

# DNS

translator : NAME → @IPaddr

[WorldWide DataBase]



NAME : rogent.ac.upc.edu  
 mode domain-name



La información referente a un NAME

está compuesta por

+1 RR

Resource Records (RR)

DNS DB is a tree

Domains:

- generic
- country
- infrastructure

|          |                    |                               |
|----------|--------------------|-------------------------------|
| TLD      | edu                | com                           |
| 2nd LD   | upc                | microsoft                     |
| :        | :                  | :                             |
| Nth LD   | fib                | sites                         |
| modename | raco               | windows                       |
|          | ↓                  | ↓                             |
|          | (raco.fib.upc.edu) | (windows.sites.microsoft.com) |

NameServers (NS) → access → DNS DB

may hold [permanent/cached RRs]

Cada dominio tiene una authority :

primary NS  
backup NS

Tiene toda la información de una zona :

- name + @ of all nodes
- name + @ of all subzone authorities.

\*NOTE

subdomain ≈ zone  
delegated subdomain ≈ subzone

## The Resolver

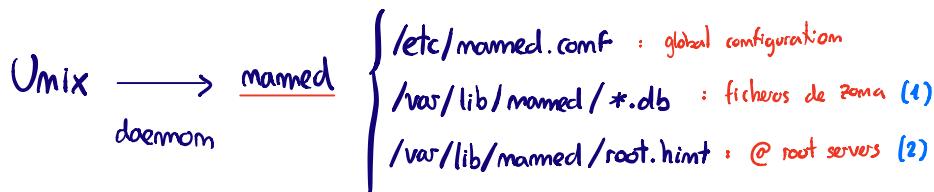
app → calls resolver

↳ · looks /etc/hosts

· contacts a NS from /etc/resolv.conf

## DNS protocol

- client-server paradigm
- UDP/TCP (si los mensajes son cortos se usa UDP)
- port 53



DNS – Unix example: zone file

1) The domain name  
The domain NS  
The domain maintainer (the @ is written as a '.')

configuration  
domain mail server  
IP addresses and alias names

comments  
server IN NS dns.foo.org.  
www IN CNAME server  
ftp IN CNAME server  
news IN A 198.133.219.26  
mail IN A 198.133.219.36  
dns1 IN A 198.133.219.50  
dns2 IN A 198.133.219.50  
dns3 IN A 10.18.8.24  
sub.foo.org. IN NS dns3.sub.foo.org.

MX preference value (used if multiple servers are available)  
IN MX 10 dns.foo.org.  
IN MX 10 mail.foo.org.  
IN A 198.133.219.10  
IN A 198.133.219.36  
IN A 198.133.219.50  
IN A 10.18.8.24

Resource Records (RR)

name (type A or CNAME), domain (type NS or MX). If the domain is missing, it is automatically added. class: IN: Internet System. type: SOA: Start of Authority, NS or CNAME... NS: NS name exchange. 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.

DNS – Unix example: root servers addresses

root.hint

2) This file holds the information on 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 /domain/named.root file on server FTP.INTERNIC.NET -OR- RS.INTERNIC.NET

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

Resource Records (RR) pointing to root-servers

address of a name  
NS name

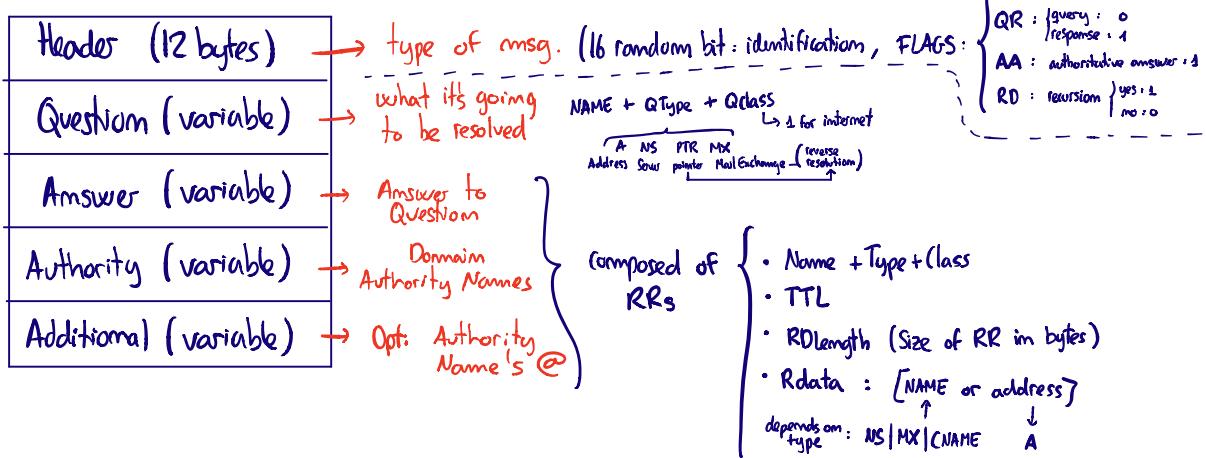
Los Root-Server son la puerta de entrada a una búsqueda. Almacenan los TLD y por lo tanto son los primeros que se deben consultar.

