Computer Networks - *Xarxes de Computadors*

**Outline**
- Course Syllabus
- Unit 1: Introduction
- Unit 2. IP Networks
- Unit 3. TCP
- Unit 4. LANs
- **Unit 5. Network applications**
Unit 5. Network applications

Outline

- DNS
- Email
- Web
- Charsets
- HTML
Unit 2: IP Networks

Domain Name System DNS (RFC 1034, 1035)

- Allows users to use names instead of IP addresses: e.g. rogent.ac.upc.edu instead of 147.83.31.7, www.upc.edu instead of 147.83.194.21, etc.
- Names consists of a node-name and a domain-name: rogent.ac.upc.edu, www.upc.edu
- DNS consists of a worldwide distributed data base.
- DNS data base entries are referred to as Resource Records (RR).
- The information associated with a name is composed of 1 or more RRs.
- Names are case insensitive (e.g. www.upc.edu and WWW.UPC.EDU are equivalent).
Unit 2: IP Networks

DNS – Domain Hierarchy

- DNS data base is organized in a tree:

```
unnamed root

Top Level Domains (TLD)
  - edu
  - com
  - net
  - ... (other TLDs)

Second Level Domains
  - upc
  - ac
  - ... (other second level domains)

node-name
  - rogent

Generic Domains
  - Allow reverse resolution

Country Domains

Infrastructure Domains
```
DNS – Domain Hierarchy

- The Internet Corporation for Assigned Names and Numbers (ICANN) is responsible for managing and coordinating the DNS.
- ICANN delegates Top Level Domains (TLD) administration to registrars: http://www.internic.net
- Domains delegate the administration of subdomains.

InterNIC—Public Information Regarding Internet Domain Name Registration Services

Do you have a complaint or dispute?

Your Registrar or Domain Name:

- Domain Name Transfer Dispute
- Unsolicited Renewal or Transfer Solicitation
- Your Registrar is Not on the Accredited List
- Unauthorized Transfer of Your Domain Name
- Trademark Infringement
- Registrar Services Dispute
  - Failure to answer phones or respond to email messages
  - Financial Transaction Issues
- Uniform Domain Name Dispute Resolution (UDRP) Intake Report System

Information about Registrars

- Search Accredited Registrar Directory
  - Alphabetical List
  - List by Location
  - List by Language Supported
- Have a Problem with a Registrar?
  - Complaint Form
  - Helpful Hints

Information about Whois

- Search Whois
- Report Inaccurate Whois Listing
Unit 2: IP Networks

DNS – Data Base Organization

- Access to DNS data base is done using *Name Servers (NS)*.
- NSs may hold permanent and *cached RRs*. Cached RRs are removed after a timeout.
- Each subdomain has an *authority* which consists of a primary and backup NSs.
- In this context, subdomains are referred to as *zones*, and delegated subdomains *subzones*.
- An authority has the complete *information of a zone*:
  - Names and addresses of all nodes within the zone.
  - Names and addresses of all subzone authorities.
Unit 2: IP Networks

DNS – Data Base Organization

- **Root Servers** are the entry point to the domain hierarchy.
- Root Servers are distributed around the world and have the TLD addresses: [http://www.root-servers.org](http://www.root-servers.org)
- Root server addresses are needed in a NS configuration.

Source: [http://www.root-servers.org](http://www.root-servers.org)
Unit 2: IP Networks

DNS - Unix example: The resolver

- The applications use the calls (*resolver* library):

  ```c
  struct hostent *gethostbyname(const char *name) ;
  struct hostent *gethostbyaddr(const void *addr, int len, int type);
  ```

- The resolver first looks the `/etc/hosts` file:

  ```
  # hosts         This file describes a number of hostname-to-address
  # mappings for the TCP/IP subsystem. It is mostly
  # used at boot time, when no name servers are running.
  # On small systems, this file can be used instead of a
  # "named" name server.
  # Syntax:
  # IP-Address  Full-Qualified-Hostname  Short-Hostname
  127.0.0.1       localhost
  10.0.1.1        massanella.ac.upc.edu massanella
  ```

- Otherwise a *name server* is contacted using `/etc/resolv.conf` file:

  ```
  search ac.upc.edu
  nameserver 147.83.32.3
  nameserver 147.83.33.4
  ```
Unit 2: IP Networks

