Computer Nerworks. Unit 5: APPs

Notes of the subject *Xarxes de Computadors, Facultat Informàtica de Barcelona, FIB*

Llorenç Cerdà-Alabern

May 22, 2019

Contents

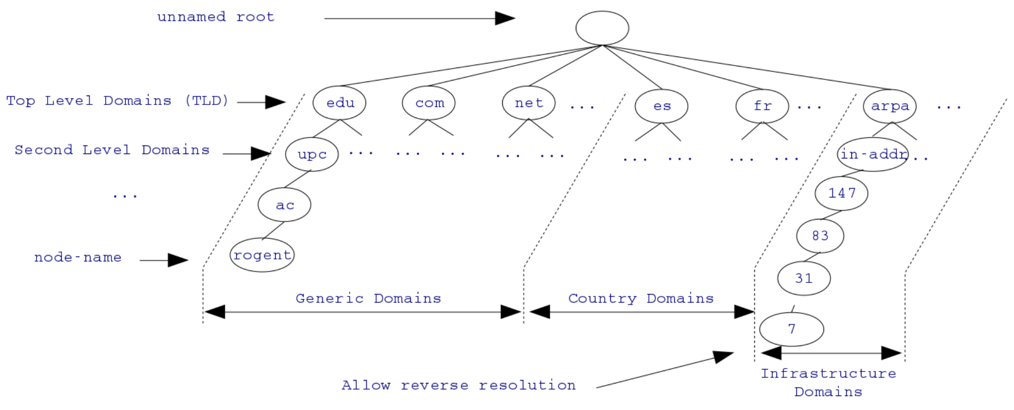
5 Unit 5: APPs

5.1 Domain Name System, DNS

5.1.1 DNS fundamentals

- Translate **names to IP** addresses
- name format: **node.domain** e.g. **rogent.ac.upc.edu**
- **Case insensitive**: www.upc.edu = WWW.UPC.EDU
- **UDP**, well-known port: 53
- **Distributed DB** in Name Servers, **NS**
- DB entries are called Resource Records (**RR**)

5.1.2 DNS hierarchy



Terminology

- **rogent.ac.upc.edu.** is a **name**
- **rogent** is the **node name**
- **ac.upc.edu.** is the **domain**

- ac.upc.edu. is a **subdomain** of upc.edu.
- The last dot represents the **root**
- The last dot is optional
- Internet Corporation for Assigned Names and Numbers, **ICANN**: DNS management and coordination
- ICANN delegates Top Level Domains (**TLD**) to registrars
- Registrars delegate **subdomains**
- Subdomains have an **authority**:
  - Consists of **primary** and **backup** NS
  - Store the domain **data base**
  - Data base entries are called **Resouce Records**, **RR**

5.1.3 Client configuration

1. The **applications** use the OS resolver library:

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

1. The **resolver** first looks the **/etc/hosts** file:

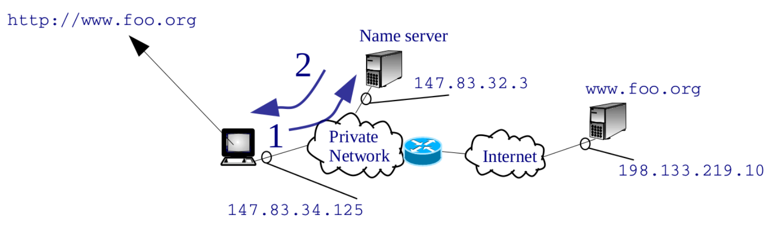
```
# Syntax:
# IP-Address  Full-Qualified-Hostname  Short-Hostname
 127.0.0.1       localhost
 10.0.1.1        mypc.ac.upc.edu mypc
```

1. Otherwise a **local NS** using **/etc/resolv.conf** file:

```
search ac.upc.edu
nameserver 147.83.32.3
nameserver 147.83.33.4
```

