Clients:

Communicate with the server

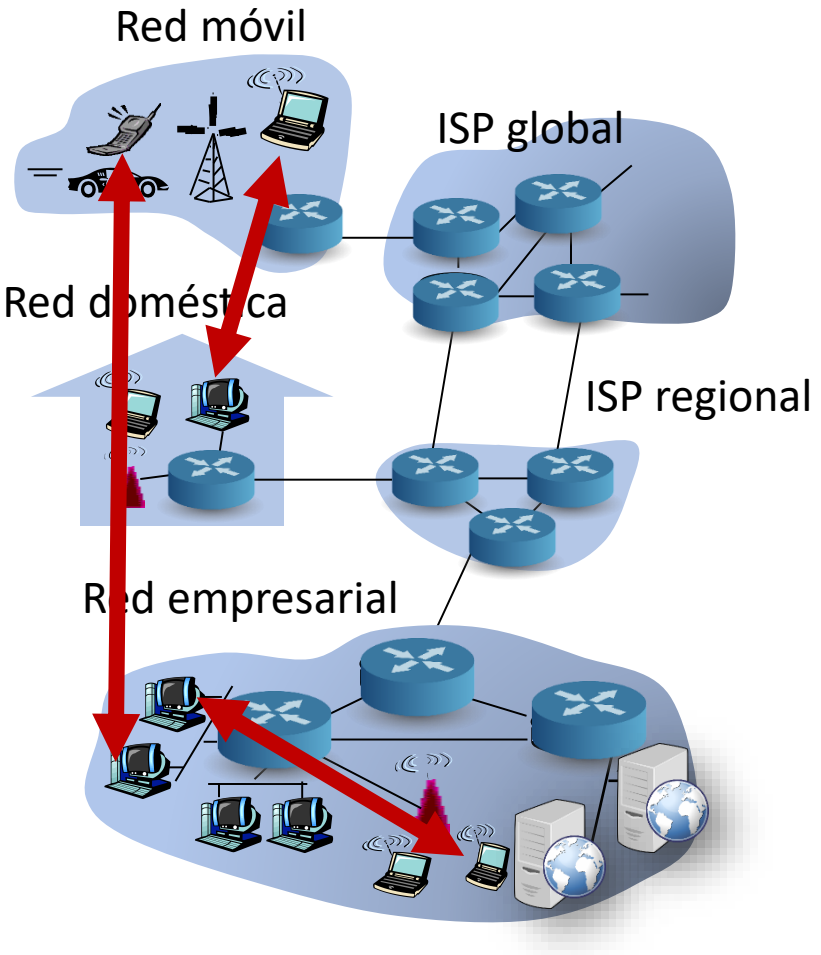
- Intermittently
- With dynamic or fixed IPs
- They do not communicate directly with each other

IP Address: Uniquely identifies computers (end systems, routers,...) connected to a TCP/IP network. It is assigned by the ISP statically (fixed) or dynamically (variable). More on lesson 4...

P2P Architecture

- The server is not always-ON.
- The final systems communicate with each other arbitrarily.
- Peers communicate intermittently and with different IP addresses each time.

Very scalable but difficult to manage.



Principles of networked applications

Examples of hybrid client-server architecture + P2P

Skype

- Voice-over-IP application P2P architecture
- Centralized server: find IP address of the remote interlocutor
- Client-client connection: direct (bypassing the server)

Instant messaging

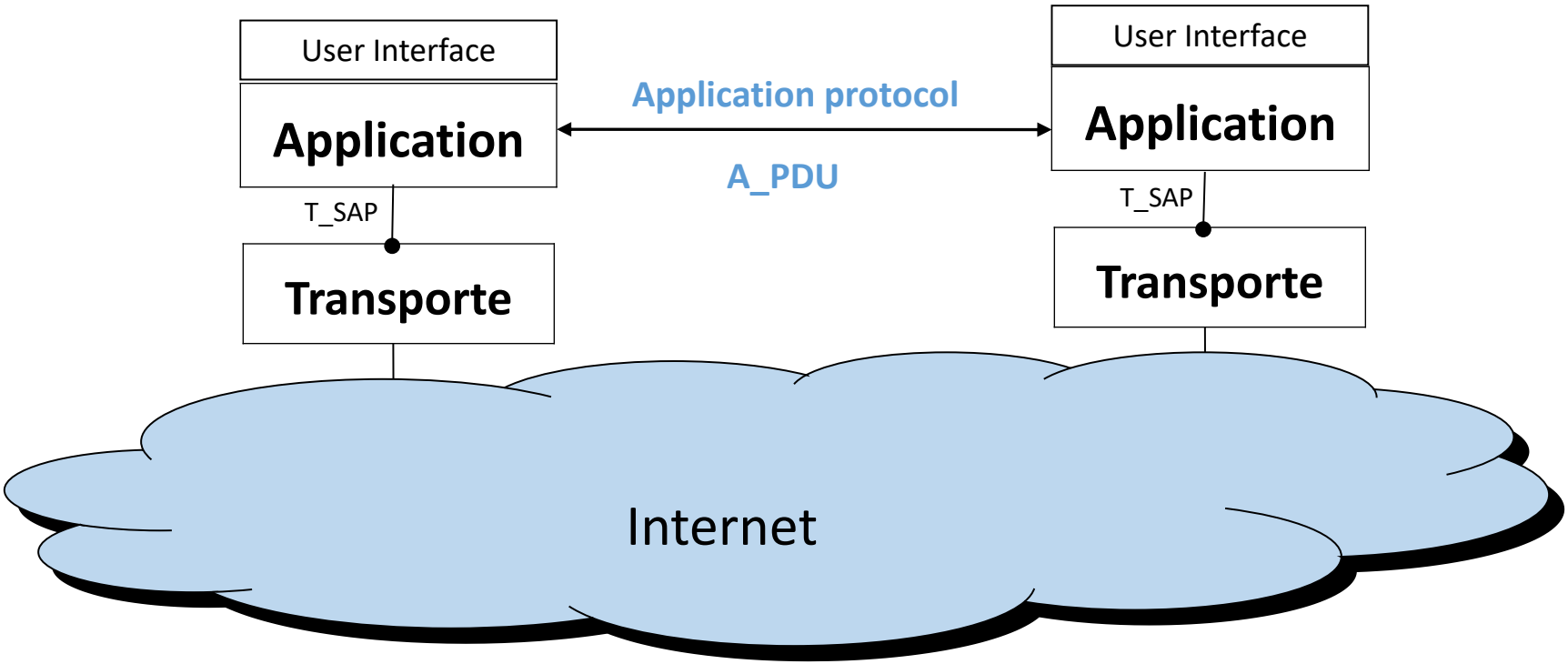
La charla entre dos usuarios es P2P

- Centralized server: detects presence and location of clients
- Users register their IP with the central server when connecting
  - Users dialogue with the central server in search of their contact's IP

IP Address: Uniquely identifies computers (end systems, routers,...) connected to a TCP/IP network. It is assigned by the ISP statically (fixed) or dynamically (variable). More on lesson 4...

Principles of networked applications

How is the application layer implemented?



Web browsers, e.g. Chrome, Opera, Microsoft Edge, Safari, Tor...

Principles of networked applications

Application-level protocol defines…

- Type of message to exchange,
  - E.g. request or response
- Message syntax
  - Number of fields and delimitation between them
- Message semantics
  - Meaning of fields
- Rules for how and when processes send and respond to messages

Public domain protocols:

- Defined in RFCs
- Allow interoperability
- E.g.: HTTP, SMTP

Protocolos propietarios:

- E.g.: Skype, Whatsapp

Principles of networked applications

Communication between processes

Process: program that runs on a computer (in our case implements a certain application protocol).