DNS - Protocol

- Client-server paradigm
- UDP/TCP. Short messages uses UDP.
- well-known port: 53

http://www.foo.org

18:36:00.322370 IP (proto: UDP) 147.83.34.125.1333 > 147.83.32.3.53: 53040+ A? www.foo.org. (31)

18:36:00.323080 IP (proto: UDP) 147.83.32.3.53 > 147.83.34.125.1333: 53040 1/2/2 www.foo.org. A 198.133.219.10 (115)
Unit 2: IP Networks

DNS – Unix example: Basic NS configuration

- Unix NS implementation is **BIND** (Berkeley Internet Name Domain), [http://www.isc.org](http://www.isc.org).
- **named** is the BIND NS daemon.
- **BIND basic configuration files:**
  - `/etc/named.conf` global configuration
  - `/var/lib/named/root.hint` root servers addresses
  - `/var/lib/named/*.db` zone files
Unit 2: IP Networks

DNS – Unix example: zone file

The domain name
- The domain NS
- The domain maintainer mail address (the @ is written as a '.')

Resource Records (RR)
- MX preference value (used if multiple servers are available)
- IN: Internet System.
- name (type A or CNAME), domain (type NS of MX).
- If the domain is missing, it is automatically added.
- class: IN: Internet System.
- type:
  - SOA: Start Of Authority.
  - NS: NS name.
  - MX: the domain mail exchange.
  - A: A host address.
  - CNAME: Canonical Name Record. E.g. the real hostname of www.foo.org is server.foo.org.

Comments
- Linux # cat /var/lib/named/foo.db
  ; BIND data file for foo.org
  ; /var/lib/named/foo.db

Configuration
- foo.org. IN SOA dns.foo.org. root.foo.org. (1998121401 ; Serial
  604800 ; Refresh
  86400 ; Retry
  2419200 ; Expire
  604800 ) ; Default TTL
- IN NS dns.foo.org.
- IN MX 10 mail.foo.org.
- server IN A 198.133.219.10
- www IN CNAME server
- ftp IN CNAME server
- news IN A 198.133.219.20
- mail IN A 198.133.219.30
- dns IN A 198.133.219.40
- dns2 IN A 198.133.219.50
- ... sub.foo.org. IN NS dns3.sub.foo.org.
- dns3 IN A 10.10.0.24
- ... IP addresses and alias names

 NS name domain mail server
- IN NS dns.foo.org.
- IN MX 10 mail.foo.org.
- server IN A 198.133.219.10
- www IN CNAME server
- ftp IN CNAME server
- news IN A 198.133.219.20
- mail IN A 198.133.219.30
- dns IN A 198.133.219.40
- dns2 IN A 198.133.219.50
- ... sub.foo.org. IN NS dns3.sub.foo.org.
- dns3 IN A 10.10.0.24
- ... IP addresses and alias names

The domain mail server
- IN NS dns.foo.org.
- IN MX 10 mail.foo.org.
- server IN A 198.133.219.10
- www IN CNAME server
- ftp IN CNAME server
- news IN A 198.133.219.20
- mail IN A 198.133.219.30
- dns IN A 198.133.219.40
- dns2 IN A 198.133.219.50
- ... sub.foo.org. IN NS dns3.sub.foo.org.
- dns3 IN A 10.10.0.24
- ... IP addresses and alias names

Address (type A), name (type NS or CNAME)...

Delegated sub-domain

Class: IN: Internet System.
DNS – Unix example: root servers addresses

```
linux # cat /var/lib/named/root.hint

; This file holds the information on root name servers needed to
; initialize cache of Internet domain name servers
; (e.g. reference this file in the "cache . <file>"
; configuration file of BIND domain name servers).
;
; This file is made available by InterNIC
; under anonymous FTP as
; file /domain/named.root
; on server FTP.INTERNIC.NET
; -OR- RS.INTERNIC.NET

. 3600000 IN NS A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET. 3600000 IN A 198.41.0.4
.
B.ROOT-SERVERS.NET. 3600000 IN A 192.228.79.201
.
C.ROOT-SERVERS.NET. 3600000 IN A 192.33.4.12
.
M.ROOT-SERVERS.NET. 3600000 IN A 202.12.27.33
```

