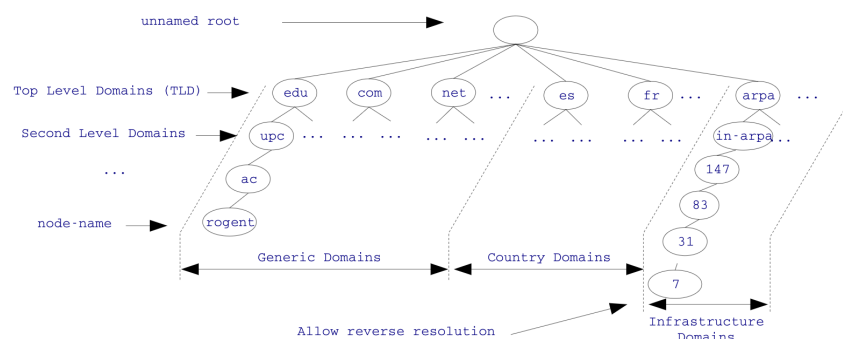


Tema 5. Aplicacions

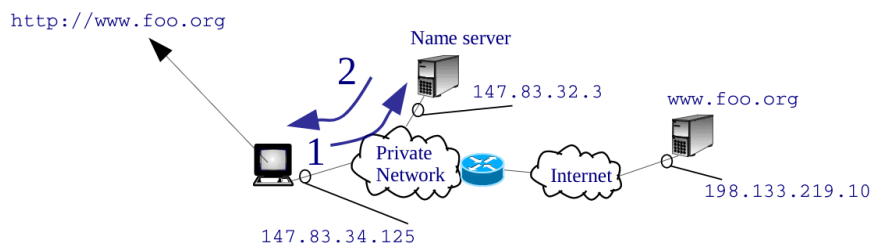
DNS (Domain Name System)

- Nom -> @IP
- Nom: rogent.ac.upc.edu
 - node-name: rogent
 - domain-name: ac.upc.edu
- DNS = worldwide distributed data base
- Entrades de la bd: Resource Records (RR)
- Informació 1 nom >= 1 RR
- Noms: case insensitive (www.upc.edu = WWW.UPC.EDU)

- Organització (Hierarchy)

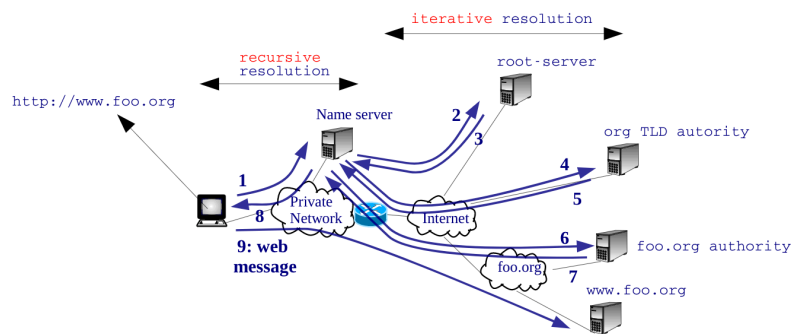


- Accés a la base de dades del DNS amb: Name Servers (NS)
- NS poden tenir RR a la cache -> es treuen després d'un timeout
- Cada subdomini té una "authority" -> NSs primària i de còpia de seguretat
- Una "authority" té:
 - Names and addresses of all nodes within the zone (subdomain).
 - Names and addresses of all subzone (delegated subdomains).
- DNS protocol: (port 53, UDP)



- DNS resolution:

- NSs cache name resolutions
- Es retorna un RR en cache sense haver de buscar a l'autoritat al NS



- DNS messages:

- Tots els missatges dns tenen el mateix format
 - Header: type of message. (12 bytes)
 - Question: What is to be resolved. (variable)
 - Answer: Answer to question. (variable)
 - Authority: Domain authority names. (variable)
 - Additional: Typically, the authority name's addresses. (variable)

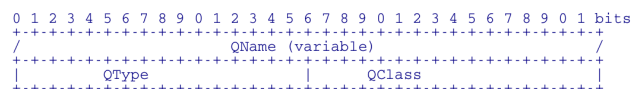
• Header:

- Identification: 16 bits aleatoris utilitzats per coincidir amb la consulta / resposta
- Flags:
 - Query-Response, QR
 - Authoritative Answer, AA
 - Recursion Desired, RD