- In the same computer, two processes communicate using inter-process communication (provided by the OS).
- Processes on different computers communicate by exchanging messages (PDU) using communication services (generally provided by OS)

Client process: process that initiates communication

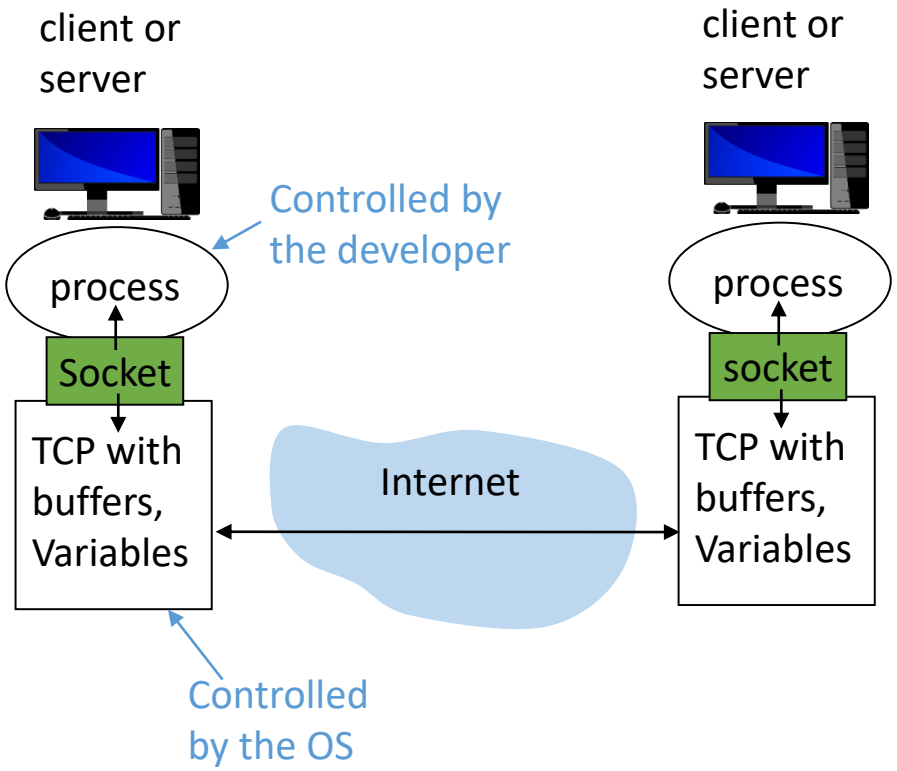
Server process: process waiting to be contacted

Note: P2P applications combine both client and server processes

Principles of networked applications
Sockets (SAP)

- A process sends/receives messages to/from its socket
- Analogy with a door:
- The sending process sends the message through the exit door
- The sending process relies on the transport infrastructure behind the door, responsible for carrying the message to the receiver's door
-

- API: (1) selection of transport service ; (2) posibility to set parameters



Principles of networked applications

How is the socket identified?

- To send a letter to someone is necessary to know their address so that it reaches the mailbox of the house.
  - Each end system has a unique 32-bit ip address.

> **Note**
>
> IP addresses are associated with a name, which is used to identify computers.
>
> For example, www.dte.us.es = 150.214.141.196
>
> More about names in the next section...

Is the address enough to get the letter to a friend? No, multiple people may be living in the same house.

Multiple application protocols may be running on a final system.

- Browser, mail reader, Skype, …

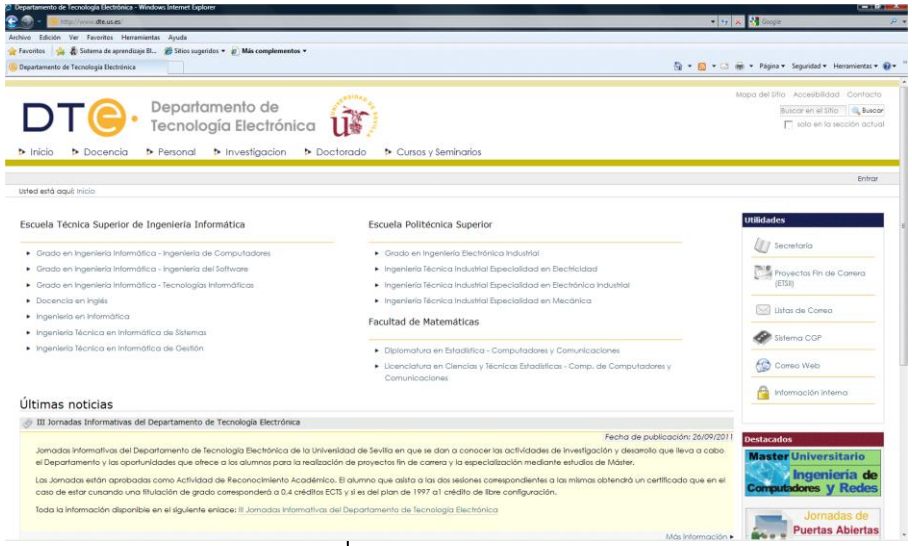
Principles of networked applications
How is the socket identified?

- Each application protocol is identified by a port number.
- the port number used to identify the client process and the server in general do not match.
- E.g. port number:
  - HTTP Server: 80
  - HTTPS Server: 443
  - Email Server: 25
  - DNS Server: 53

< 1024 are well-known ports.
>= 1024 are dynamic or private ports.

- ICANN (Internet Corporation for Assigned Names and Numbers), is responsible for the registration of public application protocol ports (http://www.iana.org/assignments/port-numbers)
- There are different types of ports.
- A socket is identified by:
  - IP address.
  - Port number.

Principles of networked applications
Example



DTE Web Server

150.214.141.196, 80

Client IP, port

Transport

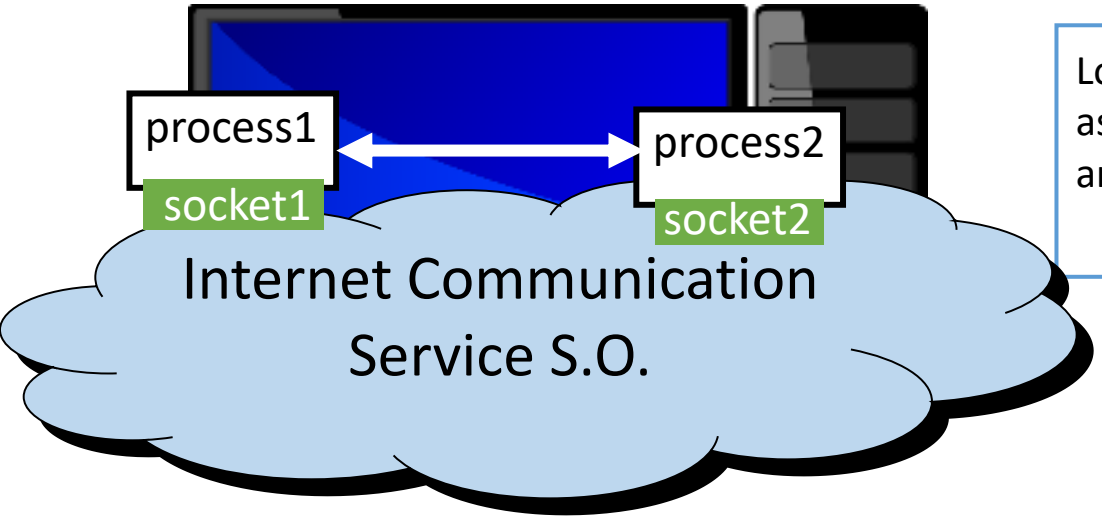
Transport

Internet

Principles of networked applications

Localhost: Connecting 2 processes from the same final system

- localhost: is a "special name" that is associated with a special IP address that serves to identify the final system itself.
- Allows you to test networked applications on a single final system without having to be connected to a network.
- In general, it allows to communicate processes in the same final system using Internet communications services.



