

# Reti di Telecomunicazioni



Application Layer

---



# Autori



Queste slide sono state scritte da

Michele Michelotto:

[michele.michelotto@pd.infn.it](mailto:michele.michelotto@pd.infn.it)

che ne detiene i diritti a tutti gli effetti



# Copyright Notice



Queste slide possono essere copiate e distribuite gratuitamente soltanto con il consenso dell'autore e a condizione che nella copia venga specificata la proprietà intellettuale delle stesse e che copia e distribuzione non siano effettuate a fini di lucro.



# Application layer



Introduzione

Layer: Modello OSI e TCP/IP

Physics Layer

Data Link Layer

MAC sublayer

Network Layer

Transport Layer

**Application Layer**



# Domain Name System



- Difficile ricordarsi i numeri
  - Es: telnet 193.84.144.1
  - http://193.83.12.1/index.html
  - Mandami un mail a michelotto@193.85.121.5
- Se sposto il mailserver su di un'altra macchina devo cambiare anche il numero?
- DNS associa nomi ASCII a indirizzi
  - Ma la rete conosce solo i numeri per cui mi serve un meccanismo per mappare gli uni sugli altri



# resolver



- Ai tempi di Arpanet c'era un mapping statico in un file `host.txt`
  - Che c'è ancora (`/etc/hosts`) ed ha la precedenza sull DNS
- Ora un'applicazione chiama una funzione di libreria chiamata **resolver**
  - Es. `gethostbyname`
  - Il **resolver** manda una richiesta **UDP** alla porta 53 del DNS locale che restituisce il numero IP al chiamante
  - Il server risponde via **UDP** se il messaggio di risposta ha meno di 512 byte o via TCP altrimenti

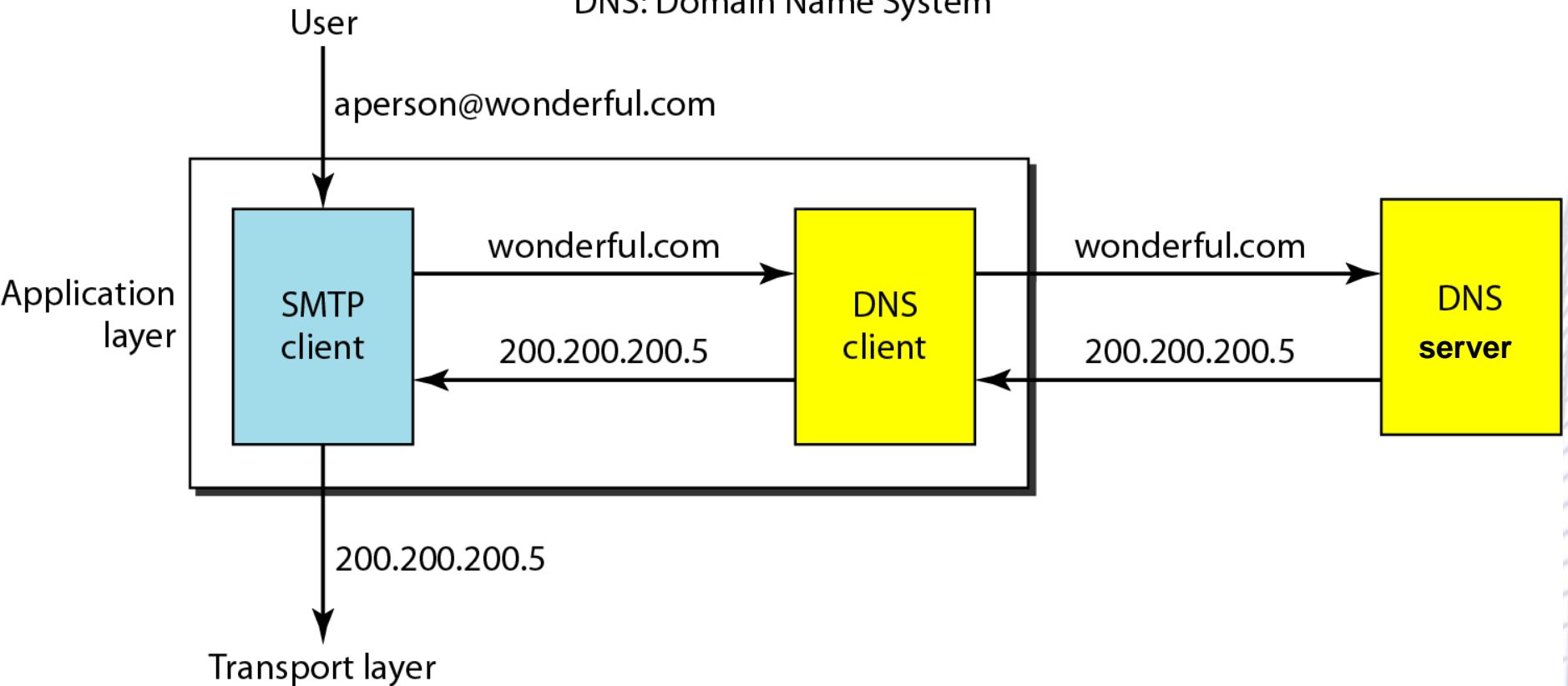


# risoluzione



SMTP: Simple Mail Transfer Protocol (e-mail)

