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**

These slides are based on the set of slides provided by Llorenç Cerdà, Leandro Navarro and Jaime Delgado for this course. They include some modifications and some new slides.

Outline

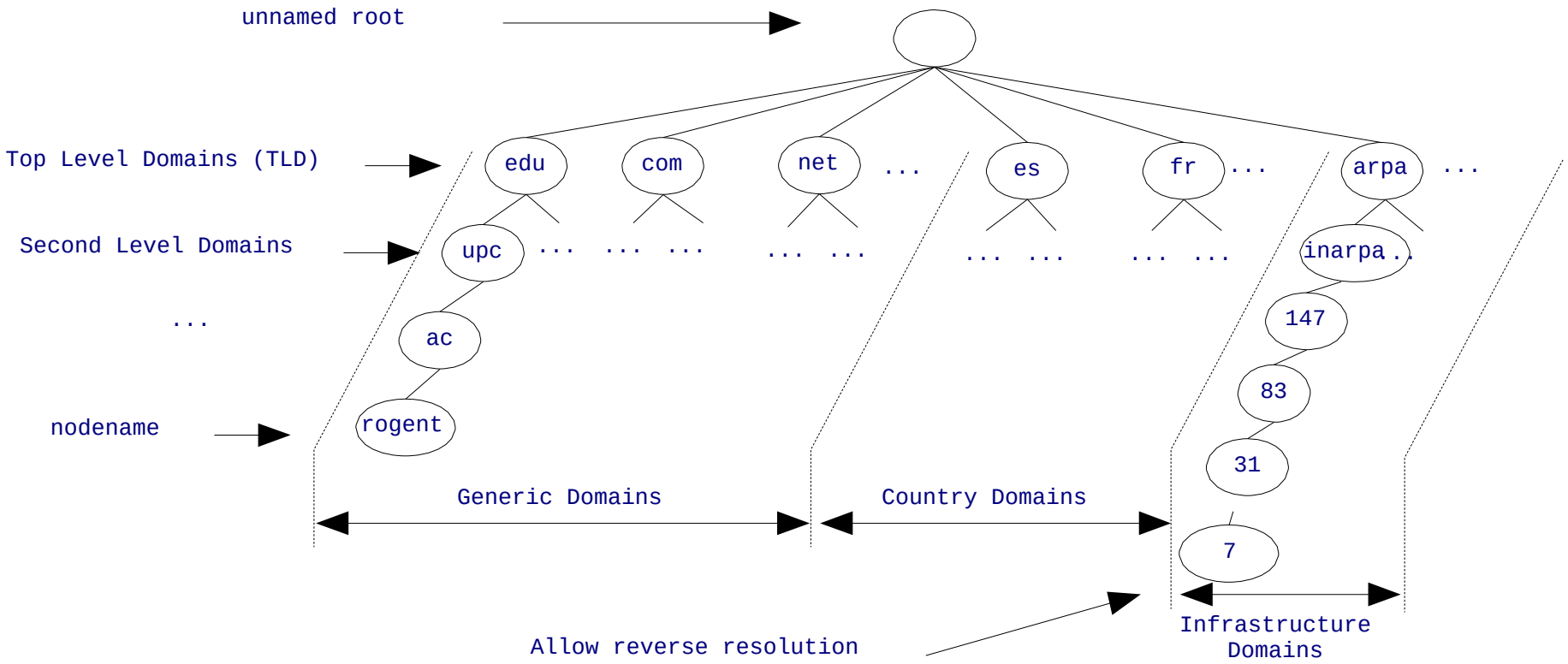
- **DNS**
- Charsets
- Email (SMTP)
- Web (HTTP)
- HTML

Domain Name System DNS (RFC 1034, 1035, Y1987)

- 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-mane: 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).

DNS – Domain Hierarchy

DNS data base is organized in a tree:



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.

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.

DNS - Unix example: The resolver

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

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

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

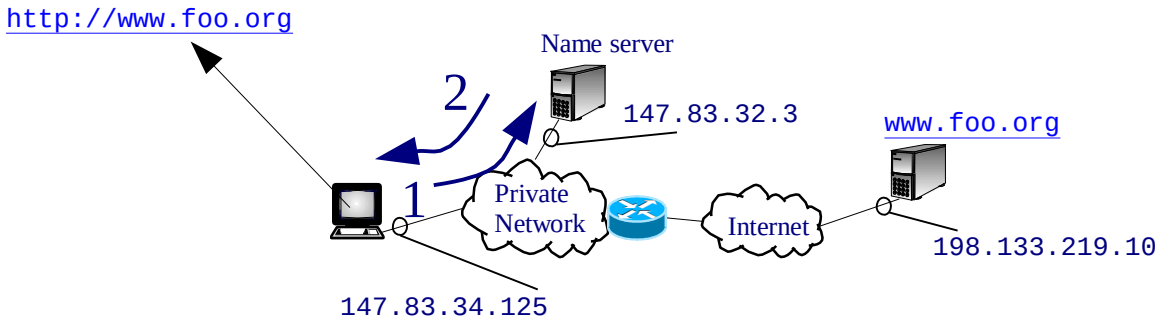
```
# hosts         This file describes a number of hostnametoaddress
#               mappings for the TCP/IP subsystem.  It is mostly
#               used at boot time, when no name servers are running.
#               On small systems, this file can be used instead of a
#               "named" name server.
# Syntax:
# IPAddress   FullQualifiedHostname    ShortHostname
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
```

DNS - Protocol

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



1 `18:36:00.322370 IP (proto: UDP) 147.83.34.125.1333 > 147.83.32.3.53: 53040+ A? www.foo.org. (31)`

2 `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)`

DNS – Unix example: Basic NS configuration

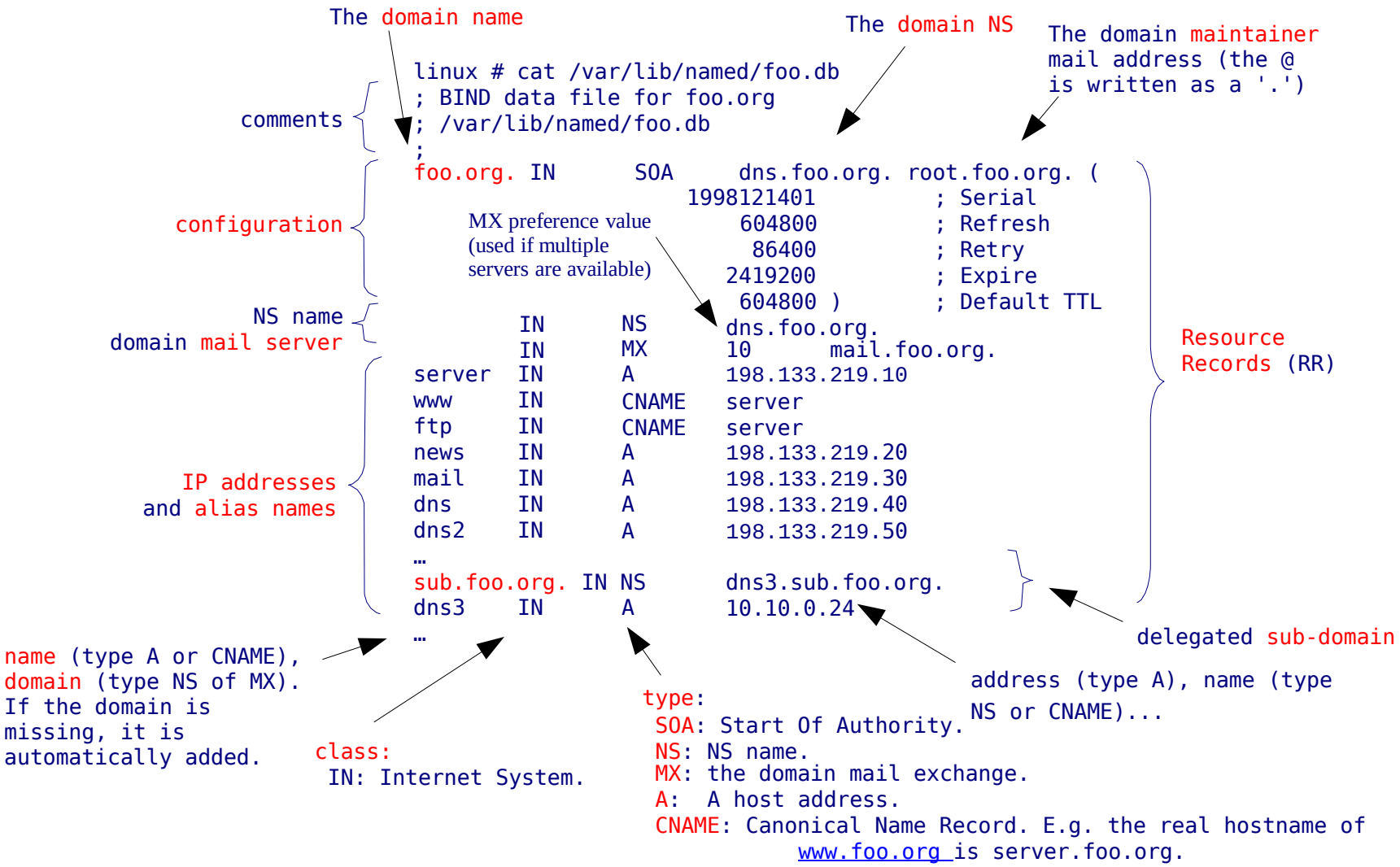
- Unix NS implementation is BIND (Berkeley Internet Name Domain), http://www.isc.org.

- 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

DNS – Unix example: zone file



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

...