Resource Records (RR) pointing to root-servers

address of a name
NS name
Unit 2: IP Networks

DNS – Resolution

- NSs **cache** name resolutions.
- A cached RR is returned without looking for in the NS authority.
- The same name may be associated with **several IP addresses** (e.g. load balancing).
- The addresses of a common domain may not belong to the same IP network (e.g. **Content Distribution Networks**).
Unit 2: IP Networks

DNS – Load balancing, example

Example using dig:

```
linux ~> dig www.microsoft.com
; <<>> DiG 9.3.2 >>> www.microsoft.com
; global options: printcmd
; Got answer:
; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 31808
; flags: qr rd ra; QUERY: 1, ANSWER: 9, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.microsoft.com.             IN      A

;; ANSWER SECTION:
toggle.www.ms.akadns.net. 181   IN      CNAME   g.www.ms.akadns.net.
g.www.ms.akadns.net.    181   IN      CNAME   lb1.www.ms.akadns.net.
lb1.www.ms.akadns.net.  181   IN      A       207.46.19.60
lb1.www.ms.akadns.net.  181   IN      A       207.46.18.30
lb1.www.ms.akadns.net.  181   IN      A       207.46.19.30
lb1.www.ms.akadns.net.  181   IN      A       207.46.198.30
lb1.www.ms.akadns.net.  181   IN      A       207.46.198.60
lb1.www.ms.akadns.net.  181   IN      A       207.46.20.60
lb1.www.ms.akadns.net.  181   IN      A       207.46.19.60
lb1.www.ms.akadns.net.  181   IN      A       207.46.198.30
lb1.www.ms.akadns.net.  181   IN      A       207.46.225.60

;; Query time: 42 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; MSG SIZE  rcvd: 203
```

```
linux ~> dig www.microsoft.com
; <<>> DiG 9.3.2 >>> www.microsoft.com
; global options: printcmd
; Got answer:
; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 17923
; flags: qr rd ra; QUERY: 1, ANSWER: 9, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.microsoft.com.             IN      A

;; ANSWER SECTION:
toggle.www.ms.akadns.net. 215   IN      CNAME   g.www.ms.akadns.net.
g.www.ms.akadns.net.    215   IN      CNAME   lb1.www.ms.akadns.net.
lb1.www.ms.akadns.net.  215   IN      A       207.46.198.30
lb1.www.ms.akadns.net.  215   IN      A       207.46.199.30
lb1.www.ms.akadns.net.  215   IN      A       207.46.18.30
lb1.www.ms.akadns.net.  215   IN      A       207.46.19.60
lb1.www.ms.akadns.net.  215   IN      A       207.46.198.60
lb1.www.ms.akadns.net.  215   IN      A       207.46.20.60
lb1.www.ms.akadns.net.  215   IN      A       207.46.19.60
lb1.www.ms.akadns.net.  215   IN      A       207.46.198.30
lb1.www.ms.akadns.net.  215   IN      A       207.46.225.60

;; Query time: 43 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; MSG SIZE  rcvd: 203
```
Unit 2: IP Networks

DNS - Content Distribution Networks, example

1. User types `www.foo.org` into their browser.
2. The browser sends a DNS query to `dns.cdn.com` for `www.foo.org`.
3. `dns.cdn.com` resolves `www.foo.org` to the IP address `80.32.40.20`.
4. The DNS response is returned to the browser.
5. The browser sends an HTTP request to `http://www.cdn.com/foo`.
6. The server at `80.32.40.20` downloads the content from the `cdn.com` servers and returns it to the user.
Unit 2: IP Networks

DNS – Messages: Message Format

- All DNS messages have the same format:
  - Header: type of message.
  - Question: What is to be resolved.
  - Answer: Answer to question.
  - Authority: Domain authority names.
  - Additional: Typically, the authority name's addresses.

```
<table>
<thead>
<tr>
<th>Header (12 bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Question (variable) /</td>
</tr>
<tr>
<td>Answer (variable) /</td>
</tr>
<tr>
<td>Authority (variable) /</td>
</tr>
<tr>
<td>Additional (variable) /</td>
</tr>
</tbody>
</table>
```
Unit 2: IP Networks