Nota

Localhost usually has the IP 127.0.0.1 associated with it, although it can be another. More on lesson 4...

P: Why is the Communication Service of the final system able to distinguish each process?

Principles of networked applications

What transportation services do I need?

Data loss

Some applications tolerate some loss (e.g. audio, video). Others require 100% reliability (e.g. login, file transfer)

Timing

Some applications require short delays to be 'effective' (e.g. Internet telephony, interactive games)

Transfer rate

Some require a minimum rate to function properly (e.g. multimedia). Others, known as "elastic applications", make use of the rate available at all times.

Safety

Encryption, data integrity, ...

Principles of networked applications

Requirements for some common applications

Application	Data loss	Transfer rate	Time sensitive
File transfer/download	No losses	Elastic	No
Email	No losses	Elastic	No
Web documents	No losses	Elastic (few kbps)	No
Internet telephony/ Videoconference	Loss tolerant	Audio: a few kbps-1 Mbps Video: 10Kbps-5Mbps	Yes; tenths of a sec
Stored audio/video streams	Loss tolerant	Audio: a few kbps-1 Mbps Video: 10Kbps-5Mbps	Yes; a few sec
Interactive games	Loss tolerant	A few Kbps – 10Kbps	Yes; tenths of a sec
Smartphone messaging	No losses	Elastic	Yes and no

Principles of networked applications
Internet Protocol Services

TCP Service

- **Connection-oriented:** requires prior agreement between client and server processes before initiating the transfer
- **Reliable transport** between sending and receiving processes
- **Flow control:** emitter will not saturate the receiver
- **Congestion Control:** Equitable Use of Bandwidth
- **Does not provide:** timing, ensure bandwidth, security

UDP Service

- Lightweight, non-connection-oriented, unreliable transport between sender and receiver processes
- **Does not provide:** prior agreement between processes, reliability, flow control, congestion control, timing, guaranteed bandwidth, or security.

Q: What is the usefulness of UDP?

Principles of networked applications

Examples: Application and Transport Protocols

Application	Application Layer Protocol	Underlying transport protocol
Email	SMTP [RFC 5321]	TCP
Remote terminal access	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2116]	TCP
File transfer	FTP [RFC 959]	TCP
Flujos multimedia	HTTP (e.g. YouTube)	TCP
Internet telephony	SIP [RFC 3261], STP[RFC 3550] or proprietary (e.g. Skype)	UDP or TCP

Lesson 2: The Application Layer

Objectives

- To understand the application layer of the TCP/IP model and the OSI model
- To know some basic protocols of this layer

Content

1. Principles of networked applications
2. **DNS**
3. Web and HTTP

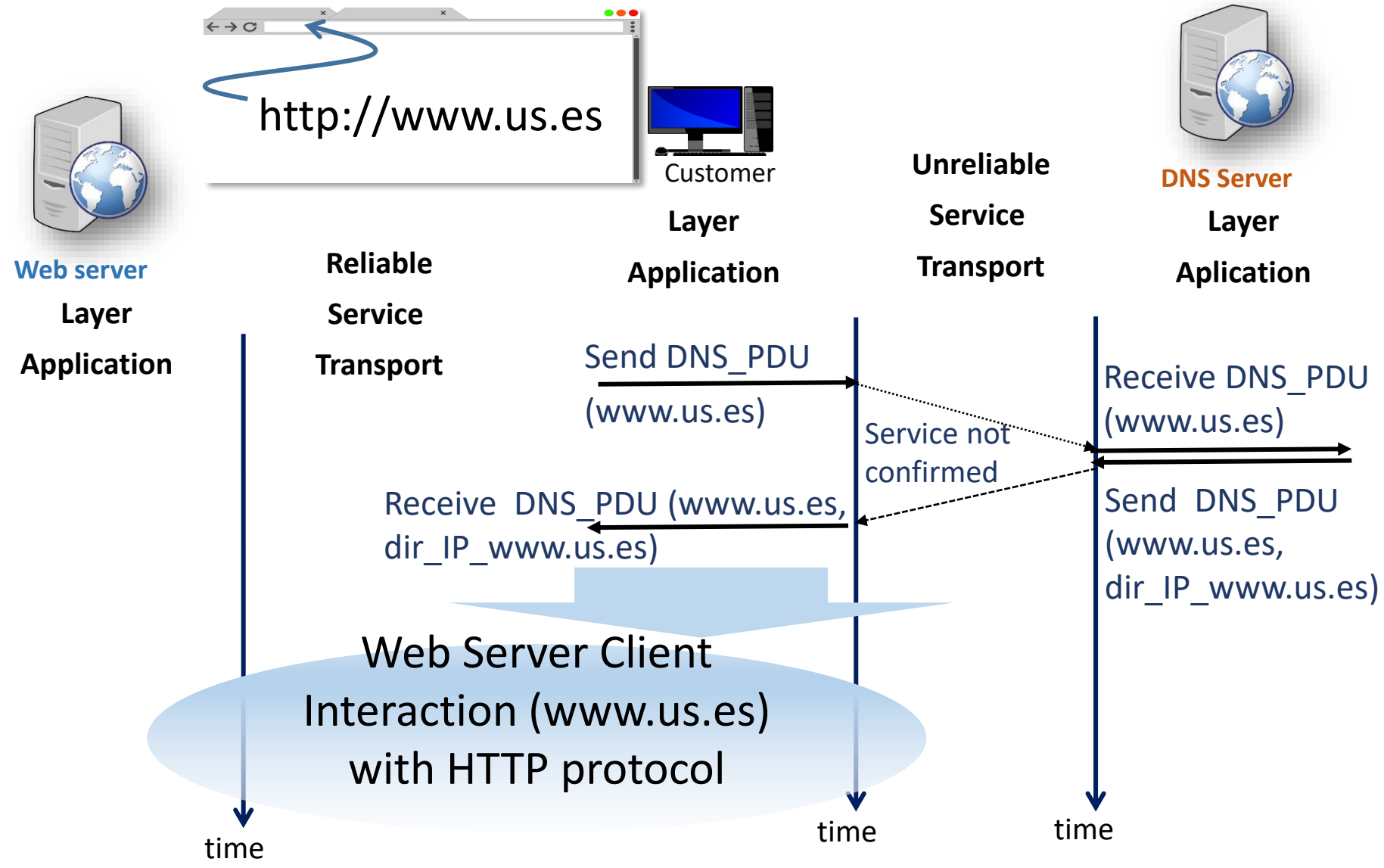
DNS: Domain Name System

- People have many IDs: DNI, name, social security number...
- Internet equipment and routers:
  - IP addresses (32 bit) – used to address datagrams
  - "name", e.g. www.google.com – used by humans
  - Q: How do we map between IP addresses and names and vice versa?