.                        3600000  IN  NS    M.ROOT-SERVERS.NET.
M.ROOT-SERVERS.NET.      3600000  IN  A     202.12.27.33
```

comments

Resource Records (RR)
pointing to root-servers

address of a name
NS name

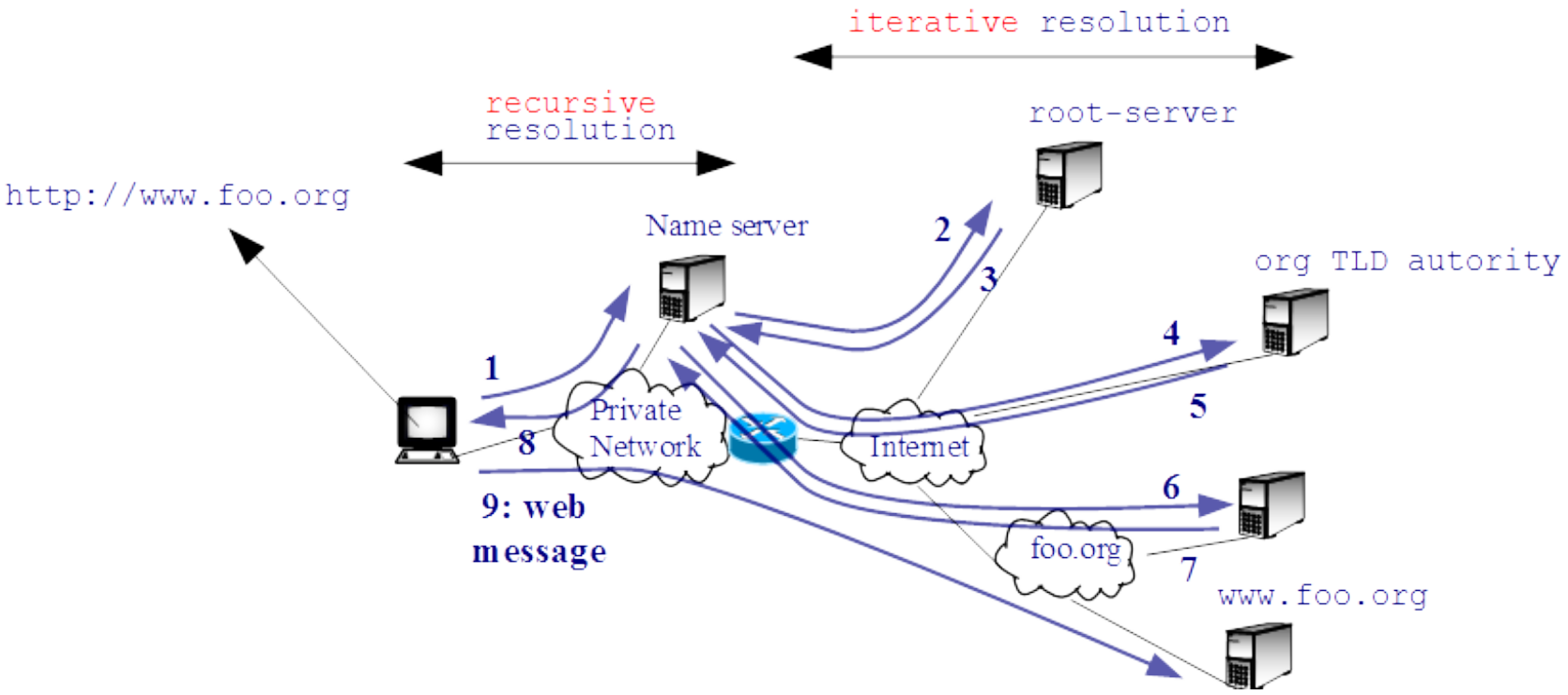
DNS – Data Base Organization

- 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
- Root server addresses are needed in a NS configuration.

Source: http://www.root-servers.org

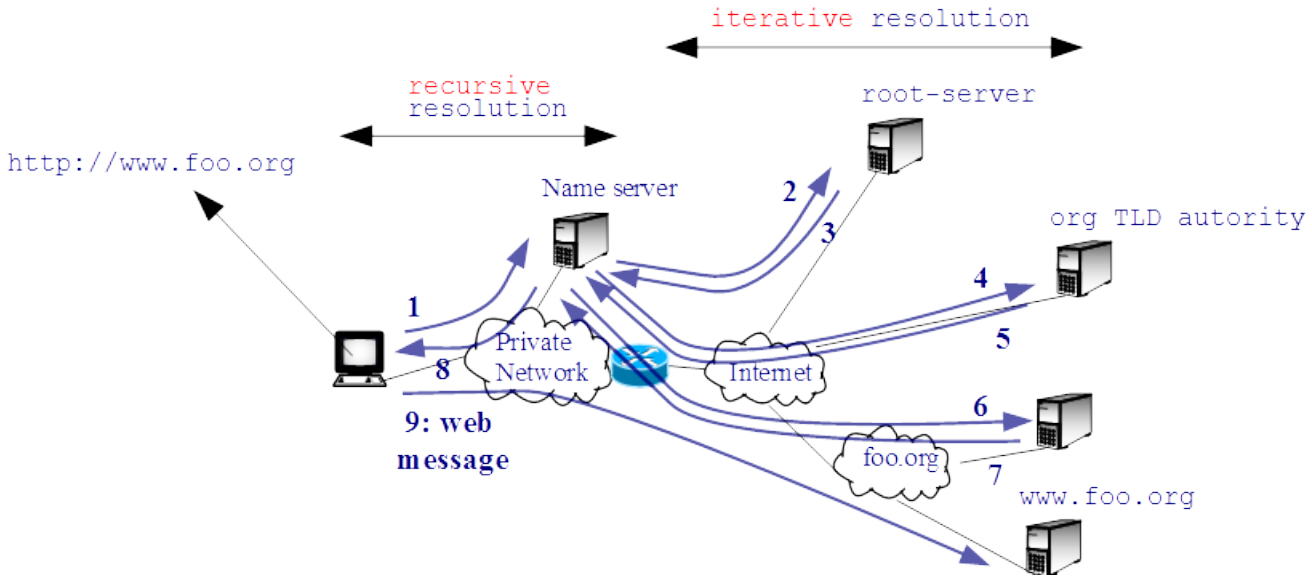
DNS – Resolution

- NSs cache name resolutions.
- A cached RR is returned without looking for in the NS authority.

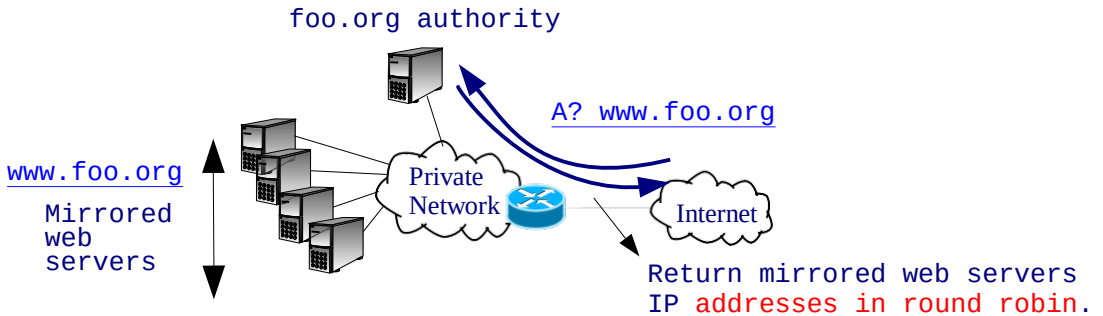


DNS – Resolution

- 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).



DNS – Load balancing, example

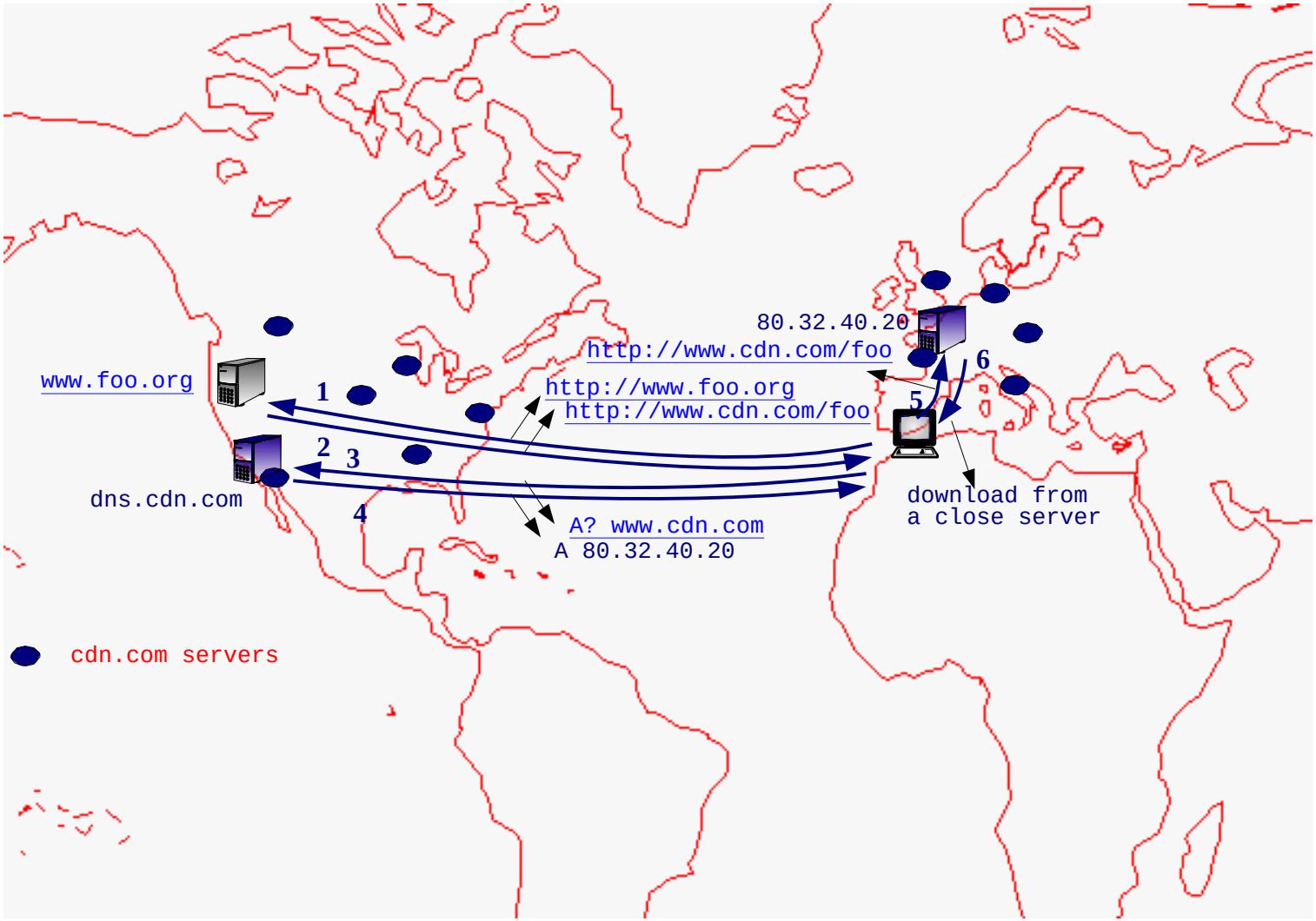


- Example using dig:

```
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:
www.microsoft.com.      3135    IN      CNAME   toggle.www.ms.akadns.net.
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       207.46.20.60
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       207.46.225.60

;; Query time: 42 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Sun Mar 11 10:48:11 2007
;; MSG SIZE  rcvd: 203
```

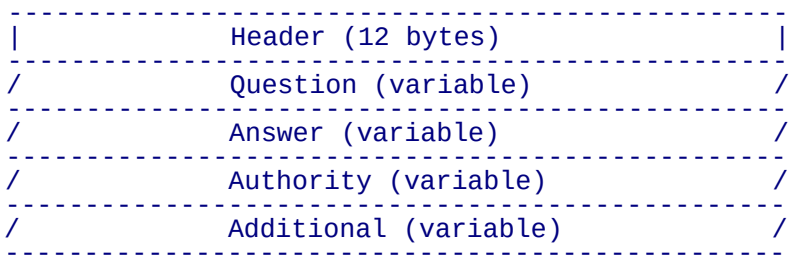
```
linux ~> dig www.microsoft.com

; <<>> 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:
www.microsoft.com.      3469    IN      CNAME   toggle.www.ms.akadns.net.
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

;; Query time: 43 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Sun Mar 11 10:42:38 2007
;; MSG SIZE  rcvd: 203
```

DNS - Content Distribution Networks, example



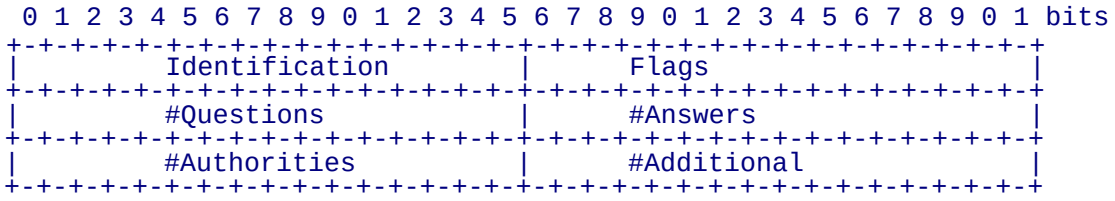
DNS – Messages: Message Format

- 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.



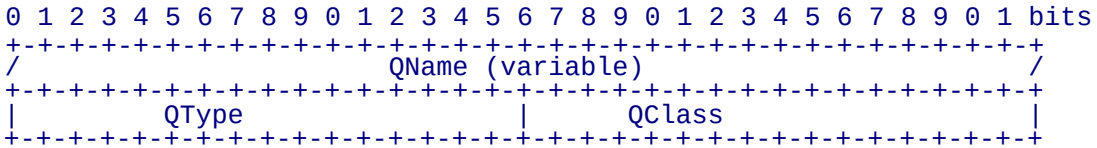
DNS – Messages: Header

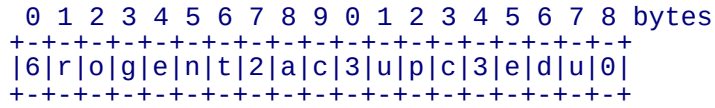
- Identification: 16 random bits used to match query/response
- Flags. Some of them:
  - Query-Response, QR: 0 for query, 1 for response.
  - Authoritative Answer, AA: When set, indicates an authoritative answer.
  - Recursion Desired, RD: When set, indicates that recursion is desired.
- The other fields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



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.

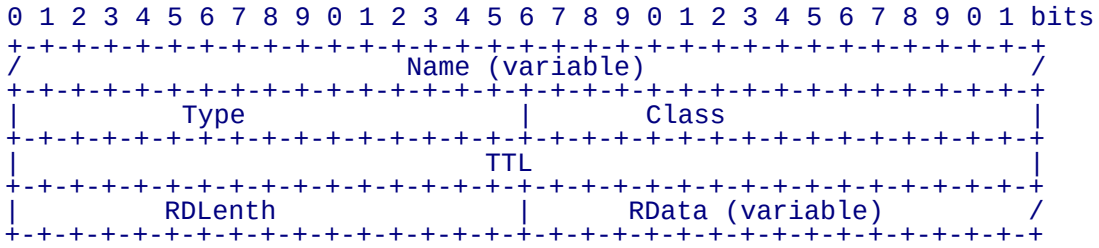




Codification example of `rogent.ac.upc.edu`

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'.



DNS – Messages: Example

```
# tcpdump s1500 vvpni eth0 port 53
tcpdump: listening on eth0, linktype 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.**
- **+: RecursionDesired 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).**

Outline

- DNS
- **Charsets**
- Email (SMTP)
- Web (HTTP)
- HTML

Languages, cultures, alphabets

7400 million people (2016)

22% speak Chinese, 11% English, 7% Spanish, 0,1% Catalan

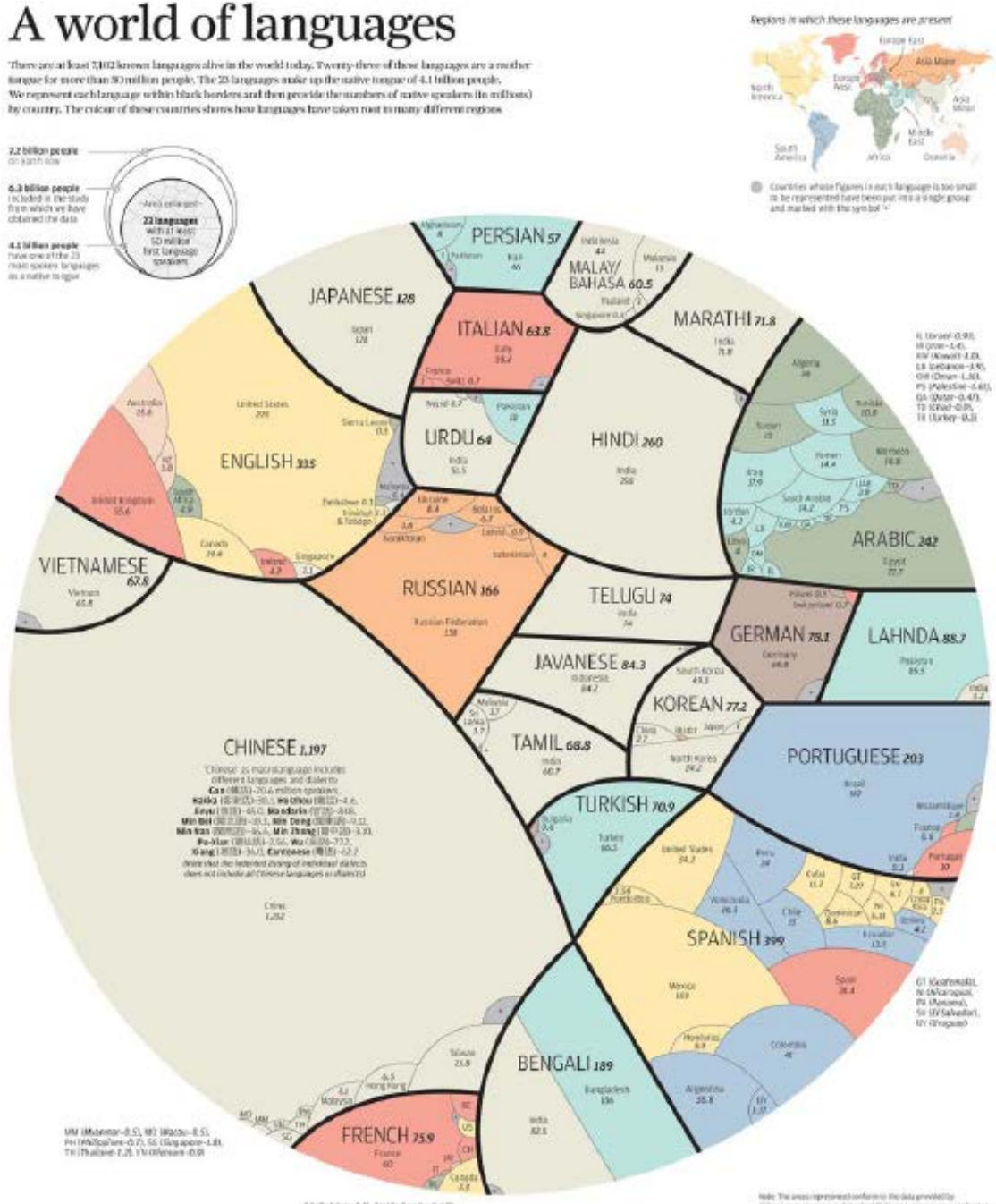
Apart from languages, there are cultures and alphabets

- 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


A world of languages
There are at least 7,102 known languages alive in the world today. Twenty-three of those languages are a mother tongue for more than 50 million people. The 23 languages make up the native tongue of 4.1 billion people. We represent each language within black borders and then provide the numbers of native speakers (in millions) by country. The colour of these countries shows how languages have taken root in many different regions.
Regions in which these languages are present
7.2 billion people on earth now
6.3 billion people included in the study from which we have obtained the data
4.1 billion people have one of the 23 most spoken languages as a native tongue
Area enlarged
23 languages with at least 50 million first language speakers
PERSIAN 57
MALAY/ BAHASA 60.5
JAPANESE 128
ITALIAN 63.8
MARATHI 71.8
URDU 64
HINDI 260
ENGLISH 335
ARABIC 242
VIETNAMESE 67.8
RUSSIAN 166
TELUGU 74
GERMAN 78.1
LAHNDA 88.7
JAVANESE 84.3
KOREAN 77.2
CHINESE 1,197
'Chinese' as macrolanguage includes different languages and dialects
TAMIL 68.8
PORTUGUESE 203
TURKISH 70.9
SPANISH 399
BENGALI 189
FRENCH 75.9

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
. . .