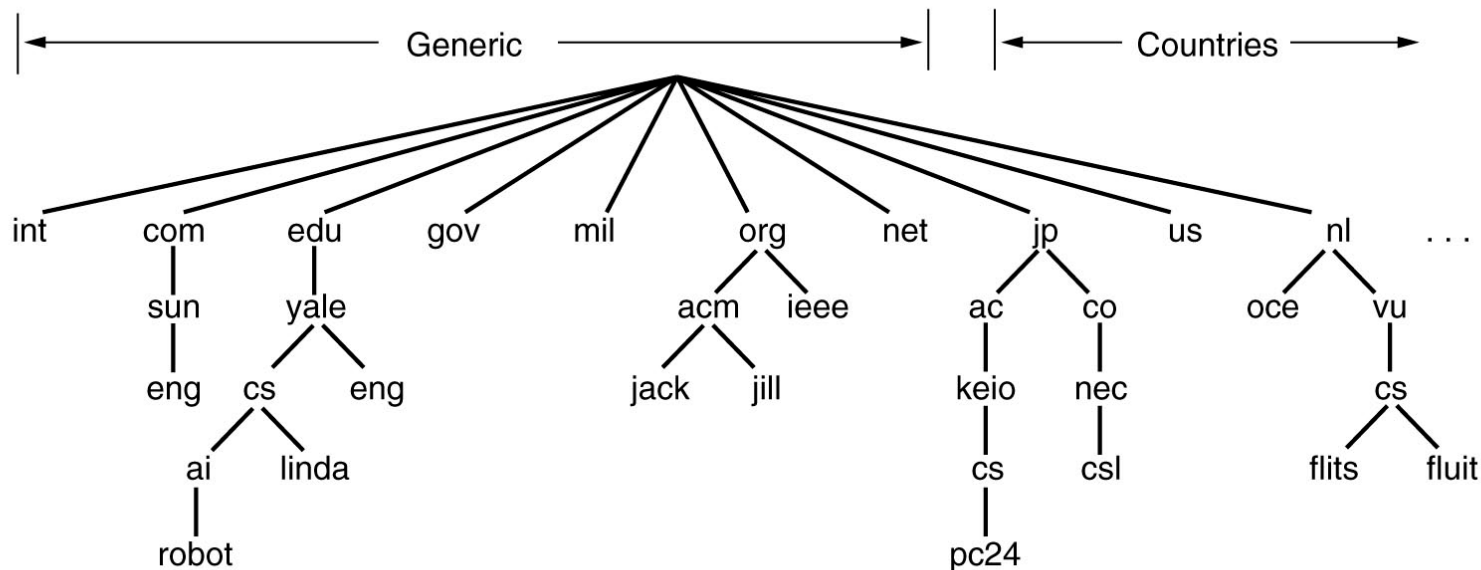
DNS: Domain Name System





# gerarchia

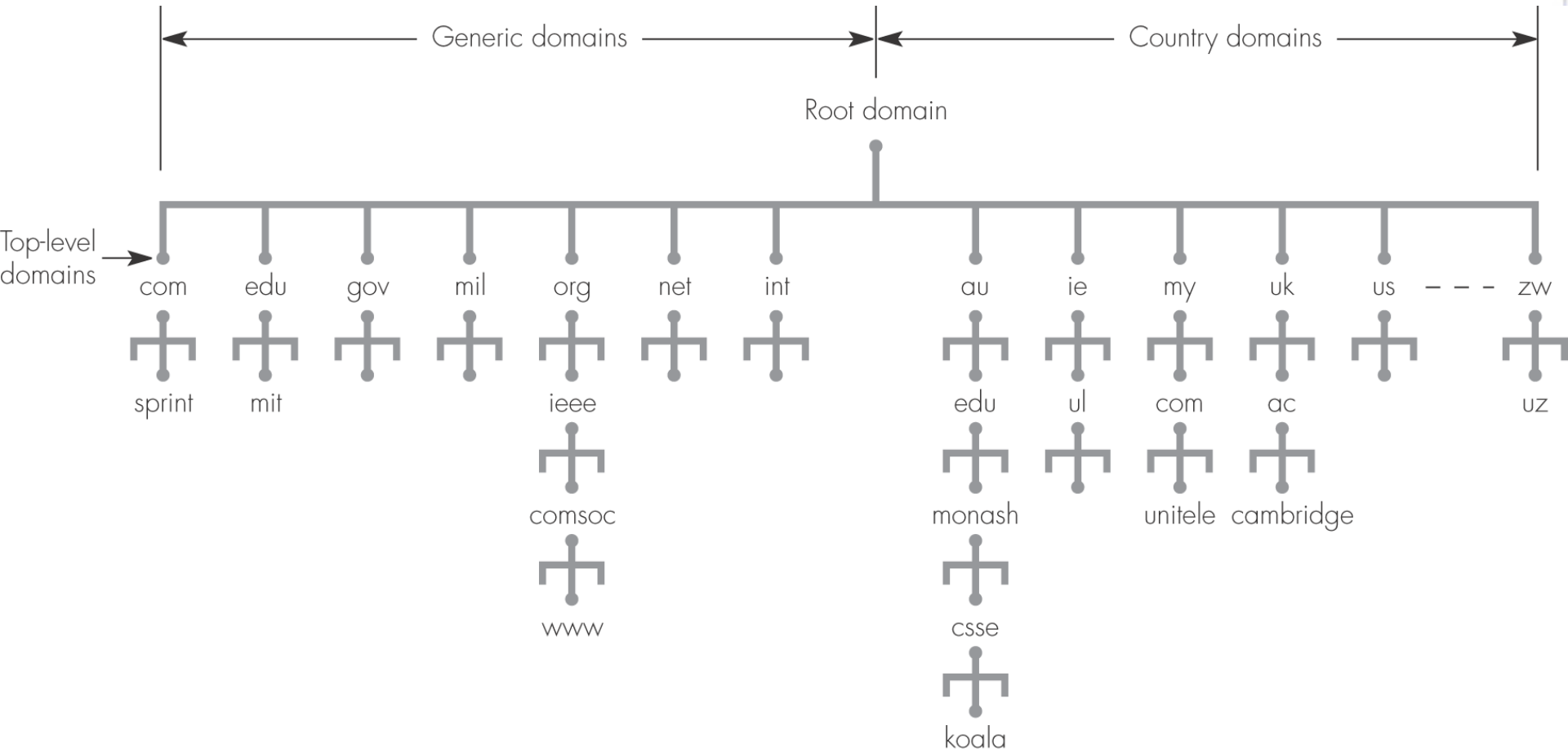
- Un server DNS può essere authoritative per il suo dominio o chiedere al server authoritative se non ha la risposta nella sua cache
- I server sono disposti in una gerarchia
- I top level sono decisi da ICANN







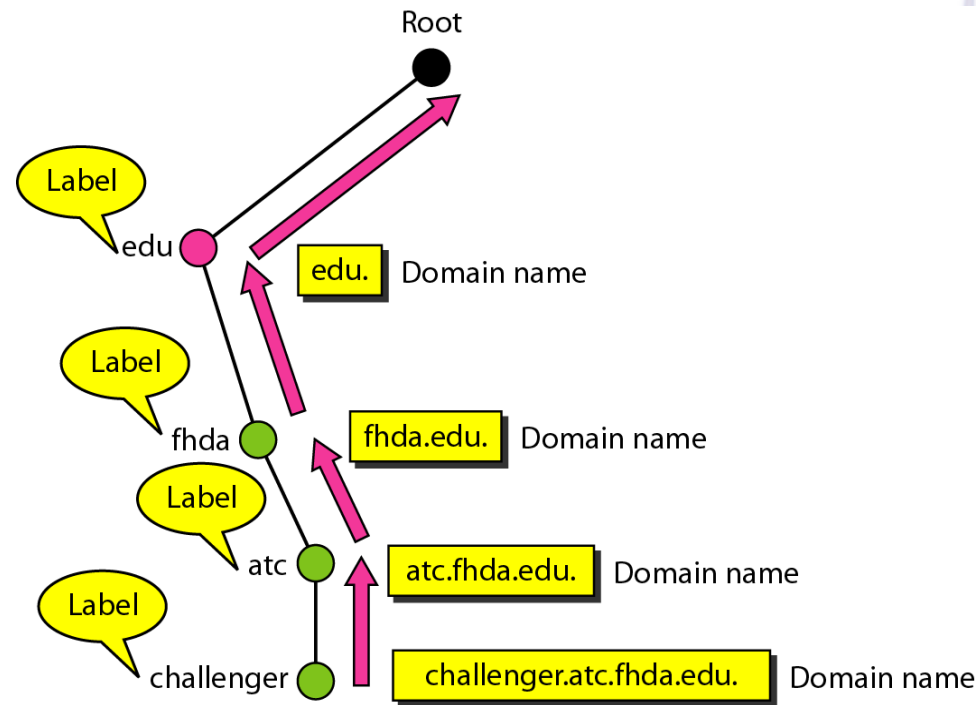
# Top level





# Top level

- I primi 7 .com (commerciale) .edu (educational) .gov (US Fed) .int (org internaz.) .mil (militari US) .net (network provider) .org (organiz. Noprofit)
- Country domain: domini nazionali .it .de .uk .ch .us .lv
- 4 nuovi domini nel 2000 .biz (business) .name (nomi di persone) .pro (professionisti) info (informazioni)
- Tutti i top level si attaccano alla root