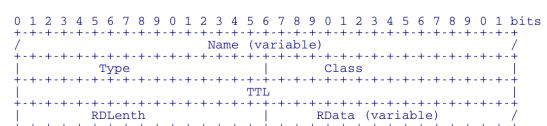
• Question:

- QName: Indica el nom que s'ha de resoldre.
- QType: Indica el tipus de pregunta
 - A: @
 - NS: name server
 - PTR: pointer (resolució inversa)
 - MX: mail exchange -> @ del servidor de correu del domini



- RR: Resource Records

- Els camps de Answer, Authority i Additional es componen de RRs:
 - Name, Type, Class: igual que en el camp de la pregunta.
 - TTL (Time To Live): nombre de segons que es pot emmagatzemar en memòria cau.
 - RDLenth: mida de RR en bytes.
 - Rdata: Exemple -> Una @IP si el tipus és 'A', o un nom si el tipus és 'NS', 'MX' o 'CNAME'.

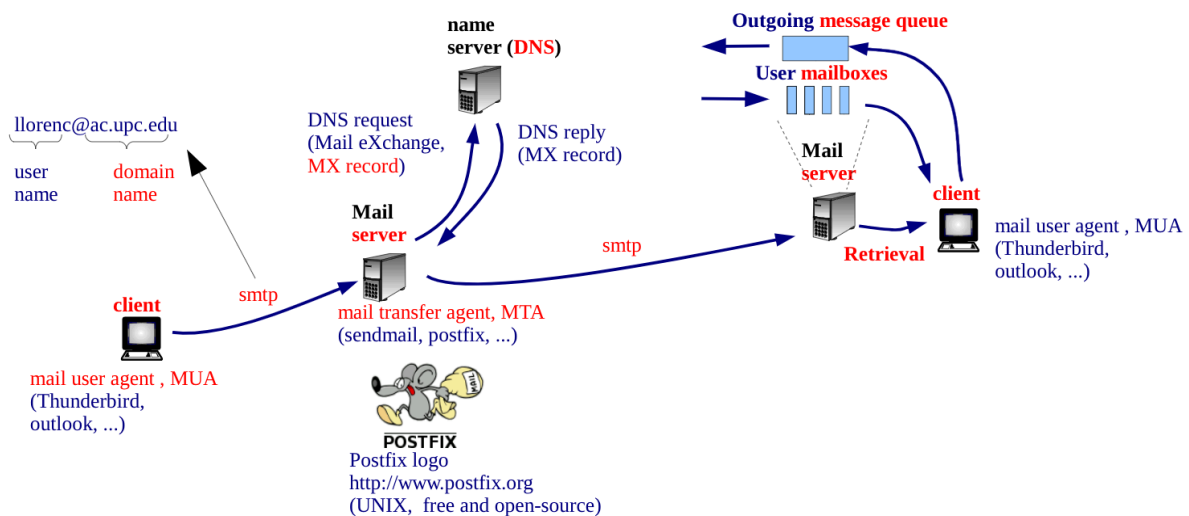
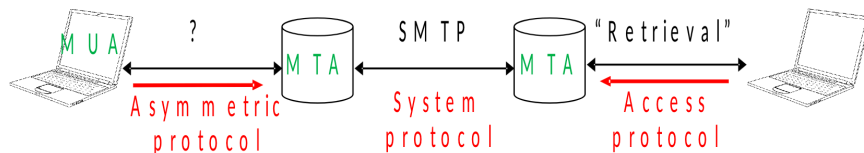
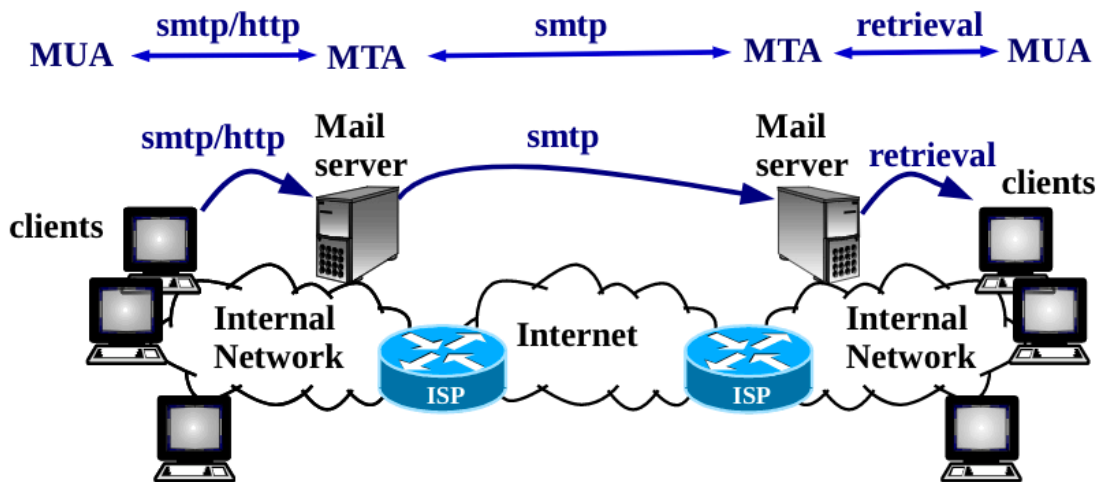