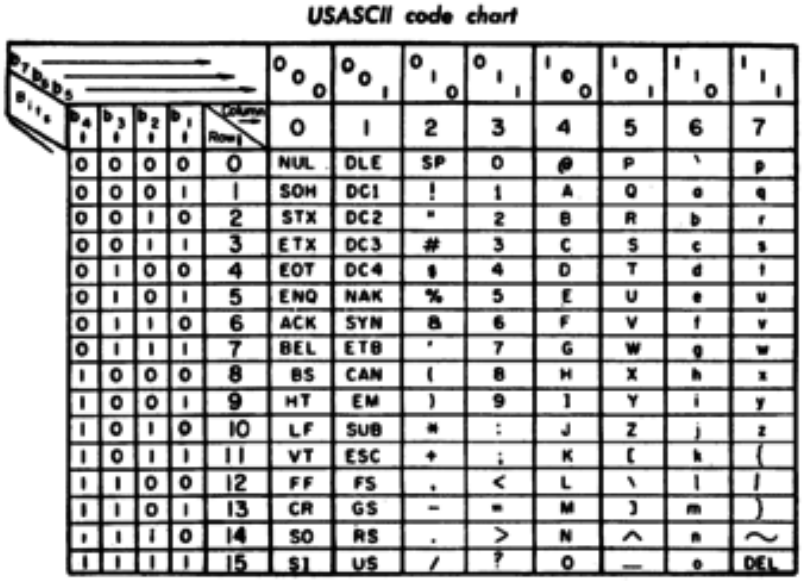
English is the default …

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)

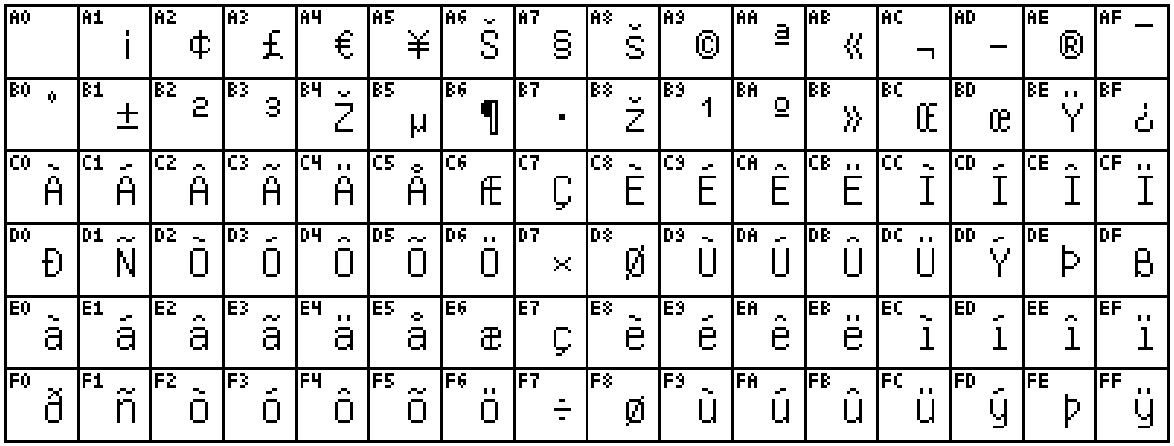


USASCII code chart

Bits b7 b6 b5 →	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
b4 b3 b2 b1 ↓ Row / Column →	0	1	2	3	4	5	6	7
0 0 0 0	0	NUL	DLE	SP	0	@	P	`	p
0 0 0 1	1	SOH	DC1	!	1	A	Q	a	q
0 0 1 0	2	STX	DC2	"	2	B	R	b	r
0 0 1 1	3	ETX	DC3	#	3	C	S	c	s
0 1 0 0	4	EOT	DC4	$	4	D	T	d	t
0 1 0 1	5	ENQ	NAK	%	5	E	U	e	u
0 1 1 0	6	ACK	SYN	&	6	F	V	f	v
0 1 1 1	7	BEL	ETB	'	7	G	W	g	w
1 0 0 0	8	BS	CAN	(	8	H	X	h	x
1 0 0 1	9	HT	EM	)	9	I	Y	i	y
1 0 1 0	10	LF	SUB	*	:	J	Z	j	z
1 0 1 1	11	VT	ESC	+	;	K	[	k	{
1 1 0 0	12	FF	FS	,	<	L	\	l	|
1 1 0 1	13	CR	GS	-	=	M	]	m	}
1 1 1 0	14	SO	RS	.	>	N	^	n	~
1 1 1 1	15	SI	US	/	?	O	_	o	DEL

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 ISO,

- Unicode 9.0.0 (7/2016): 128,172 symbols  
- U+hex code: U+0020 = ' ' (*code points* of 21 bits)

Character Encodings: Universal Transformation Format (UTF)

- Difficulty or impossibility to transport 8 o 16 bits data in Internet protocols:
- UTF-7, **UTF-8**, UTF-16, UTF-32 (variable length)

  UTF-8 uses 1..4 "*code units*" (bytes)

http://www.unicode.org http://unicode-table.com

Universal Coded Character Set Unicode

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

Char. number range (hexadecimal)	UTF-8 octet sequence (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

  - 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

UTF-8

Unicode (or *Universal Coded Character Set*) *Transformation Format – 8-bit*

This table shows UTF-8 as it is since 2003 (the `x` characters are replaced by the bits of the code point):

UTF-8 (2003)

Number of bytes	Bits for code point	First code point	Last code point	Byte 1	Byte 2	Byte 3	Byte 4
1	7	U+0000	U+007F	0xxxxxxx			
2	11	U+0080	U+07FF	110xxxxx	10xxxxxx		
3	16	U+0800	U+FFFF	1110xxxx	10xxxxxx	10xxxxxx	
4	21	U+10000	U+10FFFF	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx

Character		Octal code point	Binary code point	Binary UTF-8	Octal UTF-8	Hexadecimal UTF-8
$	U+0024	044	010 0100	00100100	044	24
¢	U+00A2	0242	000 1010 0010	11000010 10100010	302 242	C2 A2
€	U+20AC	020254	0010 0000 1010 1100	11100010 10000010 10101100	342 202 254	E2 82 AC
𐍈	U+10348	0201510	0 0001 0000 0011 0100 1000	11110000 10010000 10001101 10001000	360 220 215 210	F0 90 8D 88

Source: Wiquipedia

Variable length encodings

- UTF-8 (8 bits) (rfc2044)

```
ContentType:  text/plain; charset=UTF8
ContentTransferEncoding:  8bit
CatalÃ , FranÃ§ais, TÃ¤mÃ¤ on testi.
```

- UTF-7 (7 bits) (smtp …)

```
ContentType:  text/plain; charset=UTF7
ContentTransferEncoding:  7bit                    on testi.
Catal+AOA,   Fran+AOcais,   T+AOQm+AOQ
```

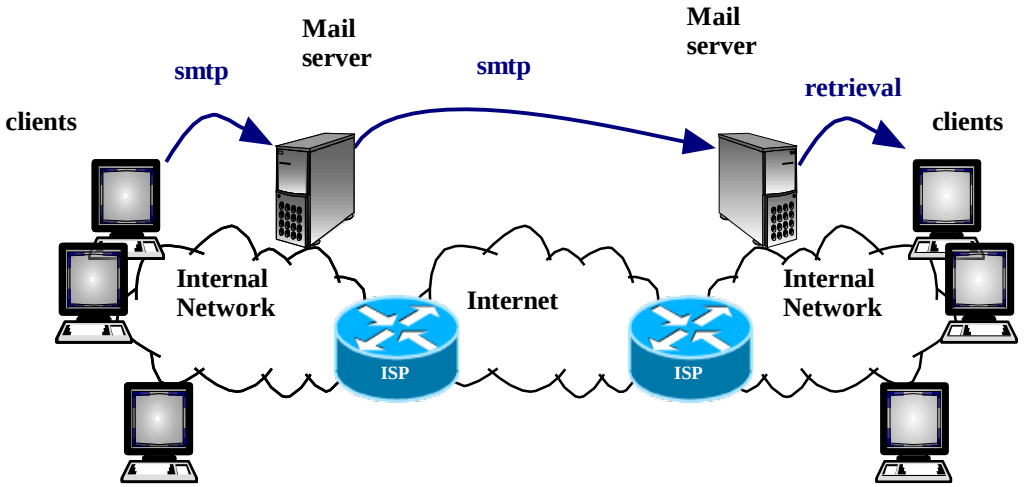
https://www.charset.org/utf8-to-latin-converter

Outline

- DNS
- Charsets
- **Email (SMTP)**
- Web (HTTP)
- HTML

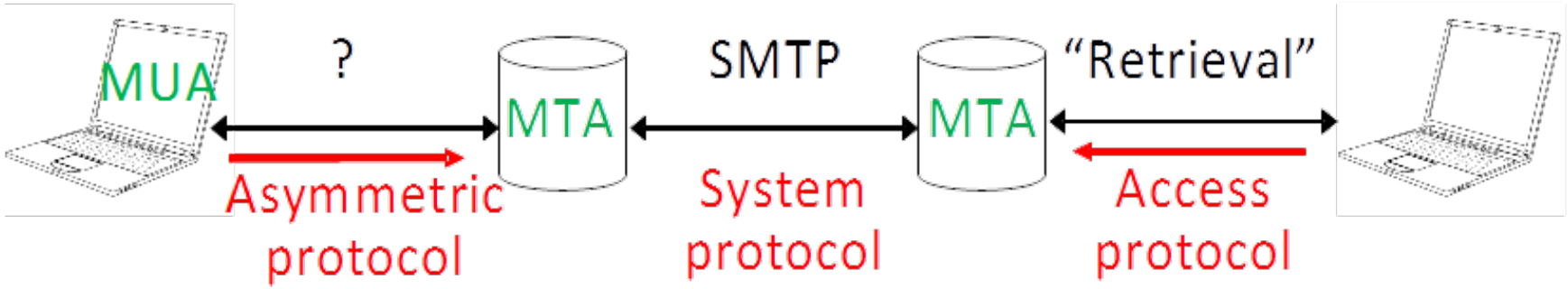
Email

- 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 (Y 1982) and last updated by RFC-5321 (Y 2008).
  - Retrieval protocols (IMAP, POP, HTTP).