Domain Name System:

- **Distributed database** deployed with a hierarchy of name servers
- **Application-level protocol**: computers and name servers communicate to resolve names (address and name translation)
- Fundamental feature of the Internet, implemented at the application level!

DNS: Simplified operation



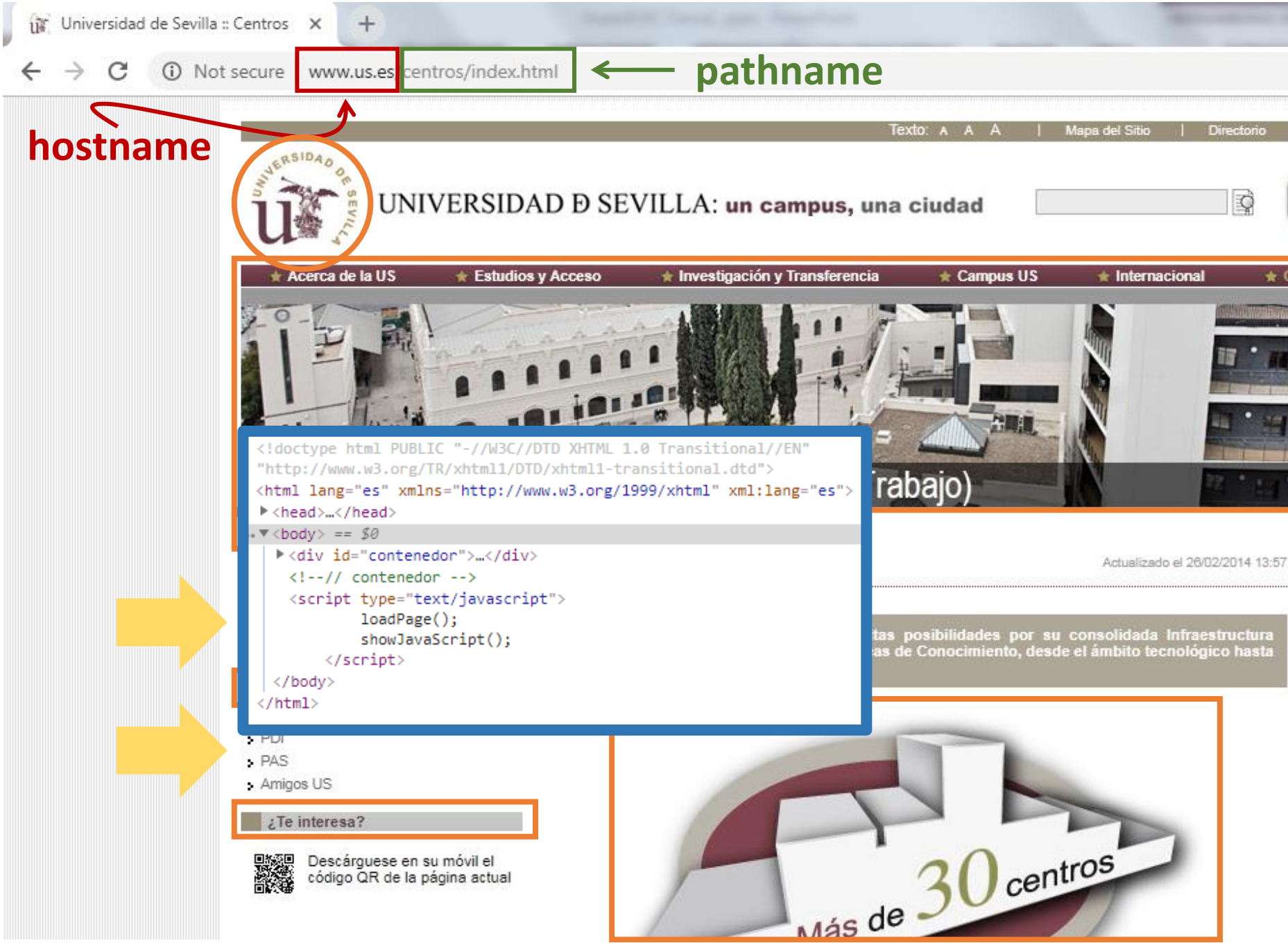
Lesson 2: The Application Layer

Objectives

- To understand the application layer of the TCP/IP model and the OSI model
- To know some basic protocols of this layer

Content

1. **Principles of networked applications**
2. **DNS**
3. **Web and HTTP**


Universidad de Sevilla :: Centros
Not secure
www.us.es
centros/index.html
pathname
hostname
Texto: A A A | Mapa del Sitio | Directorio
UNIVERSIDAD Ð SEVILLA: un campus, una ciudad
Acerca de la US
Estudios y Acceso
Investigación y Transferencia
Campus US
Internacional
<!doctype html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html lang="es" xmlns="http://www.w3.org/1999/xhtml" xml:lang="es">
<head>…</head>
<body> == $0
…
<!--// contenedor -->
<script type="text/javascript">
loadPage();
showJavaScript();
</script>
</body>
</html>
Actualizado el 26/02/2014 13:57
PDI
PAS
Amigos US
¿Te interesa?
Descárguese en su móvil el código QR de la página actual
Más de 30 centros

Web and HTTP

HTML Markup Language Format

It is used to elaborate the web pages, since 1991 (current version: HTML5). Elements with tags between <> . Each element usually has 4 fields: start (<html>) and a close(</html>), some attributes (in the start) and a content (between them). It is interpreted by the client.

```
<!DOCTYPE html…>          defines document start (optional)
<html>página</html>       defines start/end document
<head>cabecera</head>     the content of the header (information
"not visible" to the user, such as title, styles, meta-information,
etc...)
<body>cuerpo</body>       defines the body, contains:
<h1> a <h6>               Headlines
<table>tabla</table>      creates a row/column table
<a href="URL">enlace</a>  defines a hyperlink; clicking on "link"
prompts for the URL page.
<img src="URL"/>          referenced image, the browser loads it
from URL to view it.
```

Note

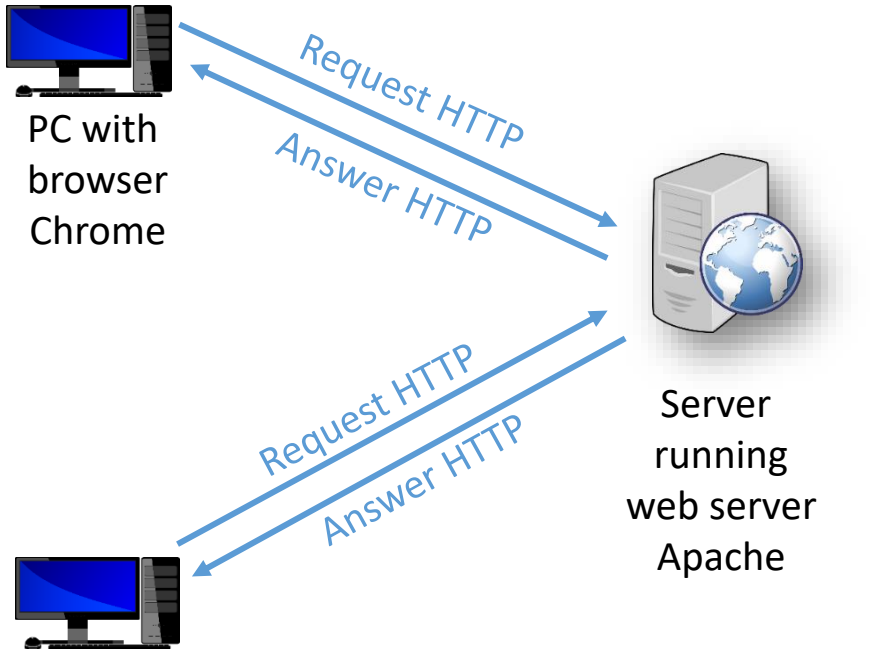
You can see the HTML code of a page in the browser, right-click → view source code