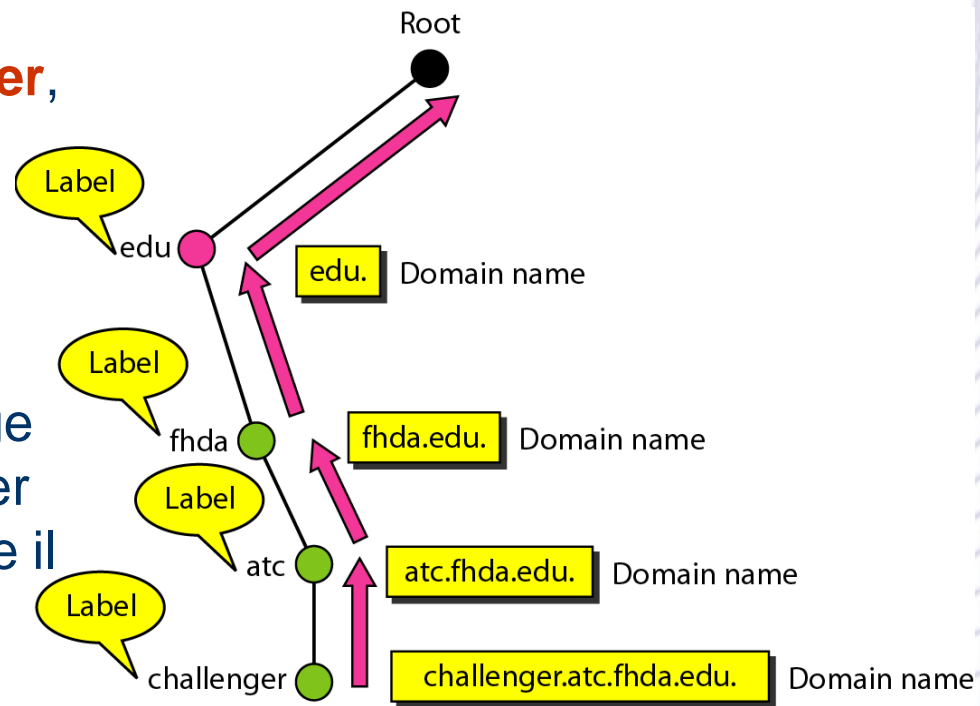


# FQDN e nomi parziali

- Il nome di dominio completo noto FQDN in questo caso è **challenger.atc.fhda.edu**.
- Posso usare anche solo **challenger**, in tal caso il client DNS prima di tradurlo completa il nome con per esempio i seguenti suffissi

**atc.fhda.edu** **fhda.edu** *null*

- *null* è la stringa vuota che aggiunge semplicemente un punto finale. Per questo motivo è possibile omettere il punto finale





# Resource record



- Ogni dominio (un host singolo o un dominio vero e proprio) ha un **resource record** associato
- Per un host c'è solo un indirizzo ma per un dominio può essere più complicato
- E' una 5-tupla
  - Domain\_name, Time\_to\_live, Class, Type, Value



# Resource record



- **Domain\_name** dice a quale dominio si riferisce il record
- **Time\_to\_live** indica quanto è stabile. Es 86400 il numero di secondi in un giorno per info stabili, 60 per info molto volatili
- **Class**: per info internet vale sempre IN
- **Type**: quale tipo di record, vedi tabella
- **Value**: un numero o nome di dominio o stringa ASCII (dipende dal tipo di record)



# Principali tipi



Type	Meaning	Value
SOA	Start of Authority	Parameters for this zone
A	IP address of a host	32-Bit integer
MX	Mail exchange	Priority, domain willing to accept e-mail
NS	Name Server	Name of a server for this domain
CNAME	Canonical name	Domain name
PTR	Pointer	Alias for an IP address
HINFO	Host description	CPU and OS in ASCII
TXT	Text	Uninterpreted ASCII text



# SOA, A ed MX



- **SOA** contiene i parametri per la zona del name server, l'indirizzo e-mail dell'amministratore, un numero seriale univoco, vari flags e timeout
- **A** (Address) contiene l'indirizzo a 32 bit di un host
  - Un host può avere diversi A, uno per ogni connessione di rete e quindi indirizzo IP. Il DNS può essere configurato per ciclare tra tutti
- **MX** indica i nomi delle macchine che accettano mail per il dominio

; Authoritative data for cs.vu.nl

cs.vu.nl.	86400	IN	SOA	star boss (952771,7200,7200,2419200,86400)
cs.vu.nl.	86400	IN	TXT	"Divisie Wiskunde en Informatica."
cs.vu.nl.	86400	IN	TXT	"Vrije Universiteit Amsterdam."
cs.vu.nl.	86400	IN	MX	1 zephyr.cs.vu.nl.
cs.vu.nl.	86400	IN	MX	2 top.cs.vu.nl.

flits.cs.vu.nl.	86400	IN	HINFO	Sun Unix
flits.cs.vu.nl.	86400	IN	A	130.37.16.112
flits.cs.vu.nl.	86400	IN	A	192.31.231.165
flits.cs.vu.nl.	86400	IN	MX	1 flits.cs.vu.nl.
flits.cs.vu.nl.	86400	IN	MX	2 zephyr.cs.vu.nl.
flits.cs.vu.nl.	86400	IN	MX	3 top.cs.vu.nl.
www.cs.vu.nl.	86400	IN	CNAME	star.cs.vu.nl
ftp.cs.vu.nl.	86400	IN	CNAME	zephyr.cs.vu.nl

rowboat		IN	A	130.37.56.201
		IN	MX	1 rowboat
		IN	MX	2 zephyr
		IN	HINFO	Sun Unix