Unit 5. Network applications

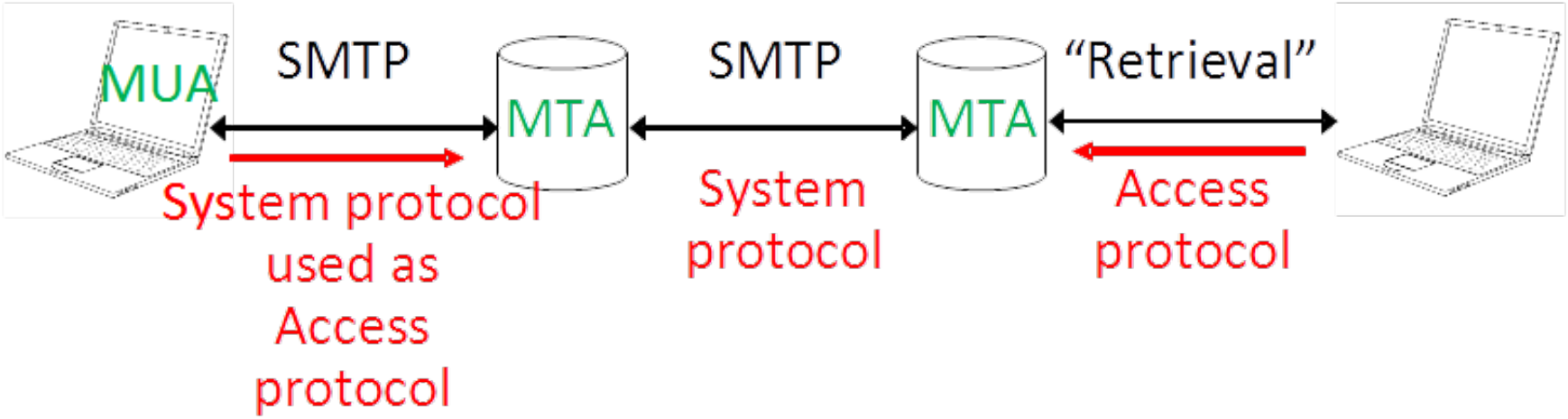
Email - Architecture



- MUA: Mail User Agent
- MTA: Mail Transfer Agent
- *SMTP: Simple Mail Transfer Protocol*

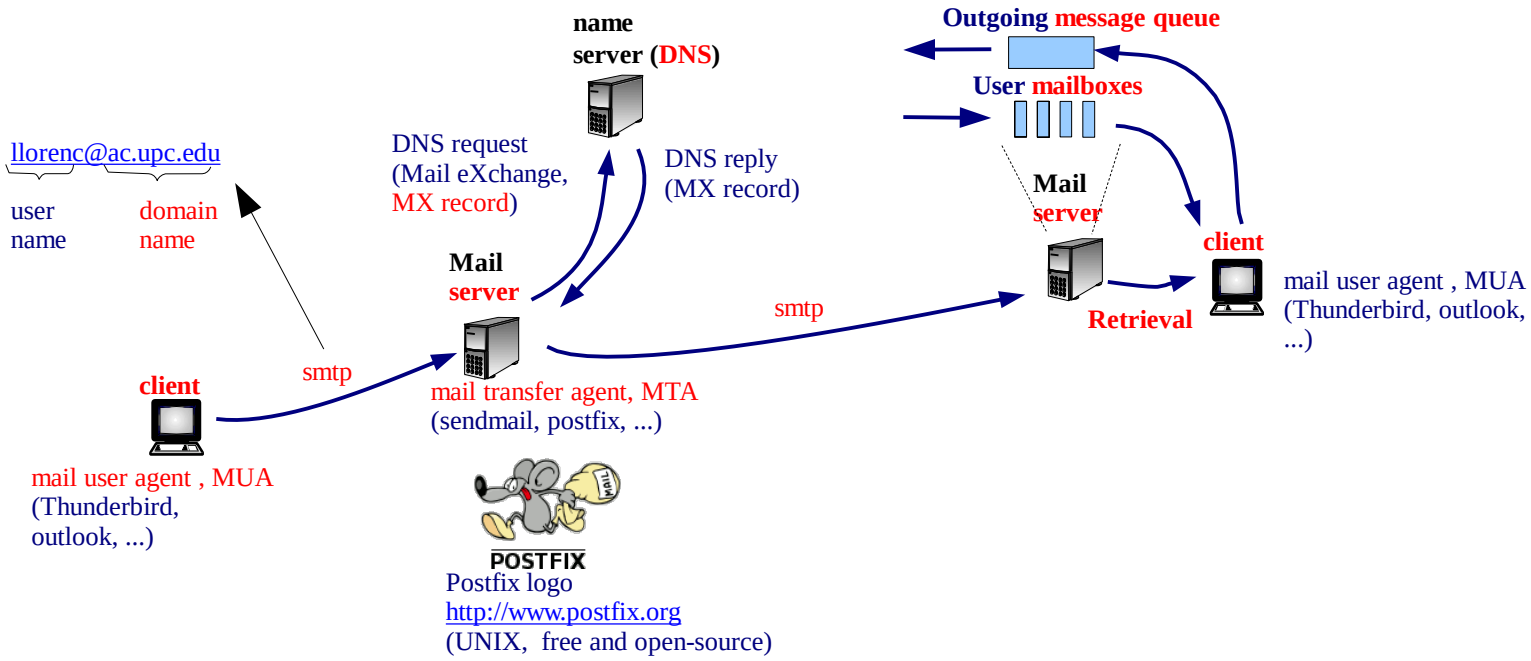
Unit 5. Network applications

Email - Protocols



- "Retrieval" protocols (mailbox access):
  - POP3 *(Post Office Protocol)*
  - IMAP *(Internet Message Access Protocol)*
- SMTP: Simple Mail Transfer Protocol

Email - SMTP processing model



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)

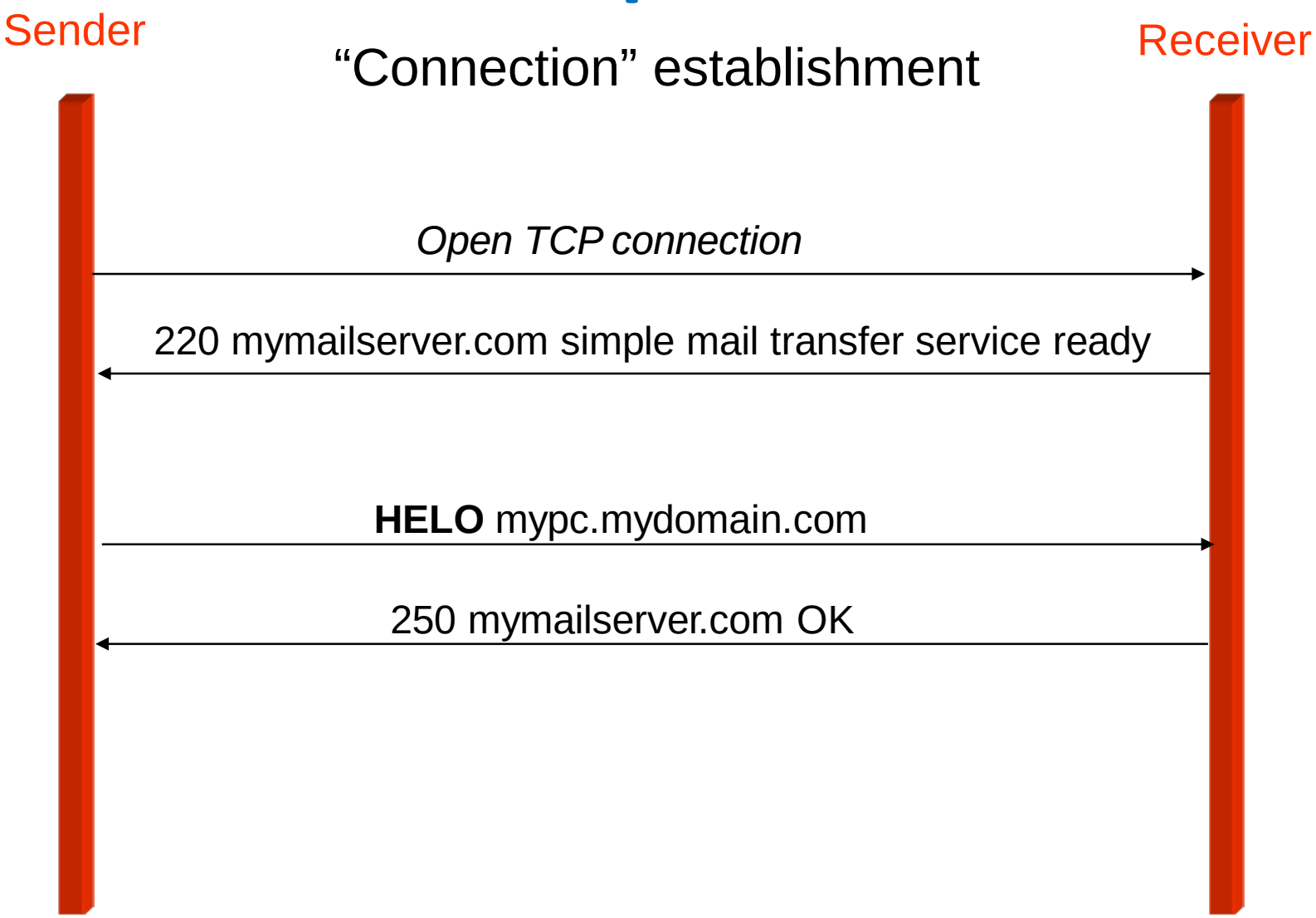
    RCPT TO: (identify recipient mailbox)

    DATA (mail message)