Web and HTTP

HTTP at a glance

HTTP: HyperText Transfer Protocol

- Application-level protocol for the web.
- Client/server model
  - **Client**: browser that requests, receives, and displays web objects.
  - **Server**: process that sends objects ordered by customers.
- Use Reliable Connection Oriented Transport (TCP).
- It is "stateless"”
  - the server does not save information about previous client requests
- Types of HTTP connections:
  - Non-Persistent
  - Persistent



Request HTTP:

- Sent by the client
- Transports information needed (HTTP_PCI) to request an object from the server (HTTP_UD)
- Consists of ASCII characters (intelligible text)

Response HTTP:

- Sent by the server
- Transports if appropriate the object (HTTP_UD) requested by the client in addition to control information (HTTP_PCI)

Web and HTTP

Type of HTTP connections

Non-persistent HTTP

At most one object is sent for each TCP connection.

Persistent HTTP

Multiple objects can be sent over the same TCP connection between client and server.

Web and HTTP

HTTP non-persistent

Suppose a user enters this URL:

http://www.us.es/centros/index.html

(contains text and references to 6 objects)

1a. The HTTP client application requests to establish a TCP connection to the server process on the computer www.us.es to the port 80

1b. The HTTP server application on the www.us.es computer which was waiting for TCP connections on port 80 accepts this connection and notifies the client.

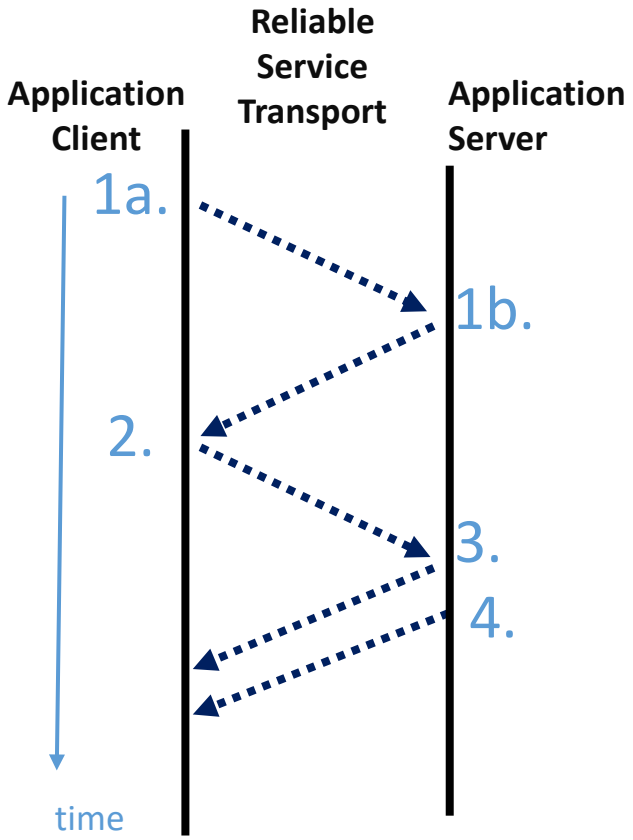
2. The HTTP client sends a request message (containing the URL) on the established TCP connection. the message indicates that the client wants the /centros/index.html object

3. The HTTP server receives the request forms a response message containing the requested object and sends it through its socket.

4. The HTTP server requests closure of the TCP connection (so has the client).

5. The HTTP client receives the response message, containing the HTML file, displays the content and analyzes it finding 6 references to other objects.

6. Steps 1-5 are repeated for each of the 6 objects (4 images and 2 JavaScript scripts) with different URLs.



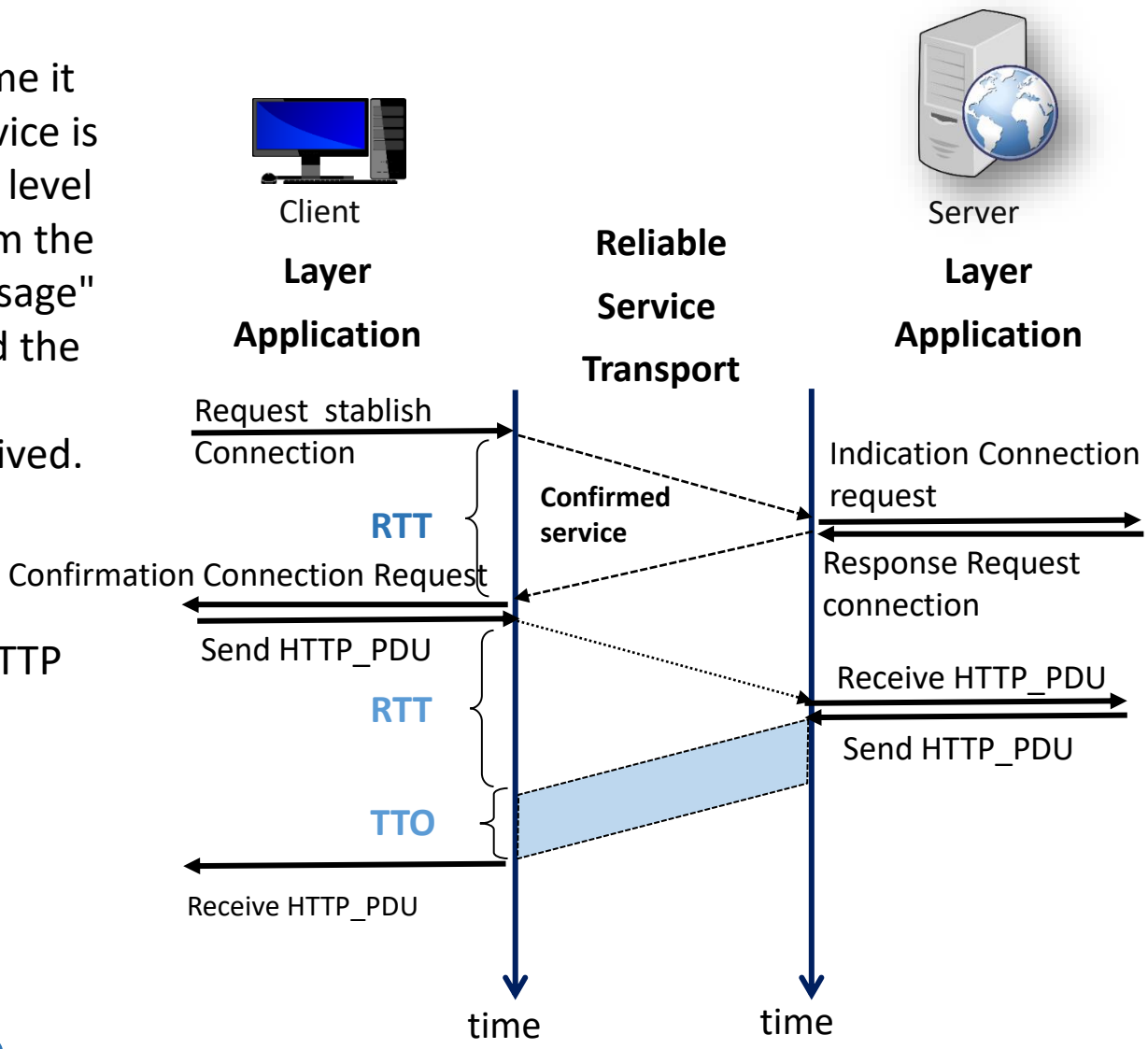
Web and HTTP

HTTP Non-persistent: Response time

RTT (*Round-Trip Time*): the time it takes from the when a service is requested at the transport level until it is completed or from the moment the "request message" is requested to be sent and the first bytes of the "reply message" begin to be received.