DNS – Messages: Header

- **Identification**: 16 random bits used to match query/response
- **Flags**. Some of them:
  - Query-Response, **QR**: 0 for query, 1 for response.
  - Authoritative Answer, **AA**: When set, indicates an authoritative answer.
  - Recursion Desired, **RD**: When set, indicates that recursion is desired.
- The other fields indicate the **number** of Questions, Answer, Authority and Additional fields of the message.
Unit 2: IP Networks

DNS – Messages: Question

- **QName**: Indicates the name to be resolved.
- **QType**: Indicates the question type:
  - Address, **A**.
  - Name Server, **NS**.
  - Pointer, **PTR**: For an inverse resolution.
  - Mail Exchange, **MX**: Domain Mail Server address.
- **Qclass**: For Internet addresses is 1.

```plaintext
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 bits
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                       QName (variable)                        /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         QType                 |      QClass                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 bytes
|6|rogent2|a|c3|u|p|c3|e|d|u|0|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Codification example of **rogent.ac.upc.edu**
Unit 2: IP Networks

DNS – Messages: Resource Records (RRs)

- The fields Answer, Authority and Additional are composed of **RRs**: 
  - **Name, Type, Class**: The same as in the Question field.
  - **TTL (Time To Live)**: Number of seconds the RR can be cached.
  - **RDLenth**: RR size in bytes.
  - **Rdata**: E.g. An IP address if the Type is 'A', or a name if the Type is 'NS', 'MX' or 'CNAME'.

```
+-------+---------+--------+----------------+-------+------------------+
<table>
<thead>
<tr>
<th>0 1 2</th>
<th>3 4 5 6</th>
<th>7 8 9 0</th>
<th>1 2 3 4 5 6 7</th>
<th>8 9 0 1</th>
<th>bits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+-----+---------+--------+----------------+-------+------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Name (variable)</td>
<td></td>
<td>Class</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+-----+---------+--------+----------------+-------+------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+-----+---------+--------+----------------+-------+------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TTL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+-----+---------+--------+----------------+-------+------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RDLenth</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+-----+---------+--------+----------------+-------+------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RData (variable)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+-----+---------+--------+----------------+-------+------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
Unit 2: IP Networks

DNS – Messages: Example

```
# tcpdump -s1500 -vv -p eth0 port 53

tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 200 bytes
11:17:30.769328 IP (UDP, length: 55) 147.83.30.137.1042 > 147.83.30.70.53: 36388+ A? ns.uu.net. (27)
11:17:30.771324 IP (UDP, length: 145) 147.83.30.70.53 > 147.83.30.137.1042: 36388
  q: A? ns.uu.net. 1/2/2 ns.uu.net. A 137.39.1.3
  ns: ns.uu.net. NS auth00.ns.uu.net., ns.uu.net. NS auth60.ns.uu.net.
  ar: auth00.ns.uu.net. A 198.6.1.65, auth60.ns.uu.net. A 198.6.1.181 (117)
```

Query message:
- 36388: Identifier.
- +: Recursion-Desired is set.
- A?: Qtype = A.
- ns.uu.net.: Name to resolve.

Response message:
- 36388: Identifier.
- q: A? ns.uu.net.: Repeat the Question field.
- 1/2/2: 1 Answers, 2 Authorities, 2 Additional follows.
- ns.uu.net. A 137.39.1.3: The answer (RR of type A, address: 137.39.1.3).
- ns: ns.uu.net. NS auth00.ns.uu.net., ns.uu.net. NS auth60.ns.uu.net.: 2 Authorities (RRs of type NS: the domain ns.uu.net. authorities are auth00.ns.uu.net. and auth60.ns.uu.net).
- ar: auth00.ns.uu.net. A 198.6.1.65, auth60.ns.uu.net. A 198.6.1.181: 2 Additional (RRs of type A: authorities IP addresses).
Unit 5. Network applications

Outline

- DNS
- Email
- Web
- Charsets
- HTML
Unit 5. Network applications

Email

- **Electronic mail** (email): One of the first applications used in the Internet to electronic messaging.