Email (Electronic Mail)

- Una de les primeres aplicacions que es fan servir a Internet per a missatges electrònics.
- Components:
 - Capa de transport: TCP, port 25
 - Protocol de la capa d'aplicació: Simple Mail Transfer Protocol (SMTP).
- Protocols de recuperació (IMAP, POP, HTTP) "Retrieval"
 - POP3 (Post Office Protocol)
 - IMAP (Internet Message Access Protocol)

MUA: Mail User Agent

MTA: Mail Transfer Agent



- SMTP protocol:
 - text-based protocol (ASCII)
 - Comandes bàsiques del client:
 - HELO (identify SMTP client)
 - MAIL FROM (identify sender mailbox)
 - RCPT TO (identify recipient mailbox)
 - DATA (mail message)
 - QUIT (close transaction)
 - Servidor contesta: Tres dígitos (identifica estat del client) + missatge comprensible.
- Multipurpose Internet Mail Extensions: MIME
 - S'utilitza per correu, web, etc.
 - Especificació per a "Transport" d'objectes multimèdia compostos
 - Informació del tipus de transport (el receptor pot presentar-se automàticament)
 - Codificació per habilitar / facilitar la transferència
 - El format intern es fa invisible per als usuaris
 - Inclou un o més objectes, text en diversos alfabetes, objectes grans (fragments, refs), alternatives, etc.

```

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

5wVlZ6enqqqr....
--CUT_HERE
Content-type: text/plain

Wasn't that neat?
--CUT_HERE--

```

- Retrieval protocols (protocols de recuperació)
 - Post Office Protocol (POP) port 110
 - L'usuari normalment elimina els missatges després de la recuperació.
 - Internet Message Access Protocol (IMAP) port 143
 - Els missatges romanen al servidor fins que l'usuari els suprimeix explícitament.
 - Proporciona comandes per crear carpetes, moure missatges, baixar només parts dels missatges (Ex: Només els headers)
 - Web based Email (HTTP)
 - Un servidor web controla les bústies dels usuaris. El User Agent és un navegador web, per tant, mitjançant HTTP per enviar i recuperar missatges de correu electrònic.

Web

- Links:
 - Tipus:
 - Uniform Resource Identifier (URI): Sintaxi genèrica per identificar un recurs.
 - Uniform Resource Locator (URL): Subconjunt d'URIs que identifica la localització d'un recurs a Internet.
 - La sintaxi general de l'URL és:

scheme://username:password@domain:port/path?query_string#fragment_id

 - scheme: Propòsit i sintaxi de la part restant. http, gopher, arxiu, ftp ...
 - domain: El nom o @IP us donen la ubicació de destinació (port opcional)
 - query_string: Conté dades que es passaran al servidor.
 - fragment_id: Especifica una posició a la pàgina html.