Response time (RT):

- 1 RTT, start connection.
- 1 RTT, HTTP request and HTTP response first bytes.
- Transmission time - bytes requested object (TTO).



TR= 2RTT + TTO

Web and HTTP

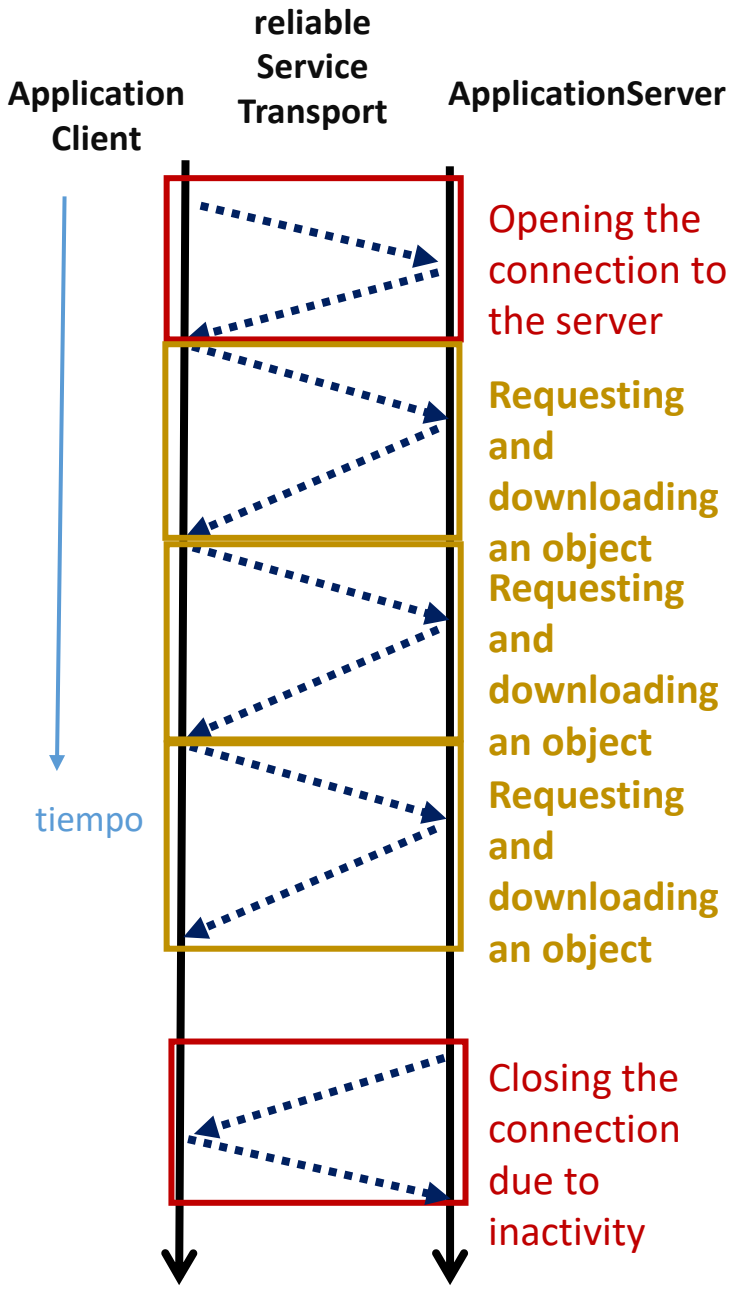
HTTP Persistent

- El servidor mantiene la conexión abierta tras enviar la respuesta.
- Los siguientes mensajes HTTP entre el mismo cliente y el servidor se envían por la conexión abierta.
- El cliente envía una nueva petición cuando acaba de recibir el objeto anterior.
- Cada objeto referenciado tarda sólo 1 RTT (más lo que tarde en transmitirse dicho objeto).

Note

Parallel HTTP connections:

Browsers often open several TCP connections in parallel to get the referenced objects more quickly, which are requested simultaneously for each connection (whether persistent or not).



Web and HTTP
Messages HTTP (or HTTP_PDU)

There are 2 types of messages:

HTTP Request:

- Sent by the client
- Transports information needed (HTTP_PCI) to request an object from the server (HTTP_UD)
- Consists of ASCII characters (intelligible text)

- **HTTP Response**:
  - Sent by the server
  - Transports if appropriate the object (HTTP_UD) requested by the client in addition to control information (HTTP_PCI)