- **Components:**
  - **Transport layer:** TCP, well-known port: 25.
  - **Application layer protocol:** Simple Mail Transfer Protocol (SMTP). First defined by RFC-821 and last updated by RFC-5321.
  - **Retrieval protocols** (IMAP, POP, HTTP).

![Diagram of email communication](image)
Unit 5. Network applications

Email – Architecture

- **MUA**: Mail User Agent
- **MTA**: Mail Transfer Agent
Email – Protocols

“Retrieval” protocols (mailbox access):
- Post Office Protocol (POP3)
- Internet Message Access Protocol (IMAP)
- Simple Mail Transfer Protocol (SMTP)
Unit 5. Network applications

Email - SMTP processing model

- Mail server
- DNS server (DNS)
  - DNS request (Mail eXchange, MX record)
  - DNS reply (MX record)
- Mail transfer agent, MTA (sendmail, postfix, ...)
- User mailboxes
- Outgoing message queue
- Retrieval
- Client
- Mail user agent, MUA (Thunderbird, outlook, ...)

User name: llorenc@ac.upc.edu
Domain name:

*POSTFIX*
Postfix logo
http://www.postfix.org
(UNIX, free and open-source)
Unit 5. Network applications

Email - SMTP protocol (RFC-821, last update RFC-5321)

- Designed as a simple (few commands) and text-based protocol (ASCII).
- **Client basic commands:** HELO (identify SMTP client), MAIL FROM: (identify sender mailbox), RCPT TO: (identify recipient mailbox), DATA (mail message), QUIT (close transaction).
- **Server replies:** Three digit number (identify what state the client to enter next), and a human understandable message.

**Example:** Manually send an email using telnet to port 25.

```
CLIENT
linux ~> telnet relay.upc.edu 25
Trying 147.83.2.12...
Connected to relay.upc.edu.
Escape character is '^]'.

SERVER

HELO linux.ac.upc.edu
250 dash.upc.es Hello linux.ac.upc.edu [147.83.34.125], pleased to meet you
MAIL FROM: <llorenc@ac.upc.edu>
250 2.1.0 <llorenc@ac.upc.edu>... Sender ok
RCPT TO: <albert@ac.upc.edu>
250 2.1.5 <albert@ac.upc.edu>... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself

Hello world
.
250 2.0.0 p14DvFOQ008320 Message accepted for delivery
QUIT
221 2.0.0 dash.upc.es closing connection
Connection closed by foreign host.
```

**SMTP transaction**
Multipurpose Internet Mail Extensions: MIME

- Used in mail, web, etc
- Specification for “Transport” of composite multimedia objects
  - Transport type information (receiver can automatically present)
  - Encoding to enable/facilitate the transfer
- The internal format becomes invisible to users
- Include one or more objects, text in diverse alphabets, large objects (fragments, refs), alternatives, etc.
MIME: examples

From: Nathaniel Borenstein <nsb@thumper.bellcore.com>
To: Ned Freed <ned@innosoft.com>
Subject: Plain old email

This is a plain old email message.
It contains ASCII text, nothing more.

From: Nathaniel Borenstein <nsb@thumper.bellcore.com>
To: Ned Freed <ned@innosoft.com>
Subject: Plain text mail
Content-type: text/plain; charset=us-ascii

This is plain text mail.

Subject: French mail
Content-type: text/plain; charset=iso-8859-1
Content-transfer-encoding: quoted-printable

Le courrier électronique à la française ...

Content-type: image/gif
Content-Transfer-Encoding: base64

R0lGODdhSgGgAfUAAENQ01NTTw8PEVF...
MIME: example multipart

From: Nathaniel Borenstein <nsb@bellcore.com>
To: Ned Freed <ned@innosoft.com>
Subject: A multipart example
Content-Type: multipart/mixed; boundary=CUT_HERE

--CUT_HERE
Content-type: text/plain

Hey, Ned, look at this neat picture:

--CUT_HERE
Content-type: image/gif
Content-Transfer-Encoding: base64

5WVlZ6enqqqqqr....

--CUT_HERE
Content-type: text/plain

Wasn’t that neat?