5.1.4 Client DNS resolution

1. **DNS request** to local NS
2. **DNS reply** from local NS



Practical example Capture a DNS resolution

```
wireshark
nslookup
```

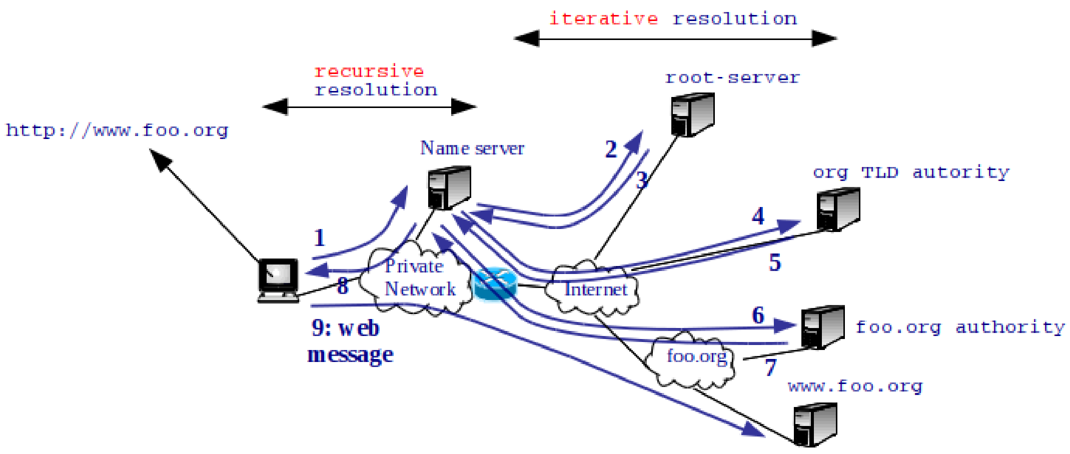
5.1.5 root-servers

- Entry point to the domain hierarchy
- Distributed around the world
- Have the TLD addresses: `http://www.root-servers.org`
- Root server addresses are needed in a NS configuration

Legend
Multiple instances
Single instance

5.1.6 Server DNS resolution

- NSs **cache** name resolutions
- Cached RR are returned, otherwise:



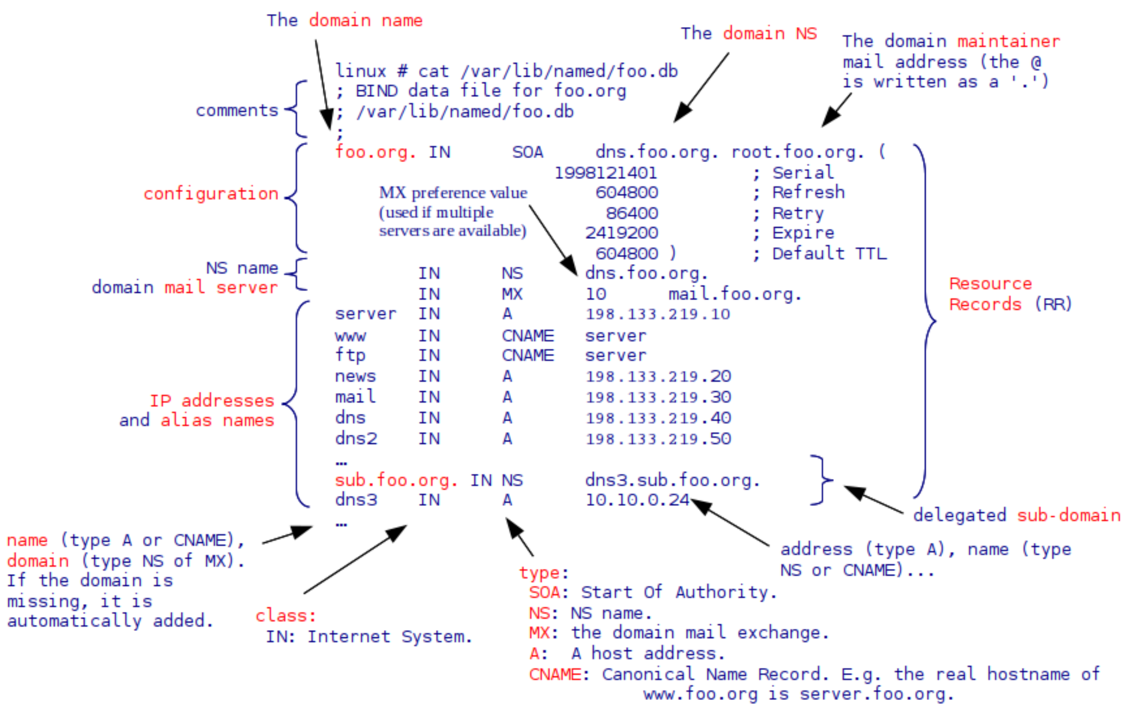
NOTE: a NS asks other NS using **iterative** resolutions