los Ns cachean las resoluciones. Si se solicta una entrada previamente cacheada se entrega sin preguntar a su authority.

NAME → could be associated to → SEVERAL IP@

why?  
(para que no lo haga todo un servidor)  
↓  
MIRRORING

## DNS : Message Format



## TCPDUMP DNS

IP source > IP:53 : id num. [+] A?  
dns (16 bit) ↑  
recursion ms. num. met.  
name to resolve

## CHARSETS

ASCII → 7 bits 64 chars

ISO - 8859 - 1 (latin-1) añade acentuaciones, etc; a ASCII.

UTF-8

first 128 = ASCII

0 ← x(7)  
110 ← x(5)  
1110 ← x(4)  
11110 ← x(3)

(ASCII / 1 byte)

(2 bytes)

(3 bytes)

(4 bytes)

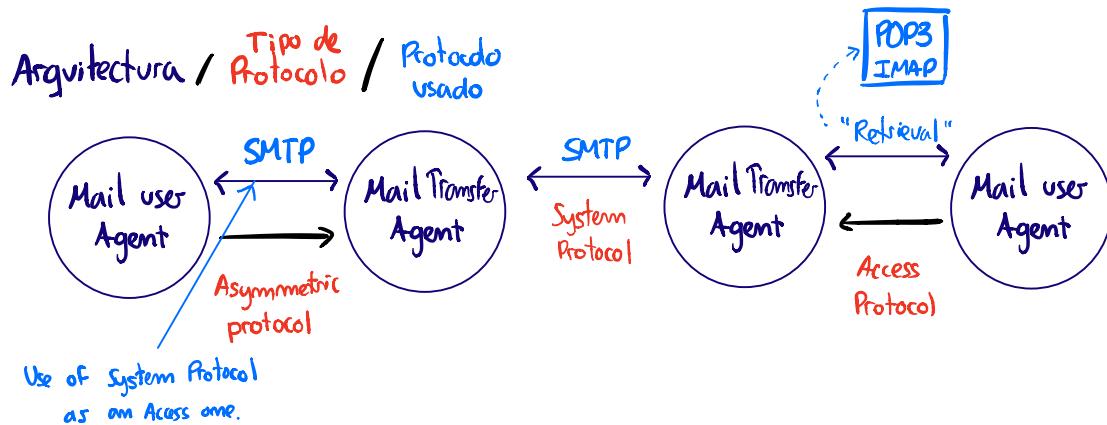
caracteres adicionales (2,3,4 Bytes) → empiezan por  
10 xx xxxx

€ = U+20AC = 10 0000 1010 1100 (16bit)

Unicode = 1110 0010 10 000010 10 101100  
          usa 3B

# e-mail

- TCP
- Port : 25
- App layer Protocol: SMTP (Simple Mail Transfer Protocol)
- Retrieval Protocol: IMAP, POP, HTTP