--CUT_HERE--
MIME: content type

- Text: ...
  - Attribute: charset=iso-8859-1
  - text/plain (simple text), text/html ...
- Image: image/gif, image/jpeg, image/png ...
- Audio: sound, voice, music ...
- Application: application specific content
  - Application/octet-stream: data without any associated application
  - Application/organization-product
- Multipart: a set of objects
  - Mixed: a combination of several objects
  - Alternative: an object in several formats to select one (text/html/rtf)
  - Parallel: several objs for simultaneous presentation (e.g. audio+video)
  - Digest: collection of messages
  - Related: set of objects part of a single object (web page)
- Message:
  - RFC822: a complete message (eg. resent message)
  - Partial: a fragment ...
  - External-Body: a reference to an external object
MIME: transfer encoding

Ways to encode content: (to “get through” a 7 bit transport)

- **Quoted-Printable:**
  - The majority of text is 7 bits, transform some characters € → =E4
  - The result “almost” legible without decoding. Depends on table (charset)
- **Base64:**
  - 3 bytes (24 bits) <=> 4 ASCII (32 bits)
  - A-Za-z0-9+/=
  - '=' as padding, other are ignored (\r, \n, …)
- **Binary:** No encoding: any character and lines of any length
- **7Bit:** No character encoding (all 7 bits) and lines of appropriate length
- **8Bit:** No character encoding (8 bits) and lines of appropriate length

In the heading:

MIME-Version: 1.0
Subject: =?iso-8859-1?Q?acentuaci=F3n=20t=EDpica?= 

Unit 5. Network applications

Email - retrieval protocols

- Post Office Protocol (POP), RFC-1939:
  - POP server listens on well-known port 110
  - User normally deletes messages upon retrieval.

- Internet Message Access Protocol (IMAP) RFC-3501:
  - IMAP server listens on well-known port 143
  - Messages remain on the server until the user explicitly deletes them.
  - Provide commands to create folders, move messages, download only parts of the messages (e.g. only the headers)

- Web based Email (HTTP)
  - A web server handles users mailboxes. User agent is a web browser, thus, using HTTP to send and retrieve email messages.
Unit 5. Network applications

Email - Webmail

- **Web front-end for mail services.** The MUA is a web browser.
- **Real protocol to access the services:** HTTP (web).
- **The HTTP server machine uses SMTP or POP3, as required.**
Unit 5. Network applications

Outline

- DNS
- Email
- **Web**
- Charsets
- HTML
Unit 5. Network applications

Web – links

- **Uniform Resource Identifier (URI) RFC3986**
  - Generic syntax to identify a resource.
- **Uniform Resource Locator (URL) RFC1738**
  - Subset of URIs identifying the locating a resource in the Internet.
- The **URL general syntax** is
  
  `scheme://username:password@domain:port/path?query_string#fragment_id`

  - **scheme**: Purpose, and the syntax of the remaining part. http, gopher, file, ftp...
  - **domain**: name or IP address gives the destination location. The port is optional.
  - **query_string**: contains data to be passed to the server.
  - **fragment_id**: specifies a position in the html page.

- **Examples**:
  - http://147.83.2.135
  - http://studies.ac.upc.edu/FIB/grau/XC/#Practs
  - file:///home/llorenc/gestio/2010/cd/autors.html
Unit 5. Network applications

Web – HTTP Messages, RFC2616

- **Client (HTTP request):**
  - method: GET, POST...
  - request line: `GET /index.html HTTP/1.1`
  - header lines: `Host: www.example.com`
  - blank line
  - body: *(data in a POST method)*

- **Header:** Allows the client to give additional information about the request and the client itself.
  - **Host:**
    - host of the resource being requested
    - mandatory in HTTP/1.1
Unit 5. Network applications

Web – HTTP Messages, RFC2616

- **Methods**:
  - **GET**: Typical command. Requests an object.
  - **POST**: Request an object qualified by the data in the body. This data is the contents of the HTML form fields, provided by the client.
  - **HEAD**: The server returns only the header.
  - **OPTIONS**: Request communication options.
  - **PUT**: Store entity.
  - **PATCH**: Modify an existing resource.
  - **DELETE**: Delete entity.
  - **TRACE**: Final recipient echoes the received message back.
  - **CONNECT**: Used with a proxy.