- Missatges HTTP
 - Client (HTTP request):

```

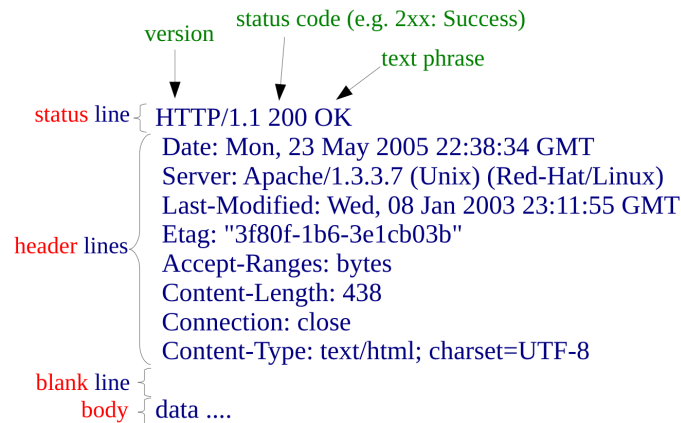
method: GET, POST,...      object      version
                             ^           ^
request line { GET /index.html HTTP/1.1
header lines { Host: www.example.com
blank line {
body { (data in a POST method)
  
```

- Methods:
 - GET: Comanda típica. Sol·licita un objecte.
 - POST: Sol·licita un objecte qualificat per les dades en el cos. Aquestes dades són el contingut dels camps del formulari HTML, proporcionats pel client....
- Header: permet al client proporcionar informació addicional sobre la sol·licitud i el propi client.
- POST uses MIME types:
 - application/octet-stream, per enviar dades binàries sense processar.
 - application/x-www-form-urlencoded, per enviar parells de name-value.

```

request line { POST /login.jsp HTTP/1.1
header lines { Host: www.mysite.com
               User-Agent: Mozilla/4.0
               Content-Length: 27
               Content-Type: application/x-www-form-urlencoded
blank line {
body { userid=llorenc&password=mypassword
  
```

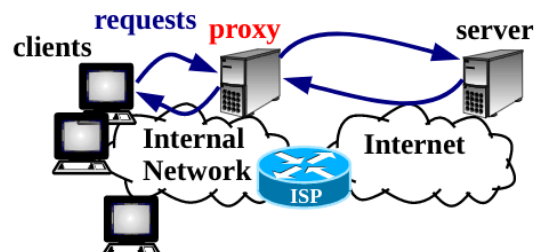
- Server (HTTP response):



- Persistent/non Persistent connections
 - Non persistent (default in HTTP/1.0): el servidor tanca la connexió TCP després de cada objecte.
 - Ex: per a una pàgina html amb 10 imatges JPEG, 11 connexions TCP s'obre de manera seqüencial.
 - Persistent (default in HTTP/1.1): el servidor manté la connexió TCP obert fins a un temps d'inactivitat.
 - Ex: All 11 objects would be sent over the same TCP connection.
 - Persistent connections with pipelining (supported only in HTTP/1.1): el client emet noves sol·licituds tan aviat com trobi noves referències, fins i tot si els objectes no s'han baixat completament.
- Caching and Proxies
 - Caching: el client emmagatzema pàgines descarregades en una memòria cau local. Les GET requests condicionals s'utilitzen per descarregar pàgines si és necessari. Es pot utilitzar la Data i / o 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: Actua com a intermediari per a requests de clients.
 - Avantatges:
 - Seguretat (el proxy pot rebutjar l'accés a servidors no autoritzats)
 - Registres
 - Caching
 - Desa adreces IP públiques (només el proxy pot tenir accés a Internet)
 - ...



- Web based applications
 - Components:
 - Presentation: Un navegador web (costat del client).
 - Engine generating “on the fly” HTML pages (server side).
 - Languages:
 - Java.
 - Hypertext Preprocessor (PHP): Elenguatge de programació incrustat i codi HTML (<http://www.php.net>).
 - Other: ASP, CGI, ColdFusion, Perl, Python...
 - Storage: a database (e.g. mysql).
 - Benefits:
 - Ràpida per implementar i actualitzar (només el costat del servidor).
 - Només cal un navegador compatible al client.
 - Proporciona compatibilitat entre plataformes (és a dir, Windows, Mac, Linux, etc.)

HTML & XML

- HTML: Hyper-Text Markup Language

- In 1986 ISO standardized the Standard Generalized Markup Language (SGML).
- SGML introduced the <> syntax, and has been used in large documentation projects.
- Tim Berners-Lee va definir HTML el 1989 inspirat en SGML. L'objectiu de correu electrònic de disseny HTML mostrava documents de text amb enllaços (inclosos enllaços a altres documents) en navegadors web.
- Based on tags e.g. <head> data </head>

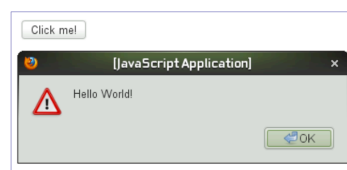
```
<html>
<head>
<title>Basic html document</title>
</head>
<body>
<h1><font color="red">First Heading</font></h1>
<p>first paragraph.</p>
</body>
</html>
```

First Heading

first paragraph.