5.1.7 Server configuration

- BIND (Berkeley Internet Name Domain)
- **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
```

Zone file



```
linux # cat /var/lib/named/foo.db
; BIND data file for foo.org
; /var/lib/named/foo.db
;
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
…
```

The domain name
The domain NS
The domain maintainer mail address (the @ is written as a '.')
comments
configuration
MX preference value (used if multiple servers are available)
NS name
domain mail server
Resource Records (RR)
IP addresses and alias names
delegated sub-domain
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.
address (type A), name (type NS or CNAME)...

Resource Record (RR) types

- **SOA** Start of Authority: administrative information
- **NS** NS name
- **MX** the domain Mail eXchange server
- **A** A host address
- **CNAME** Canonical Name Record
- **PTR** Reverse resolution: IP to name

```
pc.example.com.        IN    A      1.2.3.4
4.3.2.1.in-addr.arpa.  IN    PTR    pc.example.com.
```

5.1.8 DNS Messages

- **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

```
------------------------------------------------
|                Header (12 bytes)             |
------------------------------------------------
/                Question (variable)           /
------------------------------------------------
/                Answer (variable)             /
------------------------------------------------
/                Authority (variable)          /
------------------------------------------------
/                Additional (variable)         /
------------------------------------------------
```

Header Field

- **Identification**: 16 random bits used to match query/response
- **Flags**. Some of them:
  - Query-Response
  - Authoritative Answer
  - Recursion Desired

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Identification          |        Flags                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         #Questions              |        #Answers             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         #Authorities            |        #Additional          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Question Field

- **QName**: the name to be resolved
- **QType**: 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.

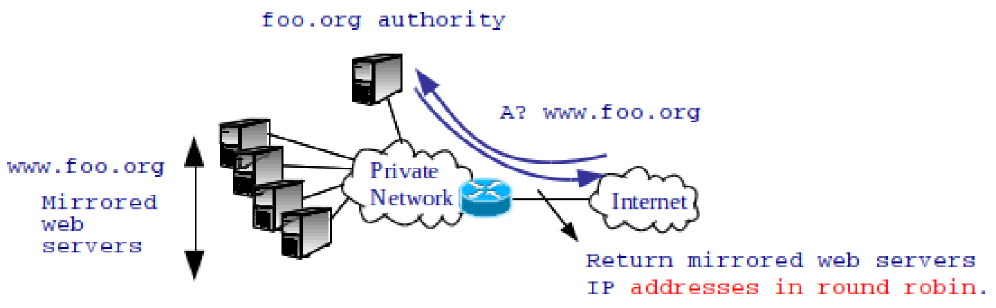
```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 bits
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                       QName (variable)                        /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         QType                   |        QClass               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Resource Record Fields