little-sister		IN	A	130.37.62.23
		IN	HINFO	Mac MacOS

laserjet		IN	A	192.31.231.216
		IN	HINFO	"HP Laserjet III Si" Proprietary



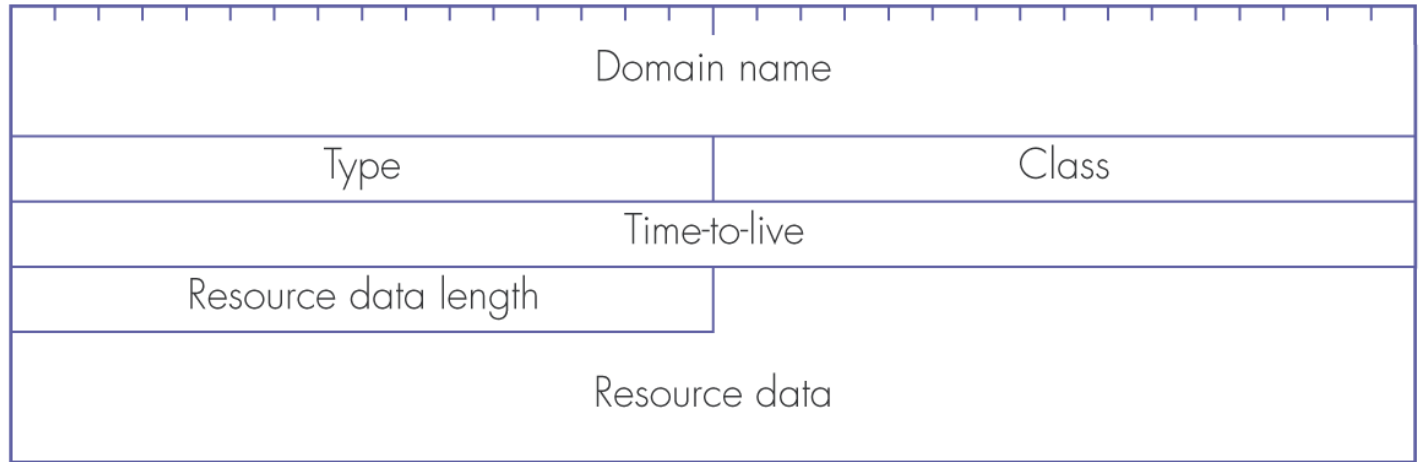


(a)

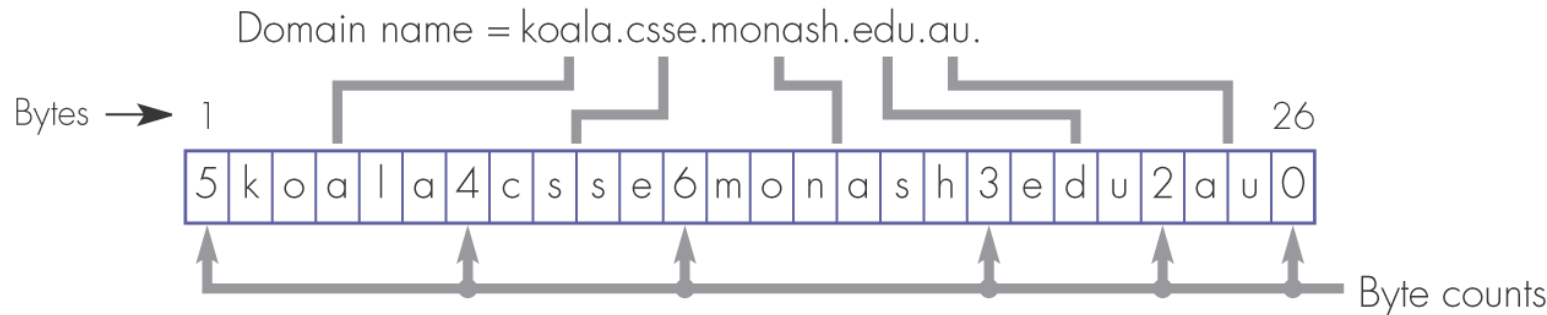
Bits → 1

16 17

32



(b)



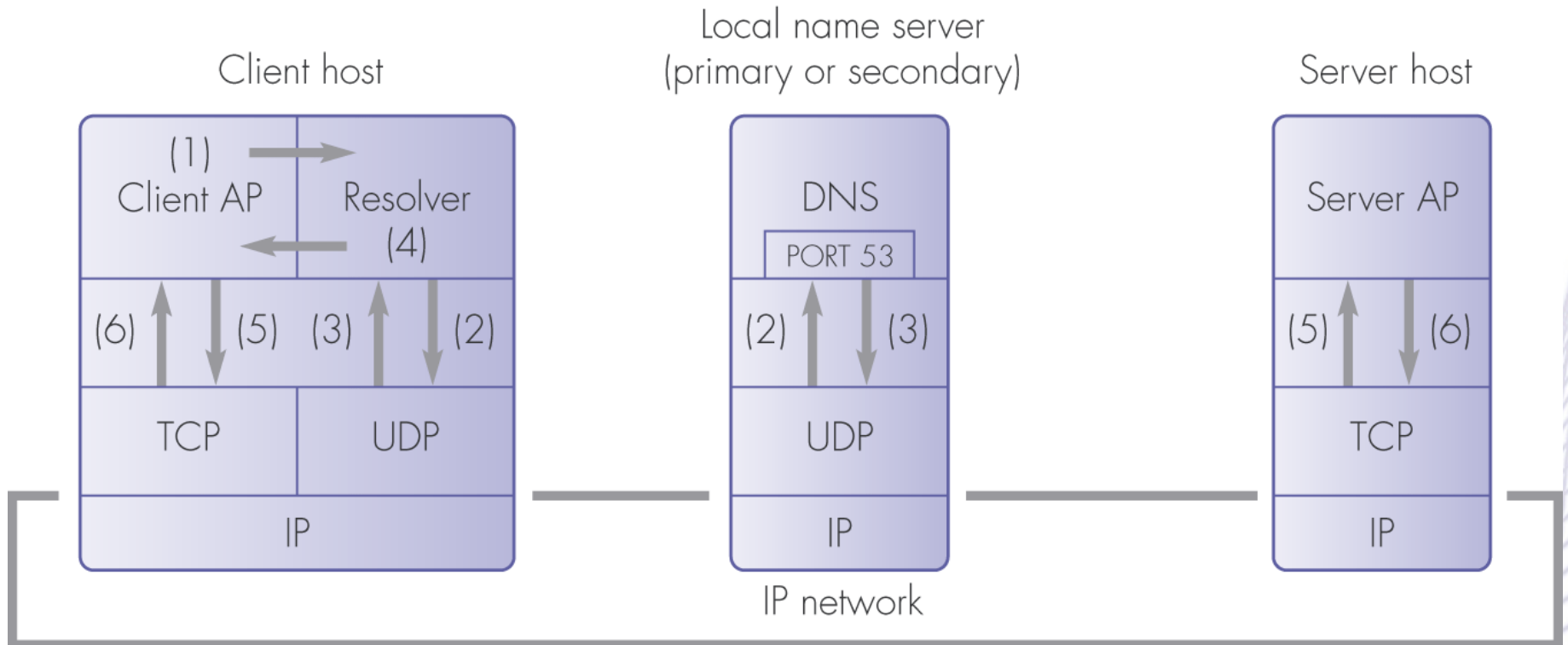
## DNS resource records and queries:

(a) resource record format;

(b) domain name format



# Risoluzione di nome locale



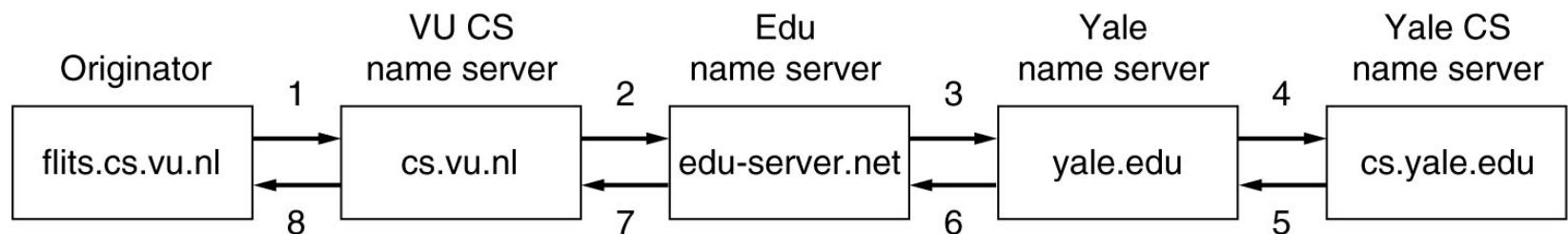
- (1) = resolver invoked by client AP with the name of the server host
- (2) = resolver sends a type-A query containing the name of the server host to its local DNS
- (3) = local DNS returns a type-A resource record containing the IP address of the server
- (4) = resolver returns IP address of the server to the client AP
- ©(5)/(6) = client and server APs carry out networked application/transaction



# Risoluzione nomi remoti



- Se si richiedono info che il name server locale non ha, il server comincia a cercare dal top level per il dominio richiesto
- Es **flits.cs.vu.nl** chiede al suo NS **cs.vu.nl** di trovare **cs.yale.edu**





# Primari e secondari



- Il server primario risponde leggendo le informazioni dal disco
- Il server secondario risponde usando le informazioni ottenute dal server primario con un zone transfer (via TCP)
- Entrambi però sono server authoritative per il dominio



# Posta elettronica



- Anticamente consisteva di un protocollo di file transfer in cui per convenzione la prima linea del file conteneva l'indirizzo del destinatario
- Svantaggi
  - Difficile mandare un mail a diversi destinatari
  - I messaggi non hanno struttura interna, per esempio difficile estrarre un messaggio forwardato dall'interno di un messaggio
  - Il mittente non sapeva mai se il messaggio era arrivato
  - Interfaccia utente molto povera. In pratica un editor in cui aprire il file, aggiungere il destinatario. Poi uscire dal file per invocare un file transfer
  - Molto difficile il supporto a contenuti multimediali



# standard



- RFC 821 protocollo di trasmissione
- RFC 822 formato del messaggio
- RFC 2821 e 2822 revisioni successive diventate standard internet
- Tutti in realtà chiamano la posta di internet con il vecchio nome di RFC 822



# sottosistemi



- MUA message user agent
  - È la parte che permette di leggere e spedire e-mail
  - Sono programmi che forniscono una interfaccia, basata su comandi o su menu o metodi grafici per interagire con il sistema di posta elettronica
- MTA message transfer agent
  - Sposta il messaggio da sorgente a destinazione
  - Sono solitamente systems daemon, cioè processi che girano in background



# Envelope ed header



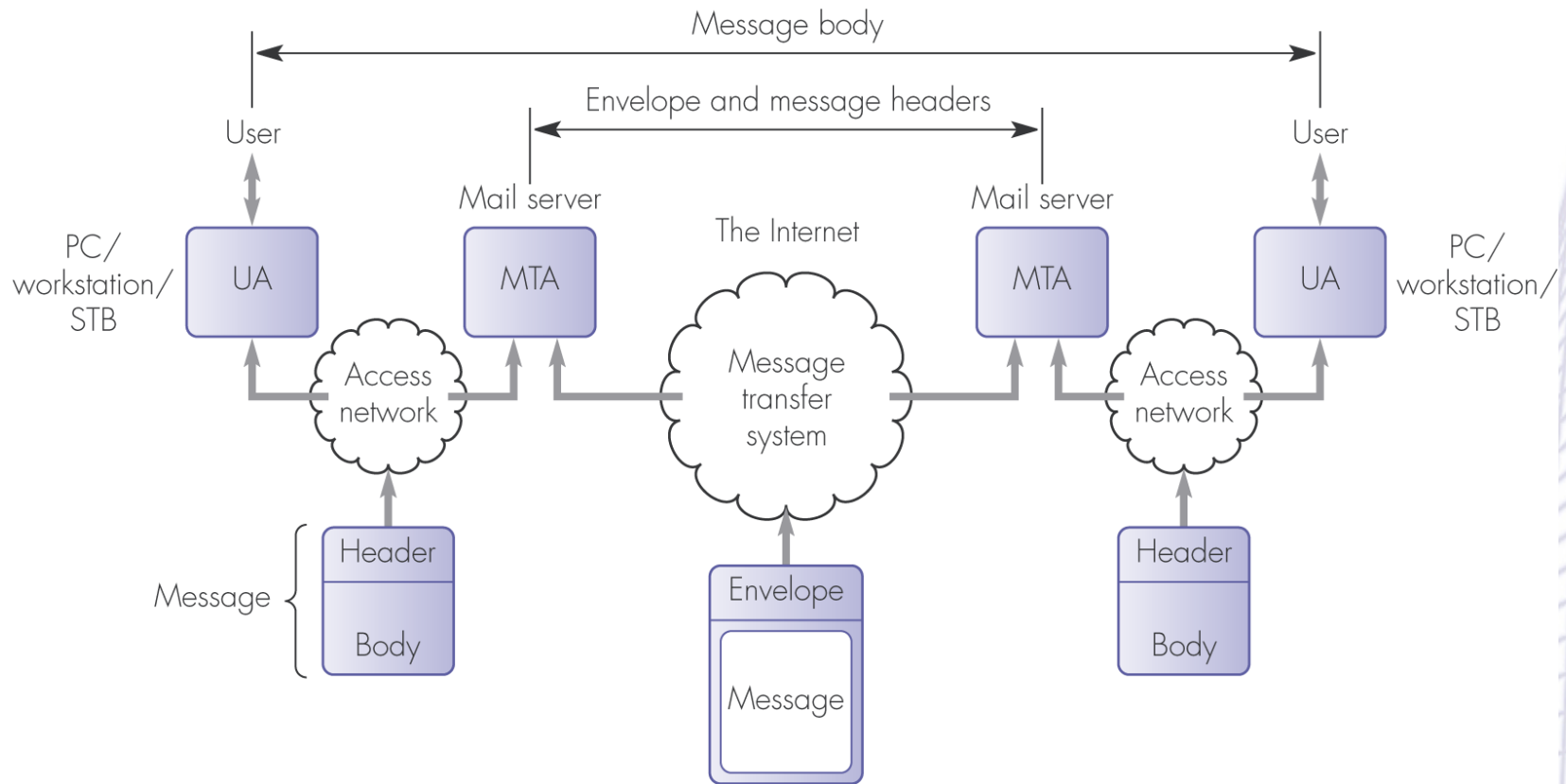
- Envelope è la busta che incapsula il messaggio
  - Contiene le informazioni per trasportare il messaggio, indirizzo di destinazione, priorità
  - L'MTA usa queste informazioni per il routing dei messaggi come farebbe l'ufficio postale
- Il Messaggio contenuto dentro l'-envelope è diviso in
  - **Header**: Informazioni per lo MUA
  - **Body**: Informazioni per l'utente umano