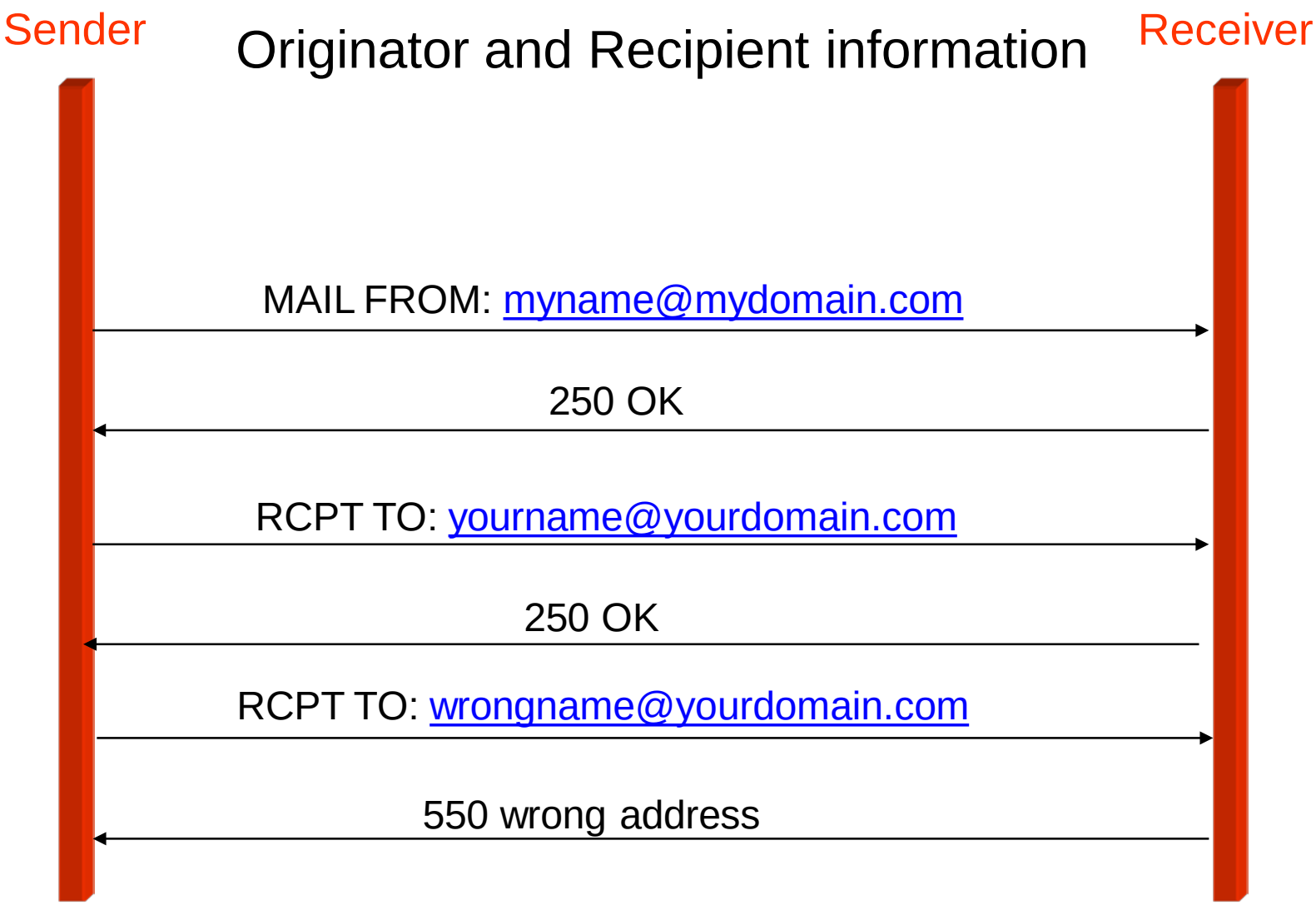
    QUIT (close transaction)

  - Server replies: Three digit number (identify what state the client to enter next), and a human understandable message.

SMTP protocol



SMTP protocol

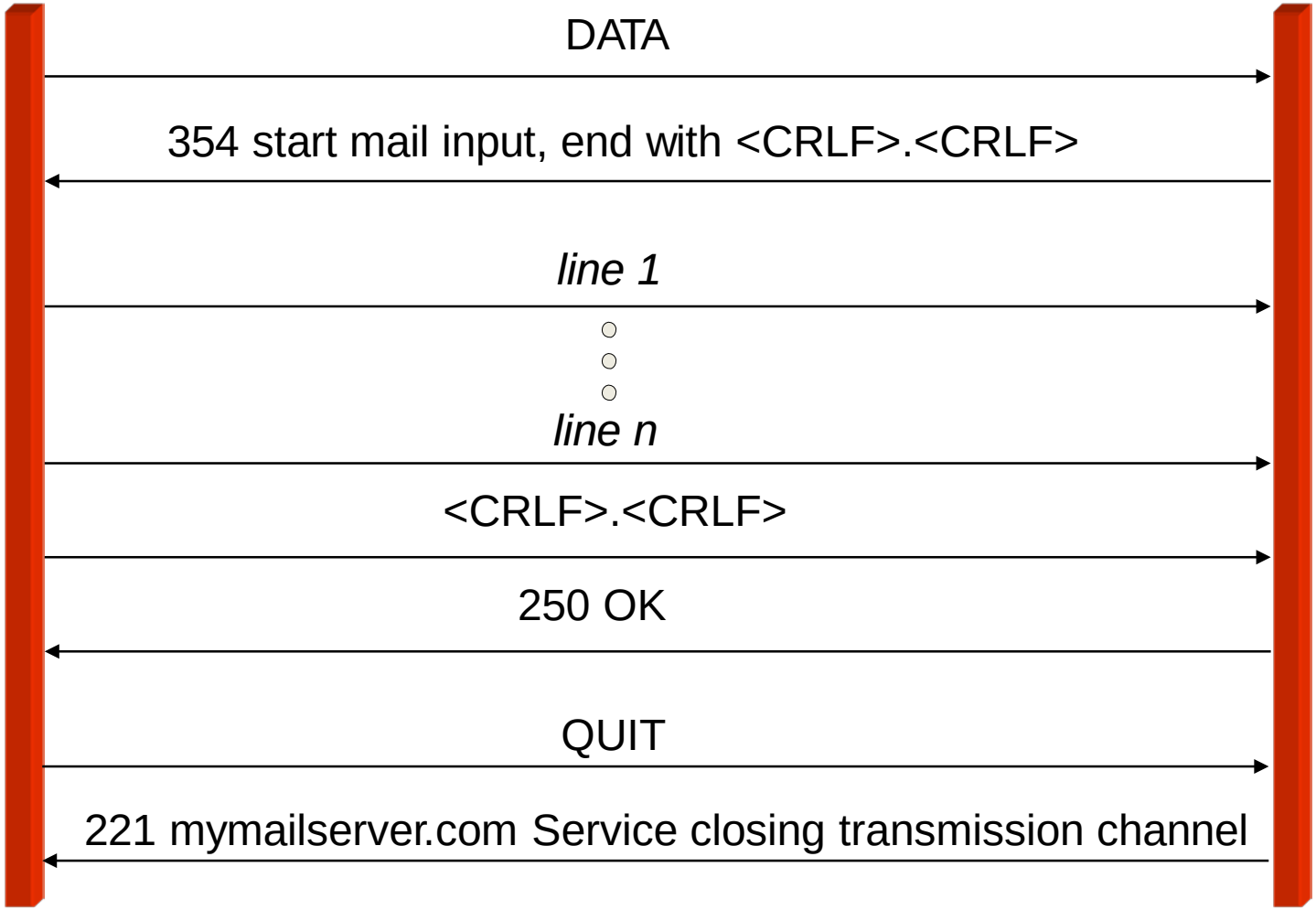


SMTP protocol

Sender

Receiver

Message transmission and Close



Email – message formats

- Format described in RFC-822 (update RFC-5322) Internet Message Format
- Example (extracted from the RFC):

```
From: John Doe <jdoe@machine.example>
To: Mary Smith <mary@example.net>
Subject: Saying Hello
Date: Fri, 21 Nov 1997 09:55:06 0600
MessageID: <1234@local.machine.example>


This is a message just to say hello. So,
"Hello".
```

Header: gives information about the message. Fields defined in RFC-5322

Empty line

Body

Email - SMTP protocol (RFC-5321, originally RFC-821)

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 '^]'.
```

SMTP transaction

SERVER COMMANDS

```
220 dash.upc.es ESMTP Sendmail 8.14.1/8.13.1; Fri, 4 Feb 2011 14:57:15 +0100
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.
linux ~>
```

Encrypted SMTP: port 465

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.
```

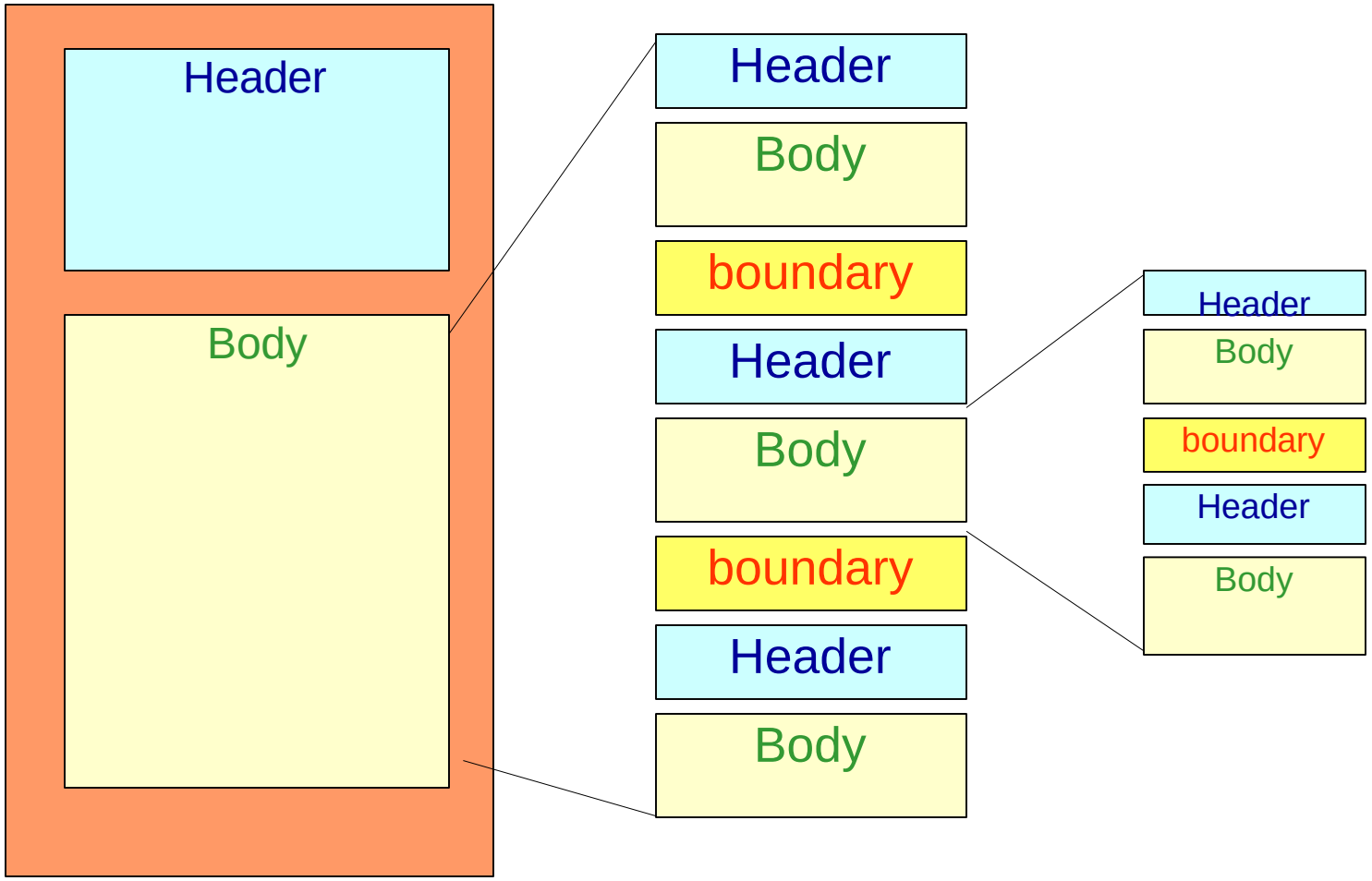
```
...Subject: French mail
Content-type: text/plain; charset=iso-8859-1
Content-transfer-encoding: quoted-printable

Le courrier =E9lectronique =E0 la fran=E7aise ...
```

```
...Content-type: image/gif
Content-Transfer-Encoding: Base64

R0lGODdhSgGgAfUAAENDQ01NTTw8PEVF...
```