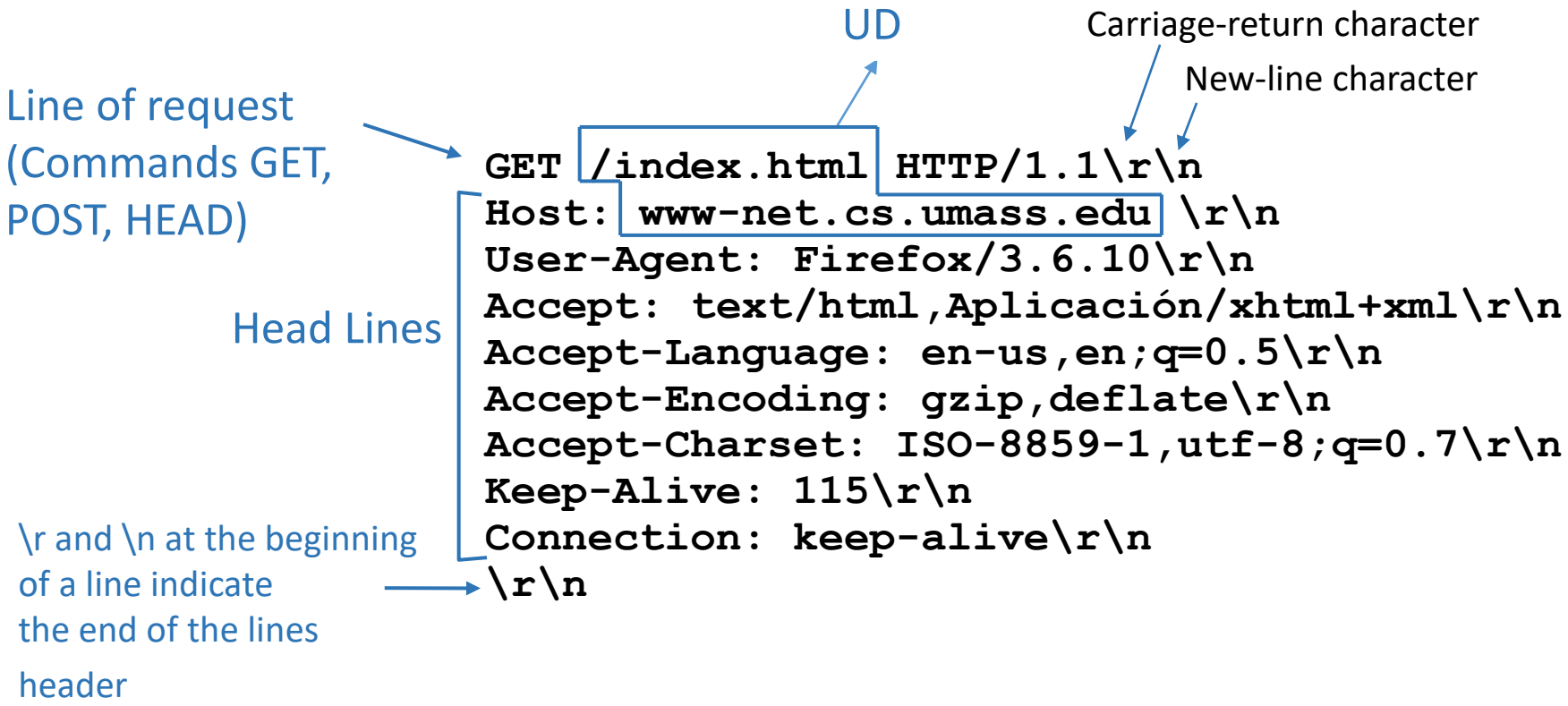
Web and HTTP

HTTP Request Message

Note

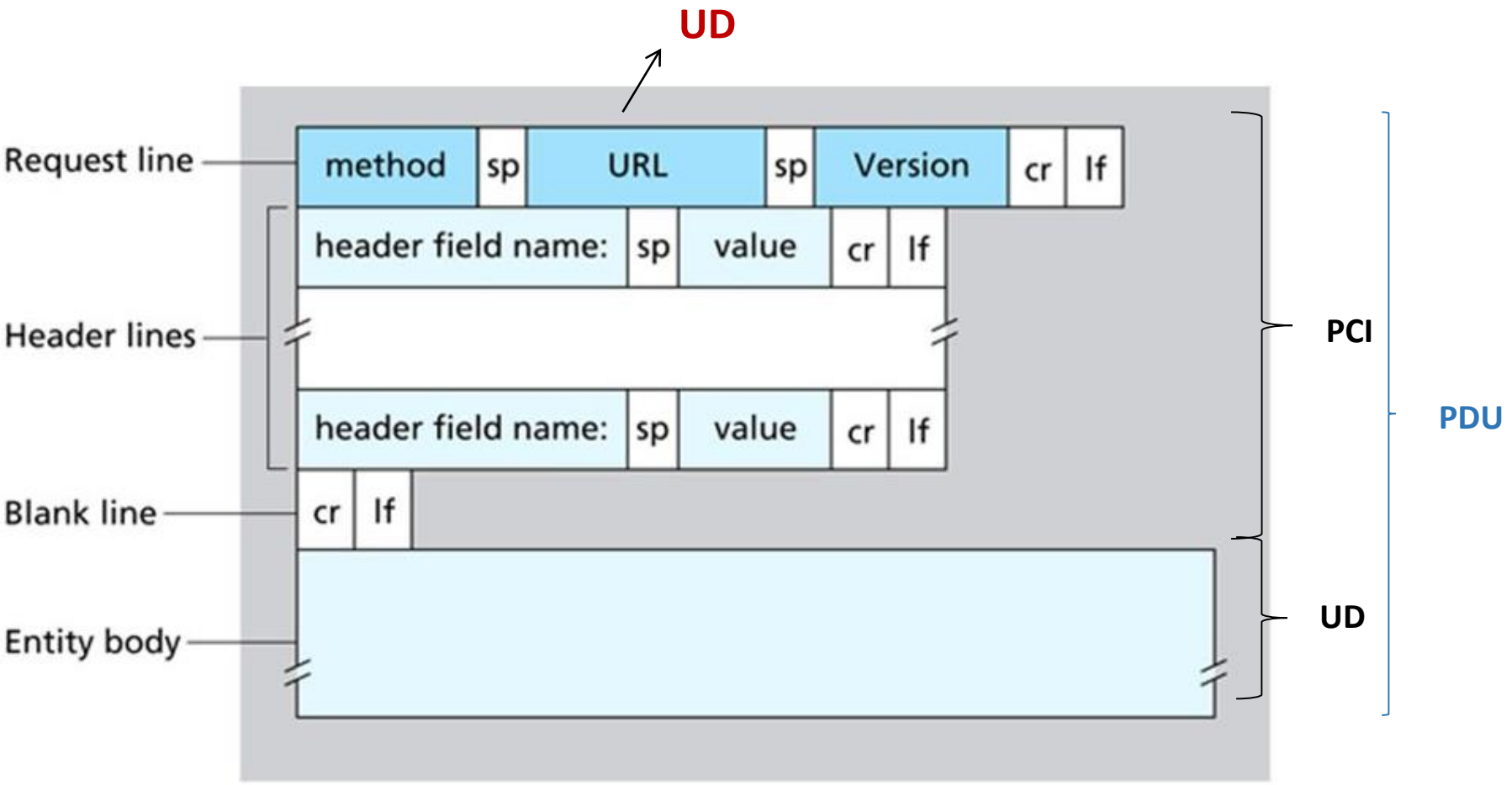
<CR>: Carriage-Return : \r

<LF>: Line-Feed: \n



Web and HTTP

HTTP Request Message: general format



Method (HTTP 1.1.): GET, POST, HEAD, PUT, DELETE

Web and HTTP

HTTP Response Message

status line (**protocol**, **status code**, **phrase status**)

Header lines

```
HTTP/1.1 200 OK\r\n
Date: Sun, 26 Sep 2010 20:09:20 GMT\r\n
Server: Apache/2.0.52 (CentOS)\r\n
Last-Modified: Tue, 30 Oct 2007 17:00:02 GMT\r\n
ETag: "17dc6-a5c-bf716880"\r\n
Accept-Ranges: bytes\r\n
Content-Length: 2652\r\n
Keep-Alive: timeout=10, max=100\r\n
Connection: Keep-Alive\r\n
Content-Type: text/html; charset=ISO-8859-1\r\n
\r\n
data data data data data data data data data data
data data data data data data data data data data
data data data data data data data data data data
data data data data data ...
```

PCI

UD

BODY
e.g. file
HTML requested

Common status codes: 200 OK; 301 Moved Permanently; 400 Bad Request; 404 Not Found; 505 HTTP Version Not Supported

Web and HTTP

Cookies: maintaining "the state"

Many web services use cookies. They have four components:

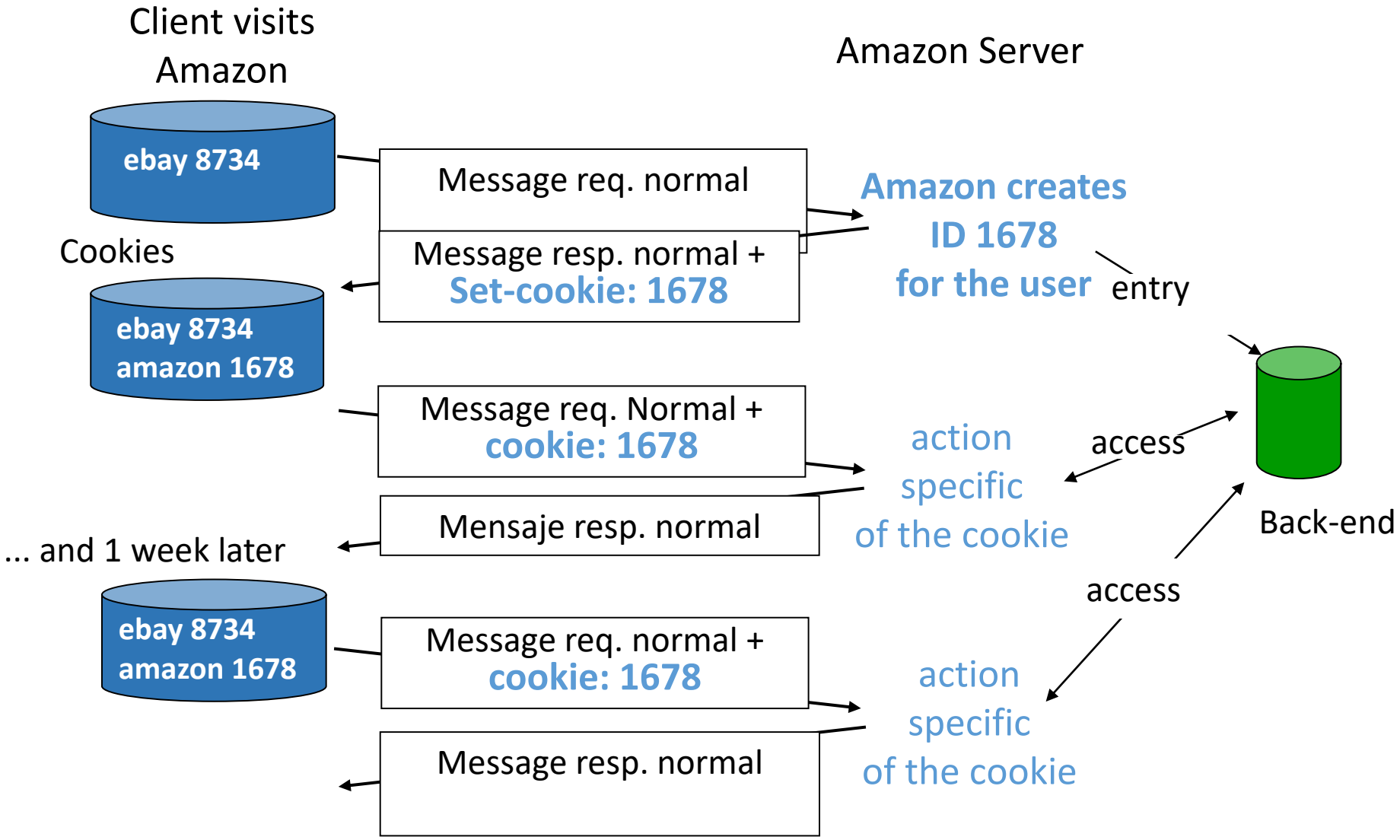
1) Set-cookie header: in the reply message
2) Cookie header: in the request message
3) cookie file stored by the user's computer and managed by the browser
4) back-end database on the web server

Example of use:

- Pedro always accesses the Internet from his PC
- Visit an online store (e.g. Amazon) for the first time
- When the requests arrive, the server creates:
  - A unique ID
  - An entry for that ID in the back-end database

Web and HTTP

Cookies: Maintaining the state. Example



Web and HTTP
Cookies: discussion

Possible applications:

- authorization
- shopping carts
- Recommendations
- user session maintenance (ex: webmail)

Cookies and privacy:

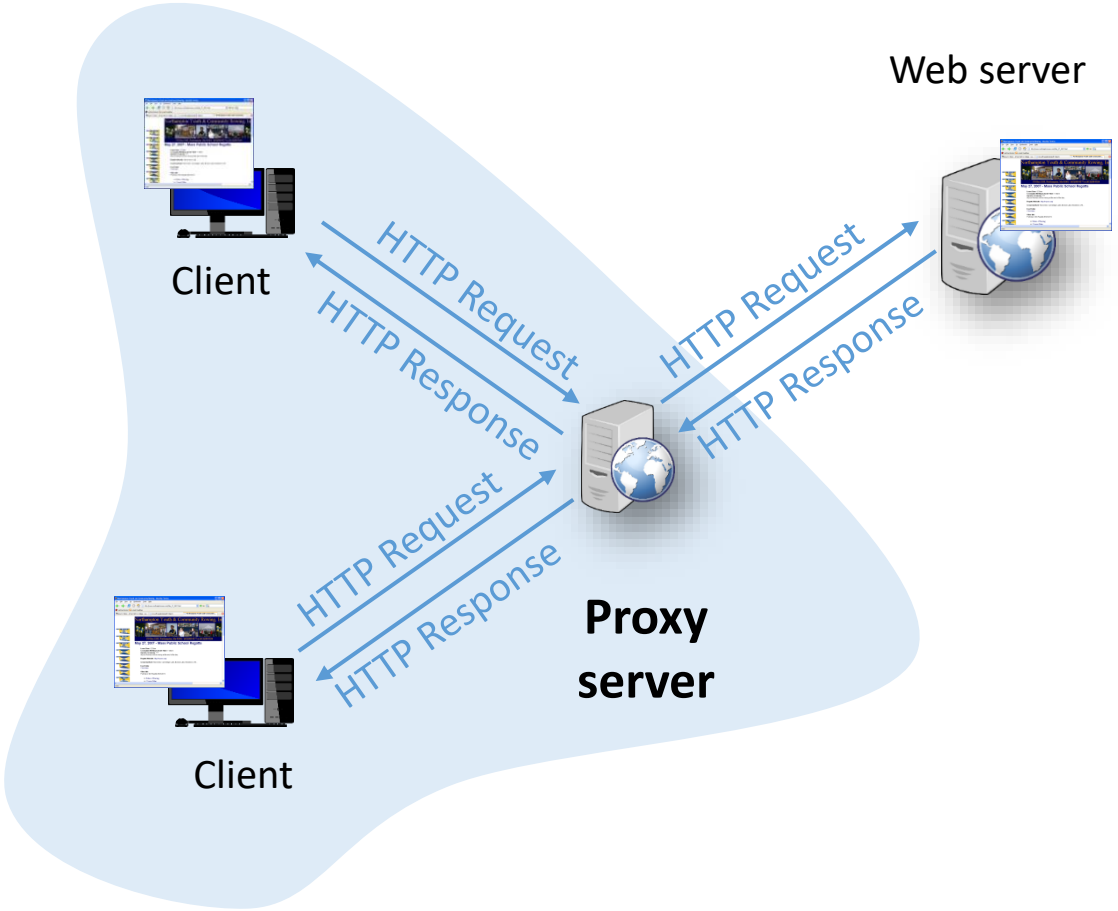
- cookies allow sites to know a lot about you
- you may be giving personal information to those pages: emails, names, etc ...

Web and HTTP

Servidor Proxy (Caché de la Web)

Objective: to satisfy the client's request without involving the original web server.

- The browser is configured to use the Proxy-Cache.
- All HTTP requests are then sent to the Proxy
  - If object in cache → object is returned
  - otherwise → cache requests the object from the original server and returns it to the client.



Web and HTTP
More about Proxy

- The cache acts as a client (from the original server) and as a server (from the client)
- They are usually installed in ISPs (universities, companies, residential ISPs)

Why is it interesting?

- Reduces client request response time
- Reduces an institution's data link traffic
- It allows "small" providers to efficiently deliver content (something that also allows P2P)

Web and HTTP
Conditional GET

- Proxy (or browser cache): Specifies the date of the cached copy in the HTTP request
- **If-modified-since: <date>**
- Servidor: in the answer there are headers and...
- a) no object is included if the copy has not been modified...

HTTP/1.0 304 Not Modified

b) or the object is sent if it is modified, along with the modification date:

Last-modified:<date>

Objective

The Web server does not send the object if the cache has an updated version of it