# Struttura dei messaggi

(a)





# Envelope e message



Behrouz Forouzan  
De Anza College  
Cupertino, CA 96014

Sophia Fegan  
Com-Net  
Cupertino, CA 95014

Sophia Fegan  
Com-Net  
Cupertino, CA 95014  
Jan. 5, 2005

Subject: Network

Dear Ms. Fegan:  
We want to inform you that  
our network is working pro-  
perly after the last repair.

Yours truly,  
Behrouz Forouzan

a. Postal mail

Mail From: forouzan@deanza.edu  
RCPT To: fegan@comnet.com

From: Behrouz Forouzan  
To: Sophia Fegan  
Date: 1/5/05  
Subject: Network

Dear Ms. Fegan:  
We want to inform you that  
our network is working pro-  
perly after the last repair.

Yours truly,  
Behrouz Forouzan

b. Electronic mail





# Formato del messaggio



- Envelope: descritta in RFC 821
- Un certo numero di header
- Un linea vuota
- Message body
- Ogni campo consiste di una linea di testo ASCII che contiene il nome del campo, “:”, e il valore del campo
- RFC 822 non distingue chiaramente tra i campi envelope e i campi header
- Il MUA crea un messaggio e lo passa all'MTA che usa alcuni di questi campi per costruirsi l'envelope vera e propria



# Campi usati dal MTA

- **From:** chi ha scritto il messaggio
- **Sender:** chi lo ha mandato
- Es un manager che scrive un messaggio ma lo fa mandare dalla segretaria
- **From:** è obbligatorio mentre Sender può essere omesso se uguale a From:
- **Received:** vanno letti dal basso in alto per capire il percorso del messaggio

Header	Meaning
To:	E-mail address(es) of primary recipient(s)
Cc:	E-mail address(es) of secondary recipient(s)
Bcc:	E-mail address(es) for blind carbon copies
From:	Person or people who created the message
Sender:	E-mail address of the actual sender
Received:	Line added by each transfer agent along the route
Return-Path:	Can be used to identify a path back to the sender



# Campi usati dal MUA

- Campi usati dallo User Agent o dal destinatario umano:
- **Reply-To**: se non voglio che si risponda al mittente
- **X-voglio-andare-in-ferie**: Gli utenti si possono inventare nuovi header purché abbiano una X- come prefisso

Header	Meaning
Date:	The date and time the message was sent
Reply-To:	E-mail address to which replies should be sent
Message-Id:	Unique number for referencing this message later
In-Reply-To:	Message-Id of the message to which this is a reply
References:	Other relevant Message-Ids
Keywords:	User-chosen keywords
Subject:	Short summary of the message for the one-line display



# Riassunto campi



(b)

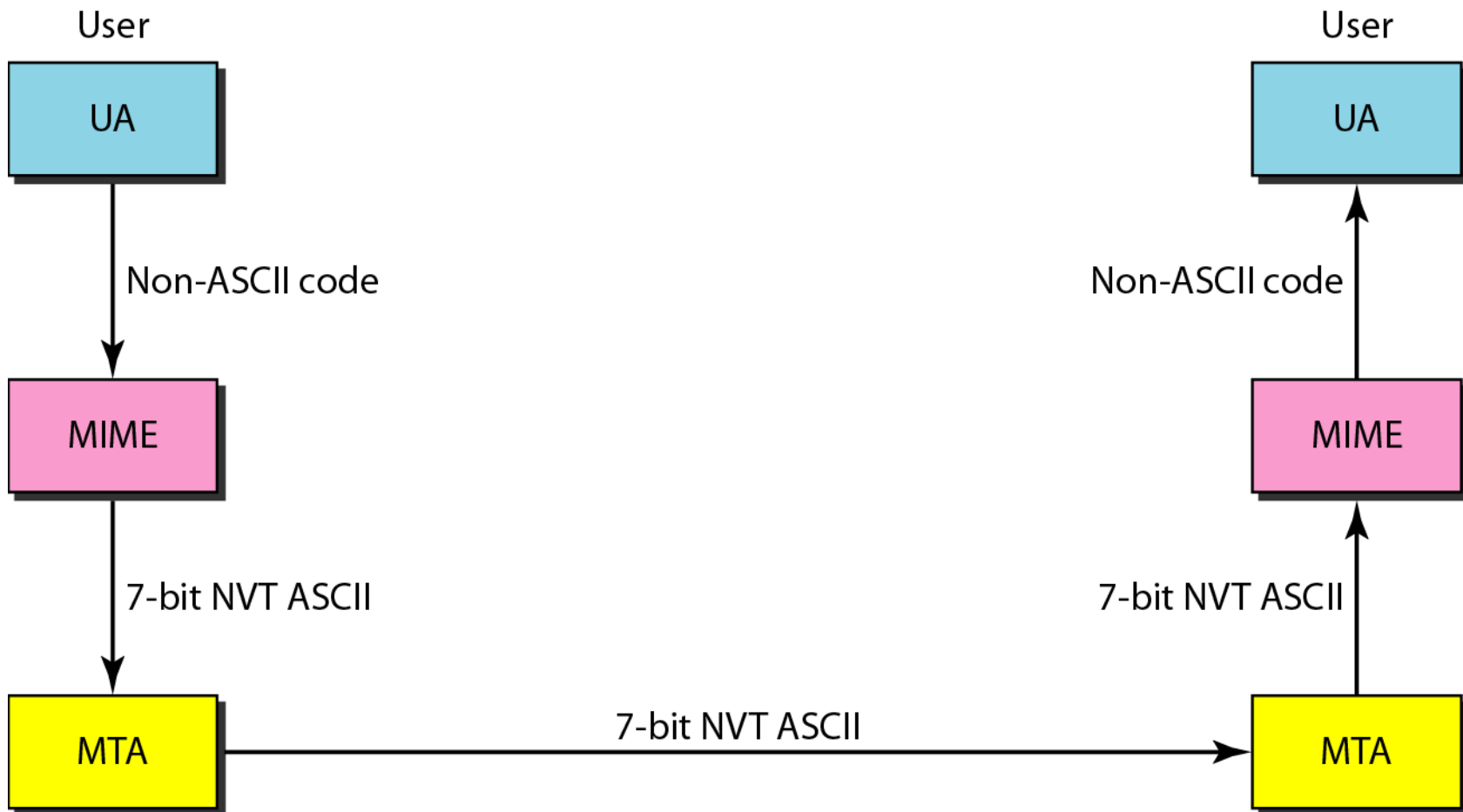
Used also by the message transfer system	{	From:	E-mail address of person who created the message
		To:	E-mail address of primary recipient
		Cc:	List of e-mail addresses of other (secondary) recipients
		Received:	Route followed through message transfer system
		Return-Path:	Name of last MTA
Used by UA/user	{	Sender:	E-mail address of the sender of the message
		Date:	Date and time message was sent by UA
		Message-Id:	Unique identifier assigned to the message by the UA
		Reply-To:	E-mail address to which a reply should be sent
		Subject:	Single-line title of the message
User-defined	{	X-PhoneNumber:	Sender's phone number
		X-FaxNumber:	Sender's fax number