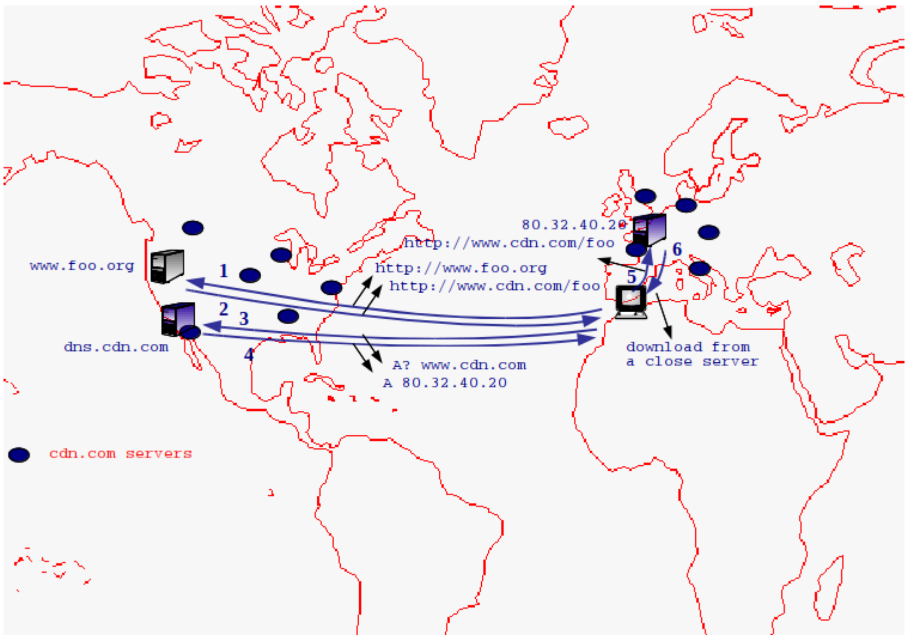
- Answer, Authority and Additional are RRs
  - **Name**, **Type**, **Class**: as in the Question field
  - **TTL** (Time To Live): Number of seconds the RR can be cached.
  - **RDLenth**: RR size in bytes.
  - **Rdata**: IP address if the Type is ’A’, a name if the Type is ’NS’, ’MX’ or ’CNAME’

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                        Name (variable)                        /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Type                   |         Class               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              TTL                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         RDLenth                 |        RData (variable)     /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

5.1.9 Load balancing with DNS



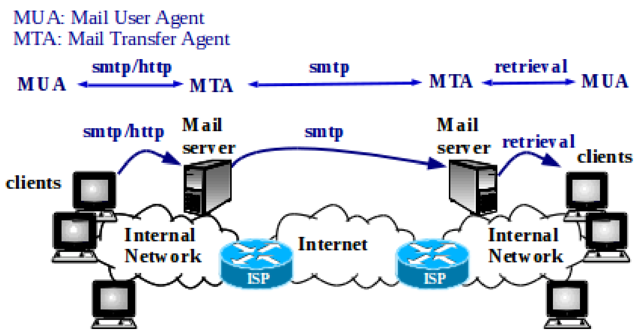
5.1.10 Content Distribution Networks, CDN



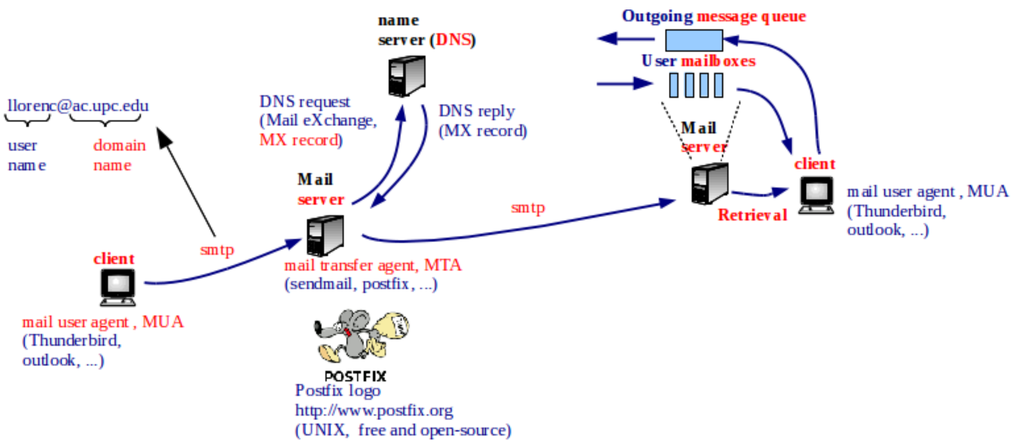
5.2 Email

5.2.1 Email Fundamentals

- One of the first Internet applications
- Transport layer: **TCP**, well-known port: **25**
- Simple Mail Transfer Protocol, SMTP, RFC821
- email text messages RFC822
- **Retrieval** protocols: IMAP, POP, HTTP