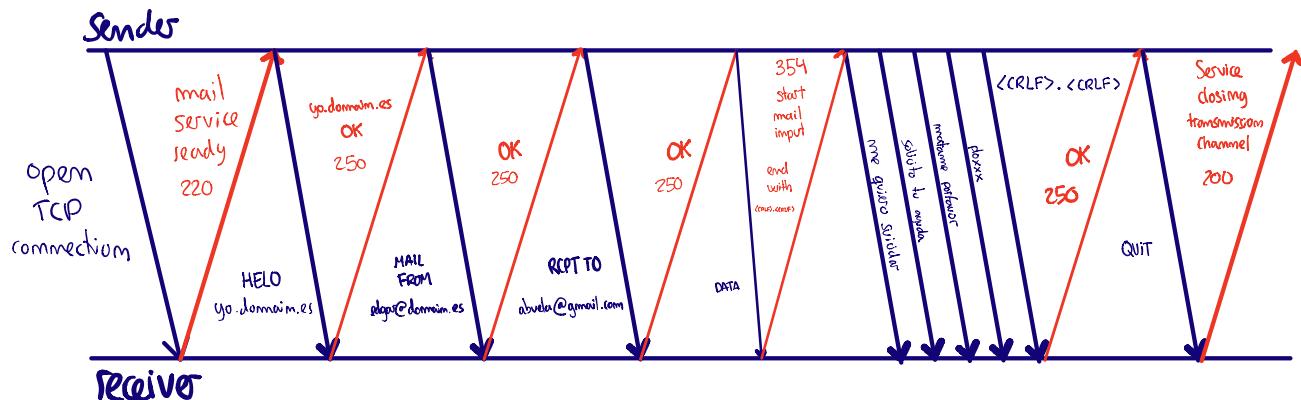
## SMTP

- text-based (ASCII)
- few commands (Client ones):
  - HELO : identifica el cliente SMTP
  - MAIL FROM : identifica mailbox del remitente
  - RCPT TO : identifica mailbox del receptor
  - DATA : mensaje
  - QUIT : close transaction

### Format:

- Header
- Empty line
- Body

- que identifica?
- server replies :  $\begin{cases} \text{3-digit number} \\ + \\ \text{message} \end{cases}$



## MIME

- Especifica como transportar "Multimedia objects" (encoding)
- Internal Format → invisible
- Se puede reenviar directamente

MIME → opciones headers

|                                             |
|---------------------------------------------|
| { content type<br>Content transfer encoding |
|---------------------------------------------|

- binary → no encoding
- base64 → fichero aumenta tamaño en 33% (printable ASCII)

## \* (CONTENT TYPE (MIME))

- Text - type / subtype

- Image

- Audio

- Application } data without associated application  
 } organization - product (PDF)

- multipart :

- mezclar : xD
- alternativa : lo mismo en diferentes formatos.
- paralelo : simultáneo (audio+video)
- digest : colección de mensajes
- Related : muchos objetos relacionados (webpage)

- Message : Mensajes completos, fragmentos o "external body": a reference to an external object.

Uso de boundary: ↗

-- UNIQUE\_BOUNDARY

content-type: text/plain

Hola que tal.

-- UNIQUE\_BOUNDARY

content-type: image

\*codificada en base64\*

-- UNIQUE\_BOUNDARY

## MAILBOX ACCESS PROTOCOLS

### Post Office Protocol (POP)

- client-server protocol (Asymmetric)
- Messages retrieved from server (local copy)

← copia local

### Internet Message Access Protocol (IMAP)

- client-server protocol (Asymmetric)
- Messages accessed and managed at the server.

← manage  
the cloud

## WEBMAIL

- front-end = website
- MUA = web browser
- access protocol = HTTP

- HTTP server uses SMTP or POP3

# WEB

- TCP • port = 80



• 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:

Protocol : HTTP  
Information: HTML  
Link : URI / URN / URL

## HTTP Request (client) [method + object + version]

### \* methods:

- GET : descargar objeto

- POST : postear objetos (pides que acepte información incluida en el body) → Se envía mediante MIME types.

\* header: give additional information about the request.

## HTTP response (server) [version + status code + text phrase]

2XX

↳ success XX .

## Web Persistence

- Non persistent: open TCP connection → send 1 object → close TCP connection

- Persistent: Maintains TCP connection opened.

!--> persistent + pipelining : cliente va pidiendo más información.

## Caching and Proxies

- Caching: The client stores downloaded pages in local cache.

↳ conditional : comprueba {  
    GET                          Date  
                                  +  
                                  Etag}

- Proxy Server: intermediary for requests from clients.

↳ Advantages:

- Security
- Logs
- Caching
- Save Public IP addresses

## Web based APPs

- Components:

- Presentation: web browser (client)
- Engine: generating on the fly HTML pages (server)
  - Languages: Java / PHP / Python / ...
- Storage: a database (SQL)

- Benefits:

- Fast to deploy + upgrade (only in server)
- Only a browser is needed.
- Cross-platform compatibility.

## HTML & XML

- element
- attribute
- text

HTML accepts

- Forms : user inputs
- Scripting : built-in programs
- CSS : modulo separado (apariencias)

Mejorar tratamiento de los datos:    HTML → XML

### XML:

- Datos estructurados
- tree structure
- Es navegable → XPath