- **NOTES**
  - Most used: GET, POST
  - Safe and mandatory: GET, HEAD
Unit 5. Network applications

Web – HTTP Messages, RFC2616

• **POST** uses MIME types: `application/octet-stream`, to send raw binary data, and `application/x-www-form-urlencoded`, to send name-value pairs. Example:

  request line
  
  ```
  POST /login.jsp HTTP/1.1
  Host: www.mysite.com
  User-Agent: Mozilla/4.0
  Content-Length: 27
  Content-Type: application/x-www-form-urlencoded
  ```

  header lines

  blank line

  body
  
  ```
  userid=llorenc&password=mypassword
  ```
Unit 5. Network applications

Web – HTTP Messages, RFC2616

- Server (HTTP response):
  - status line: HTTP/1.1 200 OK
  - header lines:
    - Date: Mon, 23 May 2005 22:38:34 GMT
    - Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)
    - Etag: "3f80f-1b6-3e1cb03b"
    - Accept-Ranges: bytes
    - Content-Length: 438
    - Connection: close
    - Content-Type: text/html; charset=UTF-8
  - blank line
  - body: data ....
Unit 5. Network applications

Web – HTTP Messages, RFC2616

- Header
  - **Last-Modified**: date, used in conditional retrieval.
  - **Etag**: id, used in conditional retrieval.
  - **Connection**: keep-alive/close, controls whether or not the network connection stays open after the current transaction.
  - **Accept**: `<MIME_type>/<MIME_subtype>`, acceptable mime types.
  - ...
Unit 5. Network applications

Web – Persistent/non Persistent connections

- **Non persistent** (default in HTTP/1.0): The server close the TCP connection after every object. E.g, for an html page with 10 jpeg images, 11 TCP connections are sequentially opened.

- **Persistent** (default in HTTP/1.1): The server maintains the TCP connection opened until an inactivity time. All 11 objects would be sent over the same TCP connection.

- **Persistent connections with pipelining** (supported only in HTTP/1.1): The client issues new requests as soon as it encounter new references, even if the objects have been not completely downloaded.
Unit 5. Network applications

Web – Caching and Proxies

**Caching**: The client stores downloaded pages in a local cache. Conditional GET requests are used to download pages if necessary. It can use the Date and/or Etag:

```
GET /index.html HTTP/1.1
Host: www.example.com
If-Modified-Since: October 21, 2002 4:57 PM
If-None-Match: "686897696a7c876b7e"
```

**Proxy server**: Acts as an intermediary for requests from clients.

- **Advantages**:
  - Security (the proxy may reject the access to unauthorized servers)
  - Logs
  - Caching
  - Save public IP addresses (only the proxy may have access to the Internet)
  - ...
Unit 5. Network applications

Web – web based applications

Components:
- **Presentation**: A web browser (client side).
- **Engine**: generating “on the fly” HTML pages (server side).
- **Storage**: a database (e.g. mysql).

Benefits:
- Fast to deploy and upgrade (only server side).
- Only a compatible browser is required at the client side.
- Provide cross-platform compatibility (i.e., Windows, Mac, Linux, etc.)
Languages, cultures, alphabets

7400 million people (2016)

22% speak Chinese, 11% English, 7% Spanish, 0.1% Catalan

Apart from languages, there are cultures and alphabets

- Language with several cultures: es_ES, es_CO ("locale")
- Alphabet shared by several languages (e.g. català & français)

Culture:

- Messages, character sets, transliteration, ordering, search in strings, hours and dates, numbers and currency, pronunciation, …

Interaction between agents in different languages and cultures: alphabets and character sets
Languages, cultures, alphabets

Internacionalization (i18n), Localization (l10n)

Alphabets

- "base": ascii
- National: e.g.: latin-1 (includes ascii), kanji
- International: e.g.: unicode (includes latin-1 and “all” languages)

Expression or language negotiation (in HTTP):

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

English is the default …

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

Character sets

Characters are encoded following several conventions:

- **repertoire**: a set of characters (name and representation (glyph))
- **code**: correspondence between repertoire and natural numbers.
- **encoding**: method (algorithm) to convert code numbers into a sequence of octets (> 256 characters)
- **US-ASCII**: 95 characters + control=128: 7 bits (1 octet sent)
Unit 5. Network applications