- Multipurpose Internet Mail Extension (RFC 2045 – 2049)
  - All’inizio i mail erano solo testo inglese in ASCII a 7bit
  - Ora sono necessari mail in lingue con accenti (Italiano, Francese, Tedesco)
  - Alfabeti non latini (Cirillico, Ebraico)
  - Linguaggi senza alfabeti (Cinese, Giapponese)
  - Messaggi senza testo (audio, immagini)
  - Si possono aggiungere dei contenuti MIME prefissati da un header MIME



# MIME







# Header MIME

- Content-Transfer-Encoding: dice come trasferire in rete contenuti più complicati del testo, ci sono 5 schemi e la possibilità di crearne altri
  - ASCII 7 bit, ok per testo con linee fino a 1000 caratteri
  - ASCII 8 bit, viola la versione originale del protocollo, ma dichiararlo aiuta a capire da dove vengono eventuali problemi
  - Base64 encoding (gruppi di 24 bit viaggiano come 4 unità da 6 caratteri ASCII regolari), usato per contenuti binari come eseguibili o immagini
  - Quoted-printable encoding per test quasi completamente ASCII con pochi caratteri non ASCII per i quali base64 sarebbe inefficiente
  - User defined

Header	Meaning
MIME-Version:	Identifies the MIME version
Content-Description:	Human-readable string telling what is in the message
Content-Id:	Unique identifier
Content-Transfer-Encoding:	How the body is wrapped for transmission
Content-Type:	Type and format of the content



# Header MIME

- Content-Type: specifica la natura del body del messaggio
- Ci sono sette tipi definiti nella RFC 2045 ma se ne possono aggiungere a piacere
- Vediamo la lista nella prossima slide

Header	Meaning
MIME-Version:	Identifies the MIME version
Content-Description:	Human-readable string telling what is in the message
Content-Id:	Unique identifier
Content-Transfer-Encoding:	How the body is wrapped for transmission
Content-Type:	Type and format of the content



# Tipi MIME e sottotipi



Type	Subtype	Description
Text	Plain	Unformatted text
	Enriched	Text including simple formatting commands
Image	Gif	Still picture in GIF format
	Jpeg	Still picture in JPEG format
Audio	Basic	Audible sound
Video	Mpeg	Movie in MPEG format
Application	Octet-stream	An uninterpreted byte sequence
	Postscript	A printable document in PostScript
Message	Rfc822	A MIME RFC 822 message
	Partial	Message has been split for transmission
	External-body	Message itself must be fetched over the net
Multipart	Mixed	Independent parts in the specified order
	Alternative	Same message in different formats
	Parallel	Parts must be viewed simultaneously
	Digest	Each part is a complete RFC 822 message



# Esempio di mail MIME

From: elinor@abcd.com  
To: carolyn@xyz.com  
MIME-Version: 1.0  
Message-Id: <0704760941.AA00747@abcd.com>  
Content-Type: multipart/alternative; boundary=qwertyuiopasdfghjklzxcvbnm  
Subject: Earth orbits sun integral number of times

This is the preamble. The user agent ignores it. Have a nice day.

--qwertyuiopasdfghjklzxcvbnm  
Content-Type: text/enriched

Happy birthday to you  
Happy birthday to you  
Happy birthday dear <bold> Carolyn </bold>  
Happy birthday to you

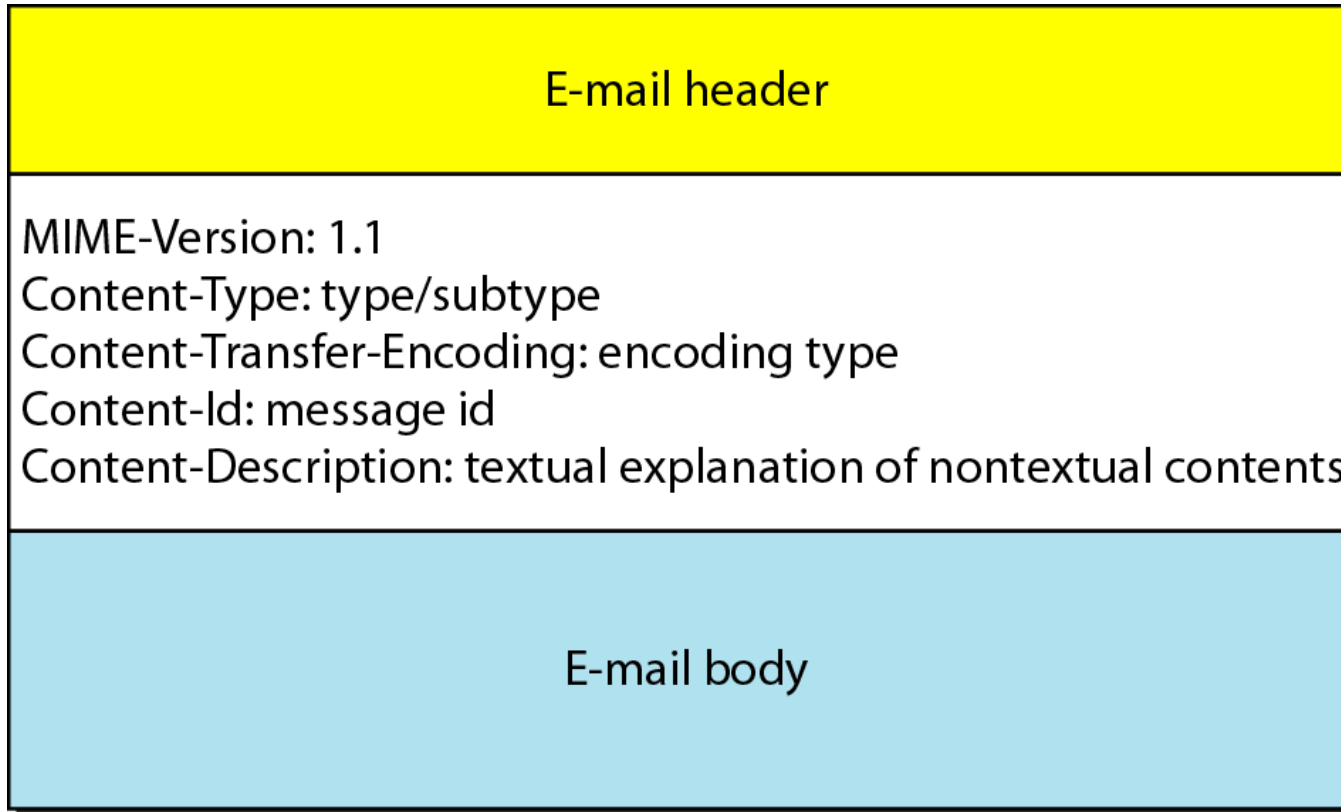
--qwertyuiopasdfghjklzxcvbnm  
Content-Type: message/external-body;  
access-type="anon-ftp";  
site="bicycle.abcd.com";  
directory="pub";  
name="birthday.snd"