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

5WVlZ6enqqqqr....
--CUT_HERE
Content-type: text/plain

Wasn't that neat?
--CUT_HERE--
```

MIME multipart message



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 objects 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

Registration scheme Type/subtype: mantained by IANA

MIME content types

- `Content-Type` element structure:
  - **`type/subtype`**

- Examples of type/subtype:
  - application/pdf, application/msword, application/soap+xml, application/vnd.ms-powerpoint, application/vnd.nokia.radio-preset, ...
  - audio/GSM, audio/mpeg, audio/vnd.dolby.mps, ...
  - image/gif, image/jpeg, image/png, image/vnd.adobe.photoshop, ...
  - text/plain, text/html, text/vnd.dvb.subtitle, ...
  - message/rfc822, message/http, ...
  - model/iges, ...
  - multipart/mixed, multipart/alternative, ...
  - video/H264, video/mp4, video/vnd.nokia.videovoip, ...

MIME: content transfer encoding

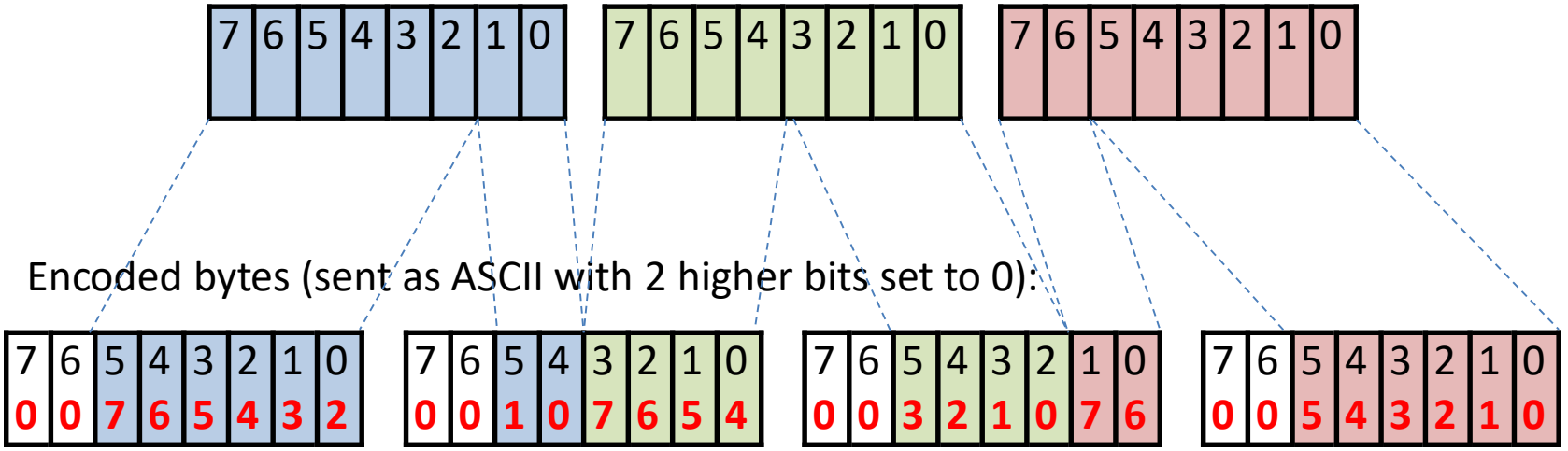
Ways to encode content: (to “get through” a 7 bit transport)

- Quoted-Printable:
  - The majority of text is 7 bits, transform some characters €→ =E4
  - The result “almost" legible without decoding. Depends on the 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 (exemple):

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

Base64 encoding

Bytes to transmit (8 bits either 0 or 1):



Only ASCII values from 0 to 63
(**64** posible values)

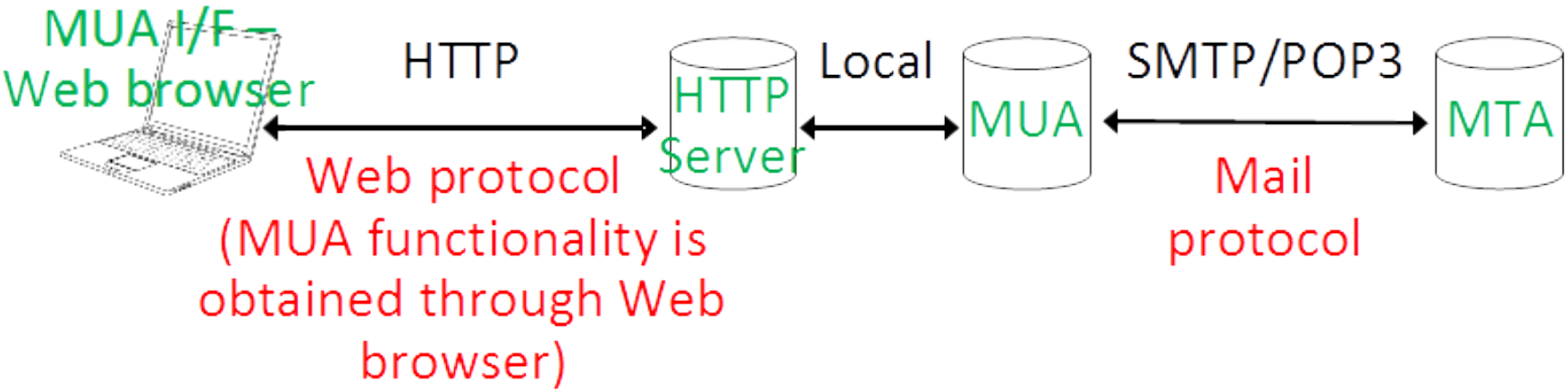
Inefficiency: 4 bytes transmitted for every 3 bytes!

Mailbox Access protocols

- Post Office Protocol (POP) version 3 (POP3)
  - RFC 1939 (1996)
  - Client-server protocol (Asymmetric)
  - Messages retrieved from the mail server (copied locally).
- Internet Message Access Protocol (IMAP)
  - RFC 3501 (2003). 1st version 4 in RFC 1730 (1994). 1st RFC (version 2) in 1988 (RFC 1064).
  - Client-server protocol (Asymmetric)
  - Messages accessed and managed (folders, …) at the server

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

Email - retrieval protocols

- Post Office Protocol (POP), RFC-1939:
  - POP server listens on well-known port 110
  - POPS port 995
  - User normally deletes messages upon retrieval
- Internet Message Access Protocol (IMAP) RFC-3501:
  - IMAP server listens on well-known port 143 (IMAPS port 993)
  - 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.

CORREO ENTRANTE

IMAP

Nombre de usuario / email:
antonio.garcia@tudominio.ext
Contraseña: la que escogiste durante la activación
Puerta: 143
Nombre del Servidor: pop.tudominio.ext

POP3

Nombre de usuario / email:
antonio.garcia@tudominio.ext
Contraseña: la que escogiste durante la activación
Puerta: 110
Nombre del Servidor: pop.tudominio.ext

CORREO SALIENTE

SMTP (correo electrónico)

Nombre de usuario / email:
antonio.garcia@tudominio.ext
Contraseña: la que escogiste durante la activación
Puerta: 25
Nombre del Servidor: authsmtp.tudominio.ext

SMTP (dominio)

Nombre de usuario / email: smtp@tudominio.ext
Contraseña: la que escogiste durante la activación
Puerta: 25
Nombre del Servidor: authsmtp.tudominio.ext

No SSL  SSL ¿Qué es SSL?

CORREO ENTRANTE

IMAP

Nombre de usuario / email:
antonio.garcia@tudominio.ext
Contraseña: la que escogiste durante la activación
Puerta: 993
Nombre del Servidor: pop.securemail.pro

POP3

Nombre de usuario / email:
antonio.garcia@tudominio.ext
Contraseña: la que escogiste durante la activación
Puerta: 995
Nombre del Servidor: pop.securemail.pro

CORREO SALIENTE

SMTP (correo electrónico)

Nombre de usuario / email:
antonio.garcia@tudominio.ext
Contraseña: la que escogiste durante la activación
Puerta: 465
Nombre del Servidor: authsmtp.securemail.pro

SMTP (dominio)

Nombre de usuario / email: smtp@tudominio.ext
Contraseña: la que escogiste durante la activación
Puerta: 465
Nombre del Servidor: authsmtp.securemail.pro

No SSL  SSL ¿Qué es SSL?

SSL (Secure Sockets Layer) es un protocolo para transmitir información de manera segura.

Baixada POP:
Més informació

1. Estat: POP està inhabilitat
- Activa POP per a **tots els missatges**
- Activa POP als **missatges que arribin a partir d'ara**

2. Quan s'accedeix als missatges a través de POP conserva la còpia de Gmail a la Safata d'entrada

3. Configureu el vostre client de correu electrònic (per exemple, Outlook, Eudora, Netscape Mail)
Instruccions de configuració

Accés IMAP:
(accedeix a Gmail des d'altres clients amb IMAP)