5.2.2 Email Fundamentals


name server (DNS)
Outgoing message queue
User mailboxes
llorenc@ac.upc.edu
user name
domain name
DNS request (Mail eXchange, MX record)
DNS reply (MX record)
Mail server
Mail server
client
Retrieval
mail user agent, MUA (Thunderbird, outlook, ...)
smtp
client
smtp
mail transfer agent, MTA (sendmail, postfix, ...)
mail user agent, MUA (Thunderbird, outlook, ...)
POSTFIX
Postfix logo
http://www.postfix.org
(UNIX, free and open-source)

5.2.3 SMTP RFC821

- Designed as a simple (few commands) and text-based protocol (**ASCII**)
- **Client** basic commands (case sensitive):
  - **HELO host** identify SMTP client
  - **MAIL FROM: email** identify sender mailbox
  - **RCPT TO: email** identify recipient mailbox
    - **More than one RCPT** are allowed for multiple recipients
  - **DATA** mail text message: RFC822
  - **QUIT** close transaction
- **Server replies**: Three digit number (what state the client to enter next), and a human understandable message

5.2.4 Email text messages RFC822

- **ASCII**
- **Format**:
  - Header (field: data, fields are case insensitive)
  - Blank Line
  - Body
  - .
- Example:

```
from: a@b.c
to: b@b.c
subject: test

helo world
.
```

- **Practical example**

Send an email manually using the UPC Mail server (bash)

```
telnet relay.upc.edu 25
```

5.2.5 Retrieval Protocols

- Post Office Protocol, **POP** RFC1939
  - User normally **deletes** messages from server upon retrieval.
- Internet Message Access Protocol, **IMAP** RFC3501
  - Messages **remain** on the server until the user deletes them.
  - Provide **commands** to create folders, move messages, etc.
- Web based Email, **HTTP**
  - A web server handles users mailboxes. User agent is a web browser: HTTP to send and retrieve

5.2.6 Multipurpose Internet Mail Extensions (MIME), RFC2045

- Extends email text messages RFC822 by defining new header fields
- Inclusion of **non-ASCI data** (e.g. files, images, audio, video...)
- Basic **MIME header fields**
  - MIME-version
  - Content-type
  - Content-transfer-encoding

```
MIME-version: 1.0
Content-Type: text/plain; charset=utf-8
Content-transfer-encoding: base64
```

Content-type

- Describe the data contained in the body

```
MIME-version: 1.0
Content-Type: text/plain; charset=utf-8
Content-transfer-encoding: base64
```

- **Type/subtype** (mantained by IANA) **; attributes**
  - **Types** application, audio, image, message, multipart, text ...
  - **Subtypes** application/ **pdf**, image/ **gif**, text/ **html** ...
  - **Attributes** text/plain; **charset=utf-8**, ...

Content-transfer-encoding

- Re-encode data into 7-bit printable format

```
MIME-version: 1.0
Content-Type: text/plain; charset=utf-8
Content-transfer-encoding: base64
```

- Invertible mapping between the original "binary" and transferred data
  - **binary** NO encoding
  - **base64** RFC3548
    - Encodes a binary string into **printable ASCII** chars (64 chars)
    - Overhead: 8/6 (the binary file increases **33%**)
  - **Quoted printable** Wikipedia
    - Printable ASCII characters are not encoded
    - Otherwise encoded as **=HH** where HH is the hex code
    - E.g. 00001100 (12) is encoded as **=0C**