- Terminology: **Element**, **Attribute**, **Text**
- HTML features (1):
 - Forms: el document accepta entrades d'usuari que s'envien al servidor
 - Scripting: permet afegir programes. El programa s'executa a la màquina del client quan es carrega el document o en algun altre moment, com ara quan s'activa un enllaç.
- Exemple amb 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>
```



- HTML features (2):
 - Cascading Style Sheets, CSS: permet descriure el disseny físic en un document separat. Per exemple. milers de pàgines HTML poden utilitzar el mateix CSS. Si l'estil s'ha de canviar, només cal actualitzar el CSS.
 - CSS Syntax:



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

- CSS exemple:

- Content of the file "mystyle.css":

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

```
<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.

- XML: eXtensible Markup Language

- Dissenyat per transportar i emmagatzemar dades (HTML per mostrar dades).
- Els usuaris defineixen les seves pròpies etiquetes per descriure estructures d'informació → Les processen automàticament amb aplicacions.

- Tree structure.

- Elements, attributes & text.

- Example:

```
<book category="COOKING">
  <title lang="en">Everyday Italian</title>
  <author>Giada De Laurentiis</author>
  ...
</book>
```

- Un document XML ben format satisfà una llista de regles de sintaxi proporcionades a l'especificació.
- És més rígid que HTML (p. Ex., Totes les etiquetes han de ser tancades: <tag> </tag> o <tag attribute1 = ... />).
- XML namespaces
 - Permet els noms de elements diferenciats definits per diferents desenvolupadors.
 - El namespace es defineix per l'atribut xmlns a l'etiqueta d'inici d'un element.
 - L'URL s'utilitza sovint com una manera senzilla de definir espais de noms "unique".

```
<widget xmlns="http://www.widget.org"
  xmlns:xhtml="http://www.w3.org/TR/xhtml1"
  type="gadget">
  <head size="medium"/>
  <big><subwidget ref="gizmo"/></big>
  <info>
    <xhtml:head>
      <xhtml:title>Description of gadget</xhtml:title>
    </xhtml:head>
    <xhtml:body>
      <xhtml:h1>Gadget</xhtml:h1>
      A gadget contains a big gizmo
    </xhtml:body>
  </info>
</widget>
```

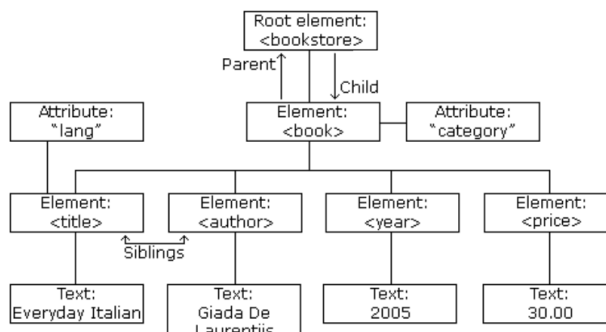
Source: <http://www.brics.dk/~amoeller/XML/xml/htmlvsxml.html>

default namespace.

namespace with prefix xhtml.
The prefix acts as a shorthand for the namespace.

- Els documents XML tenen una tree structure:

```
<bookstore>
  <book category="COOKING">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="CHILDREN">
    <title lang="en">Harry Potter</title>
    <author>J. K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  ...
</bookstore>
```



- Terminology: Element, Attribute, Text

- Validation of XML documents
 - Un document XML "Valid" s'ajusta a la sintaxi d'un esquema XML.
 - L'esquema XML defineix les etiquetes vàlides i com es poden utilitzar.
- Most known schema languages:
 - Document Type Definition, DTD:
 - Primer llenguatge d'esquema XML (obsolet ara).
 - No segueix la sintaxi XML.
 - XML Schema Definition, XSD:
 - Segueix la sintaxi XML (permet els espais de noms).
 - Pot expressar normes més complexes que DTD.