Més informació

Estat: IMAP està habilitat
- Activa IMAP
- Desactiva IMAP

Quan marco un missatge a IMAP com a suprimit:
- Eliminació automàtica activada: actualitza immediatament el servidor. (predeterminat)
- Eliminació automàtica desactivada: s'espera que el client actualitzi el servidor.

Quan un missatge es marca com a suprimit i s'elimina de l'última carpeta IMAP visible:
- Arxiva el missatge (predeterminat)
- Mou el missatge a la Paperera
- Suprimeix el missatge de manera immediata i definitiva

Límits de mida de les carpetes
- No limitis el nombre de missatges en una carpeta IMAP (predeterminat)
- Limita les carpetes IMAP perquè continguin aquesta quantitat de missatges com a màxim 1.000

Configureu el vostre client de correu electrònic (per exemple, Outlook, Thunderbird, iPhone)
Instruccions de configuració

Desa els canvis | Cancel·la

Utilitzeu la taula següent per actualitzar el vostre client amb la informació correcta. Si necessiteu ajuda per actualitzar la configuració d'IMAP, cerqueu les instruccions al Centre d'ajuda del client de correu electrònic corresponent.

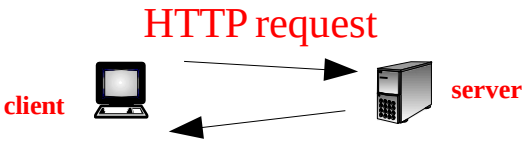
Servidor de correu entrant (IMAP)	imap.gmail.com Requereix SSL: sí Port: 993
Servidor de correu sortint (SMTP)	smtp.gmail.com Requereix SSL: sí Requereix TLS: sí (si està disponible) Requereix autenticació: sí Port per a SSL: 465 Port per a TLS/STARTTLS: 587
Nom complet o visible	El vostre nom
Nom del compte, nom d'usuari o adreça electrònica	La vostra adreça electrònica completa
Contrasenya	La vostra contrasenya de Gmail

Outline

- DNS
- Charsets
- Email (SMTP)
- **Web (HTTP)**
- HTML

Web

- World Wide Web, www: was started by Tim John Berners-Lee in 1989 and developed in the 90s to provide an easy access to information in the Internet.
- Components:
  - Transport layer: TCP, well-known port: 80.
  - Application layer protocol: HyperText Transfer Protocol (HTTP). RFC1945 (HTTP-1.0 Y1996), RFC2616 (HTTP-1.1 Y1999).
  - HyperText Markup Language (HTML): Language used to format web documents.



Web browser:
- firefox
- explorer
- ...

HTTP response (HTML page)

Web server:
- apache
- ...

Source: wikipedia

Web elements

- Protocol
  - **HTTP** (HyperText Transfer Protocol)
- Information (format)
  - **HTML** (HyperText Markup Language)
- LINK to information
  - **URI** (Uniform Resource Identifier):

    URN (Name), **URL** (Locator)

Web – links

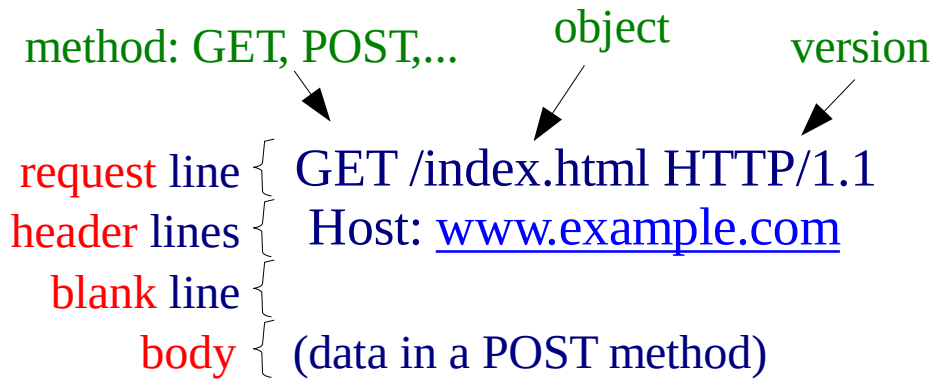
- Uniform Resource Identifier (URI) RFC3986
  - Generic syntax to identify a resource.
- Uniform Resource Locator (URL) RFC1738
  - Subset of URIs identifying the locating a resource in the Internet.
- The URL general syntax is

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://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/03879/refs9?pf_ra=ATVPD&pf_rd=07HR2

Web – HTTP Messages, RFC2616

- Client (HTTP request):



- 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

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

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

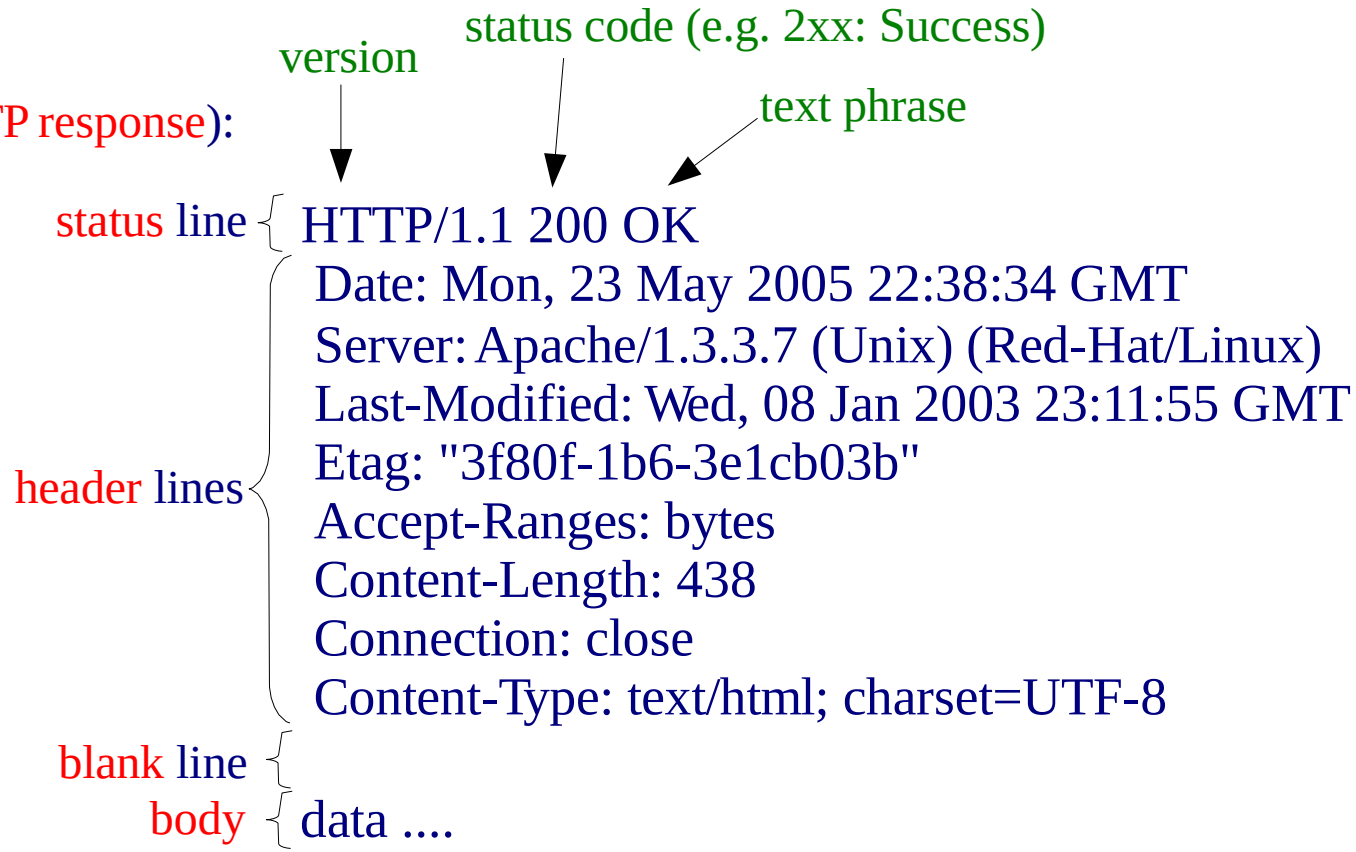
header lines {
Host: www.mysite.com
User-Agent: Mozilla/4.0
Content-Length: 27
Content-Type: application/x-www-form-urlencoded

blank line {

body { userid=llorenc&password=mypassword

Web – HTTP Messages, RFC2616

- Server (HTTP response):



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.
  - ...

Web – Persistent/non Persistent connections

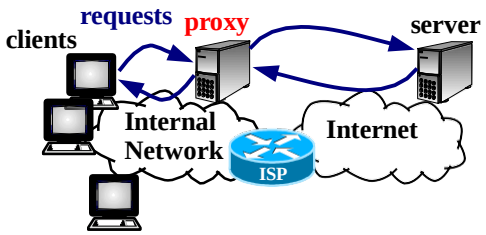
- Non persistent (default in HTTP/1.0): The server closes the TCP connection after every object. E.g, for an html page with 10 jpeg images, 11 TCP connections are sequentially opened.
- Persistent (default in HTTP/1.1) : The server maintains the TCP connection open until an inactivity time. All 11 objects 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 encounters new references, even if the objects have been not completely downloaded.

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)
    - ...



Web – web based applications

- Components:
  - Presentation: A web browser (client side).
  - Engine generating “on the fly” HTML pages (server side).
    - Languages:
      - Java.
      - Hypertext Preprocessor (PHP): Embedded program language and HTML code (http://www.php.net).
      - Other: ASP, CGI, ColdFusion, Perl, Python...
  - 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.)