Multipart messages

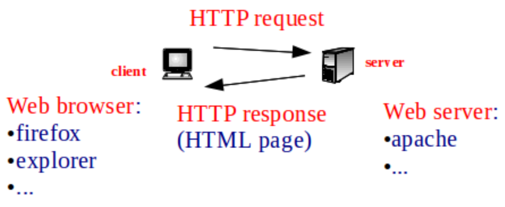
- One or more different sets of data

```
From: Nathaniel Borenstein <nsb@bellcore.com>
To: Ned Freed <ned@innosoft.com>
Subject: A multipart example
MIME-version: 1.0
Content-Type: multipart/mixed; boundary=UNIQUE_BOUNDARY

--UNIQUE_BOUNDARY
Content-type: text/plain

Hey, Ned, look at this neat picture:
--UNIQUE_BOUNDARY
Content-type: image/gif
Content-Transfer-Encoding: base64

5WVlZ6enqqqr....
--UNIQUE_BOUNDARY
```

5.3 World Wide Web, www

5.3.1 www fundamentals

- Started by **Tim John Berners-Lee** in 1989.
- Transport layer: **TCP**, well-known port **80**
- HyperText Transfer Protocol, **HTTP** RFC1945 (HTTP-1.0), RFC2616 (HTTP-1.1)
- HyperText Markup Language, **HTML**: Language used to format web documents.



5.3.2 www links

- Uniform Resource Locator, **URL** RFC1738. Resource in the Internet.

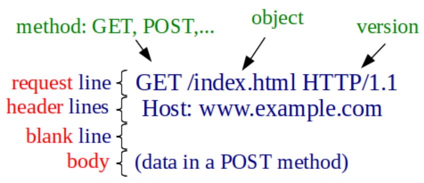
```
scheme://username:passw@domain:port/path?query_string#fragment_id
query_string: variable1=value1&variable2=value2&...
```

- **Examples**

```
http://tools.ietf.org/html/rfc1738
http://147.83.2.135
http://studies.ac.upc.edu/FIB/grau/XC/#Practs
file:///home/llorenc/gestio/2010/cd/autors.html
http://www.amazon.com/product/refs9?pf_ra=ATVPD&pf_rd=07HR2
```

5.3.3 HTTP Messages RFC2616

- Client **HTTP request**



Host header field RFC2616

- host of the resource being requested
- mantadory in HTTP/1.1

HTTP Methods

1. **GET** requests an object
2. **POST** request qualified by the data in the body, e.g. HTML form fields provided by the client
3. **HEAD** the server returns only the header
4. **OPTIONS** request communication options
5. **PUT** store entity
6. **PATCH** RFC5789 modify an existing resource
7. **DELETE** delete entity

8. **TRACE** final recipient echoes the received message back
9. **CONNECT** used with a proxy

NOTES

- **Most used** GET, POST
- **Safe and mandatory** GET, HEAD

Practical Example

Download a web page using telnet HTTP 1.0 (bash)

```bash
telnet www.upc.edu 80
GET / HTTP/1.0
```

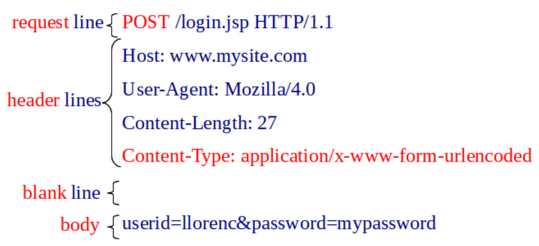
HTTP Header

- **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$_{\text{type}}$>/<MIME$_{\text{subtype}}$>**, acceptable mime types.
- ...
- **Practical example:**

Download a header using telnet HTTP 1.0 (bash)