Outline

- DNS
- Email
- Web
- Charsets
- HTML
ISO 8859

- ISO 8859-1 (ISO Latin 1): 190 + control = 256: 1 octet
  - Western European, default for HTTP

- More variants
  - ISO 8859-15 extends -1 + Ÿ, €
  - ISO 8859-2 (Central European)
  - ISO 8859-4 (North European)
  - ISO 8859-5 (Cyrillic)
  - ISO 8859-6 (Arabic) — Most common Arabic glyphs
  - ISO 8859-7 (Greek)
  - ISO 8859-8 (Hebrew) — modern Hebrew.
  - ISO 8859-9 (Turkish, Kurdish)
  - ISO 8859-11 (Thai) — Contains most glyphs needed
Universal Coded Character Set

Unicode

All characters from all written languages + math + emoticons +
+=Universal Character set (ucs)

Encoding: UCS-4 bytes (fixed length)

Proportional spacing, language independent

Unicode consortium: synchronized with ☺ ☻ ☼ ☽ ☾ ☽

- Unicode 9.0.0 (7/2016): 128,172 symbols
- U+hex code: U+0020 = ' ' 

Character Encodings: Universal Transformation Format (UTF)

- Difficulty or impossibility to transport 8 o 16 bits data in Internet
  protocols:
- **UTF-8**, UTF-16, UTF-32 (variable length)

http://www.unicode.org
Variable length encodings

- **UTF-8 (8 bits) (rfc2044)**
  - One to four 8-bit code units
  - Most common in the Internet
  Content-Type: text/plain; charset=UTF-8
  Content-Transfer-Encoding: 8bit
  Català, Français, Tämä on testi.

- **UTF-16 (16 bits)**
  - One or two 16-bit code units

- **UTF-32 (32 bits)**
  - Fixed-length 32-bit code units
Universal Coded Character Set
Unicode

- **UTF-8 Encoding**
  - Determine high-order bits from the number of octets
  - Fill in the bits marked x

<table>
<thead>
<tr>
<th>Char. number range</th>
<th>UTF-8 octet sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>(hexadecimal)</td>
<td>(binary)</td>
</tr>
<tr>
<td></td>
<td>----------------------</td>
</tr>
<tr>
<td>0000 0000–0000</td>
<td>0xxxxxxx</td>
</tr>
<tr>
<td>0000 0080–0000</td>
<td>110xxxxx 10xxxxxx</td>
</tr>
<tr>
<td>0000 0800–0000</td>
<td>1110xxxx 10xxxxxx 10xxxxxx</td>
</tr>
<tr>
<td>0001 0000–0010</td>
<td>11110xxx 10xxxxxx 10xxxxxx 10xxxxxx</td>
</tr>
</tbody>
</table>

- **Example**
  - character: €
  - code point: U+20AC
  - code point in binary (12 bits): 10 0000 1010 1100
  - 3 code units required:
    - UTF-8: 11100010 10000010 10101100
    - UTF-8 in hex: E282AC
Unit 5. Network applications

Outline

- DNS
- Email
- Web
-Charsets
- HTML
Unit 5. Network applications

HTML – Hyper-Text Markup Language, HTML

Tim Berners-Lee defined HTML in 1989. HTML design mail goal was displaying formatted text documents with hyperlinks (including links to other documents) in web browsers.

Based on tags e.g. <head> data </head>

Example:

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

Terminology:
- element
- attribute
- text

First Heading

first paragraph.
Unit 5. Network applications

HTML – Hyper-Text Markup Language, HTML

- HTML features (1):
  - Hyperlinks: Click on a link and jump to another document
  - Forms: The document accept user inputs that are sent to the server
  - Scripting: Allow adding programs. The program executes on the client's machine when the document loads, or at some other time such as when a link is activated.

- Hyperlinks
  - `<a>` tag defines an hyperlink

  - Syntax:
    » `<a href="url">link text</a>`

  - Example:
    » `<a href="https://studies.ac.upc.edu/FIB/grau/XC/">XC-GRAU</a>`
Unit 5. Network applications

HTML – Hyper-Text Markup Language, HTML

- HTML features (2):
  - Javascript example:

```html
<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>
```