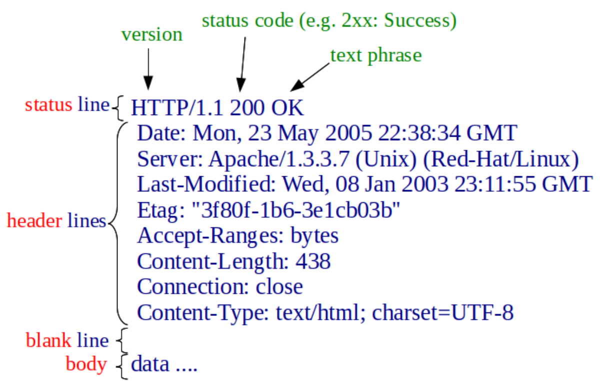
```bash
telnet www.upc.edu 80
HEAD / HTTP/1.0
```

- HTTP uses **MIME**, e.g. POST
  - **application/octet-stream** send raw binary data
  - **application/x-www-form-urlencoded** send name-value pairs:



```
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

userid=llorenc&password=mypassword
```

- Server **HTTP response**



```
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
Etag: "3f80f-1b6-3e1cb03b"
Accept-Ranges: bytes
Content-Length: 438
Connection: close
Content-Type: text/html; charset=UTF-8

data ....
```

5.3.4 HTTP status codes

- **1xx** Informational - Request received, continuing process
- **2xx** Success
- **3xx** Redirection - Further action must be taken
- **4xx** Client Error
- **5xx** Server Error
- **Practical example** capture an HTTP response

```
$ telnet www.upc.edu 80
Trying 147.83.2.135...
Connected to www.upc.es.
Escape character is '^]'.
GET / HTTP/1.0

HTTP/1.1 301 Moved Permanently
Date: Sun, 03 Jun 2018 18:23:44 GMT
Content-Length: 0
Connection: close
Location: http://www.upc.edu/ca
...
```

5.3.5 Persistent/non Persistent connections

- **Non persistent** (default in **HTTP/1.0**): The server close the TCP connection after every object.
- **Persistent** (default in **HTTP/1.1**) : The server maintains the TCP connection opened until an inactivity time.
- 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.

Practical example

Download a web page using telnet HTTP 1.0 (bash)

```
telnet www.upc.edu 80
GET / HTTP/1.0
```

Download a web page using telnet HTTP 1.1 (bash)

```
telnet www.upc.edu 80
GET / HTTP/1.1
host: www.upc.edu
```

5.3.6 Caching

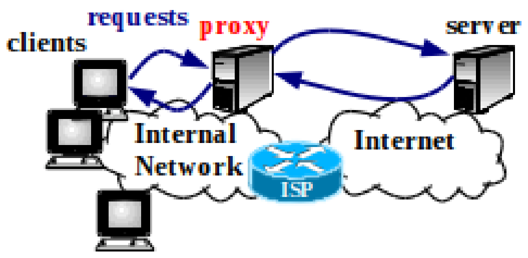
- The client **stores** downloaded pages in a local **cache**
- **Conditional GET** requests are used to download pages if necessary. It can use the **Date** and/or **Etag**:

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

5.3.7 Web proxy

- 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)
  - ...



5.4 HTML

5.4.1 HTML fundamentals

- 1986 ISO Standard Generalized Markup Language, SGML, for documentation projects
- Introduced the **<>** syntax
- HTML is inspired in SGML

HTML Example (html)

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

HTML Terminology

- tag: < **head** >
- element: **<title>Basic html document</title>**
- attribute: <font **color="red"** > ... </font>
- text: <p> **first paragraph.** </p>

5.4.2 HTML features

- **Forms** user inputs that are sent to the server
- **Scripting** program executed on the client's machine
- HTML Cascading Style Sheets, **CSS**

Form example

Form example (html)

```html
<html>
  <h1>HTML Form Example</h1>
  <form action="http://localhost/~llorenc/php-example.php" target="_blank">
    First name:<br>
    <input type="text" name="firstname" value=""><br>
    Last name:<br>
    <input type="text" name="lastname" value="">
    <br><br>
    <input type="submit">
  </form>
</html>
```

php example (html)

```
<html>
<?php
 $name  = $_GET["firstname"] ;
 $lname = $_GET["lastname"] ;
 echo "Hi $name $lname" ;
?>
</html>
```

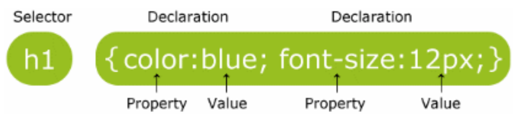
Javascript example

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>
```

HTML Cascading Style Sheets, CSS

- Allows describing the **layout** in a separate document
- CSS **syntax**



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

- Example CSS file **base.css**

CSS example (html)

```
h1 {
    color: red;
}
body {
  font-family: "Times New Roman", serif;
  color: blue;
}
```

Reference to the external style sheet base.css

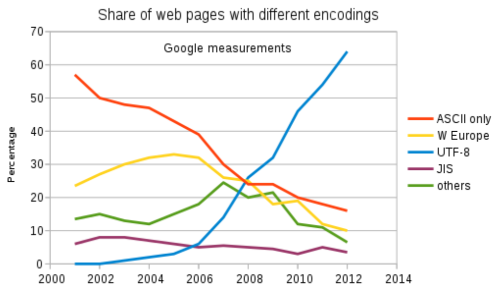
```
<html>
<head>
  <link type="text/css" href="base.css" rel="stylesheet" />
</head>
<body>
<h1>CSS Test</h1>
bla bla...
</body>
</html>
```

NOTE: **rel** attribute specifies the relationship with the **link**
CSS Tutorial

5.5 Charsets

5.5.1 Standards

- ASCII
- ISO-8859-1 AKA **latin-1**
  - 1 byte extended ASCII including latin specific characters used in different languages
- Windows-1252 similar to ISO-8859-1
- JIS Japanese language
- unicode (from UNI-fy en-CODE-ings) / ISO-10646
- ...



5.5.2 Charsets usage examples

- Computer **terminal**: environment variables
- IEEE-POSIX (Portable **Operating System** Interface)
  - **locale variables** = language[$_{\text{territory}}$][.codeset]
  - E.g. `LANG=ca_ES.UTF-8`
- email & web: **MIME Content-Type**: text/plain; **charset=UTF-8**

5.5.3 Charsets terminology (unicode)

- **characters**: smallest interpretable units of stored text
- **character repertoire**: name and representation of characters
- **glyphs**: shapes of characters when rendered. A repertoire of glyphs makes up a **font**
- **code point**: mapping characters repertoire <-> numbers
- **encoding**: algorithm code point <-> one or more **code units**

E.g. unicode Greek characters

5.5.4 Unicode

- Characters from all languages + math + emoticons + ...
- Unicode 9.0.0 (7/2016): 128.172 symbols
- **notation**: **U+hex code point**. E.g. U+0020 = ' ' (blank space)
- first 128 code points correspond to ASCII
- **encoding**: Unicode Transformation Format, **UTF**
- UTF-code units (bits): UTF-7, UTF-8, UTF-16, UTF-32

5.5.5 UTF-8 (wikipedia)

- **Dominant** unicode encoding
- **Internet** standard RFC3629
- **Preserves ASCII** codes
- **Variable length**
- **Encoding** algorithm. Given the code point U+ **1.** Determine high-order bits from the **number of octets 2.** Fill in the bits marked **x**

```
 Char. number range |   UTF-8 octet sequence
   (hexadecimal)    |         (binary)
--------------------+-----------------------------------
0000 0000-0000 007F | 0xxxxxxx
0000 0080-0000 07FF | 110xxxxx 10xxxxxx
0000 0800-0000 FFFF | 1110xxxx 10xxxxxx 10xxxxxx
0001 0000-0010 FFFF | 11110xxx 10xxxxxx 10xxxxxx 10xxxxxx
```

UTF-8 example:

- character: **€**
- code point: U+20AC
- code point in bynary (12 bits): **10 0000 1010 1100**
- 3 code units required:
- UTF-8:

  ```
  11100010 10000010 10101100
  ```

- UTF-8 in hex: E282AC