content-type: audio/basic  
content-transfer-encoding: base64  
--qwertyuiopasdfghjklzxcvbnm--

- Messaggio di auguri con contenuto multimediale esterno
- Se l'utente non ha capacità multimediali gli appare solo il text enriched nel silenzio più assoluto



# Mail MIME



MIME headers



# Message Transfer



- SMTP: Simple Mail Transfer Protocol
  - Si occupa della consegna dei messaggi
  - Nel caso più semplice da una sorgente direttamente a destinazione
  - Un semplice protocollo ASCII
  - Connessione TCP sulla porta 25
  - Il mittente opera come un client, aspetta che il server parli per primo
  - Il server si presenta e annuncia che è pronto a ricevere un mail
  - Il client ora annuncia da dove viene il mail e a chi è destinato. Se il destinatario esiste il server dà l'ok per continuare
  - Il client manda il messaggio e il server invia ricevuta
  - Non c'è alcun checksum dal momento che si opera su data-stream affidabile



# Comandi SMTP



<i>Keyword</i>	<i>Argument(s)</i>
HELO	Sender's host name
MAIL FROM	Sender of the message
RCPT TO	Intended recipient of the message
DATA	Body of the mail
QUIT	
RSET	
VERFY	Name of recipient to be verified
NOOP	
TURN	
EXPN	Mailing list to be expanded
HELP	Command name
SEND FROM	Intended recipient of the message
SMOL FROM	Intended recipient of the message
SMAL FROM	Intended recipient of the message



# Risposte



<i>Code</i>	<i>Description</i>
<b>Positive Completion Reply</b>	
<b>211</b>	System status or help reply
<b>214</b>	Help message
<b>220</b>	Service ready
<b>221</b>	Service closing transmission channel
<b>250</b>	Request command completed
<b>251</b>	User not local; the message will be forwarded
<b>Positive Intermediate Reply</b>	
<b>354</b>	Start mail input
<b>Transient Negative Completion Reply</b>	
<b>421</b>	Service not available
<b>450</b>	Mailbox not available
<b>451</b>	Command aborted: local error
<b>452</b>	Command aborted: insufficient storage

<i>Code</i>	<i>Description</i>
<b>Permanent Negative Completion Reply</b>	
<b>500</b>	Syntax error; unrecognized command
<b>501</b>	Syntax error in parameters or arguments
<b>502</b>	Command not implemented
<b>503</b>	Bad sequence of commands
<b>504</b>	Command temporarily not implemented
<b>550</b>	Command is not executed; mailbox unavailable
<b>551</b>	User not local
<b>552</b>	Requested action aborted; exceeded storage location
<b>553</b>	Requested action not taken; mailbox name not allowed
<b>554</b>	Transaction failed





# 3-way handshake

## Esempio



```
S: 220 xyz.com SMTP service ready
C: HELO abcd.com
S: 250 xyz.com says hello to abcd.com
C: MAIL FROM: <elinor@abcd.com>
S: 250 sender ok
C: RCPT TO: <carolyn@xyz.com>
S: 250 recipient ok
C: DATA
S: 354 Send mail; end with "." on a line by itself
C: From: elinor@abcd.com
C: To: carolyn@xyz.com
C: MIME-Version: 1.0
C: Message-Id: <0704760941.AA00747@abcd.com>
C: Content-Type: multipart/alternative; boundary=qwertyuiopasdfghjklzxcvbnm
C: Subject: Earth orbits sun integral number of times
C:
C: This is the preamble. The user agent ignores it. Have a nice day.
C:
C: --qwertyuiopasdfghjklzxcvbnm
C: Content-Type: text/enriched
C:
C: Happy birthday to you
C: Happy birthday to you
C: Happy birthday dear <bold> Carolyn </bold>
C: Happy birthday to you
C:
C: --qwertyuiopasdfghjklzxcvbnm
C: Content-Type: message/external-body;
C:     access-type="anon-ftp";
C:     site="bicycle.abcd.com";
C:     directory="pub";
C:     name="birthday.snd"
C:
C: content-type: audio/basic
C: content-transfer-encoding: base64
C: --qwertyuiopasdfghjklzxcvbnm
C: .
S: 250 message accepted
C: QUIT
S: 221 xyz.com closing connection
```

## Notare

I comandi client di 4 caratteri. La sintassi lato client è molto rigida

Le risposte del server prefissate da un codice numerico (la parte testo è molto libera)



# Altro esempio



- *\$ telnet mail.adelphia.net 25*
- *Trying 68.168.78.100 . . .*
- *Connected to mail.adelphia.net (68.168.78.100).*

===== Connection Establishment =====

**220 mta13.adelphia.net SMTP server ready Fri, 6 Aug 2004 . . .**

**HELO mail.adelphia.net**

**250 mta13.adelphia.net**

===== Mail Transfer =====

**MAIL FROM: forouzanb@adelphia.net**

**250 Sender <forouzanb@adelphia.net> Ok**

**RCPT TO: forouzanb@adelphia.net**

**250 Recipient <forouzanb@adelphia.net> Ok**

**DATA**

**354 Ok Send data ending with <CRLF>.<CRLF>**

**From: Forouzan**

**TO: Forouzan**

**This is a test message  
to show SMTP in action.**



# Esercizio



- Contattare un server SMTP su porta 25
- Usare il comando HELP
- Provare a mandare un mail via telnet



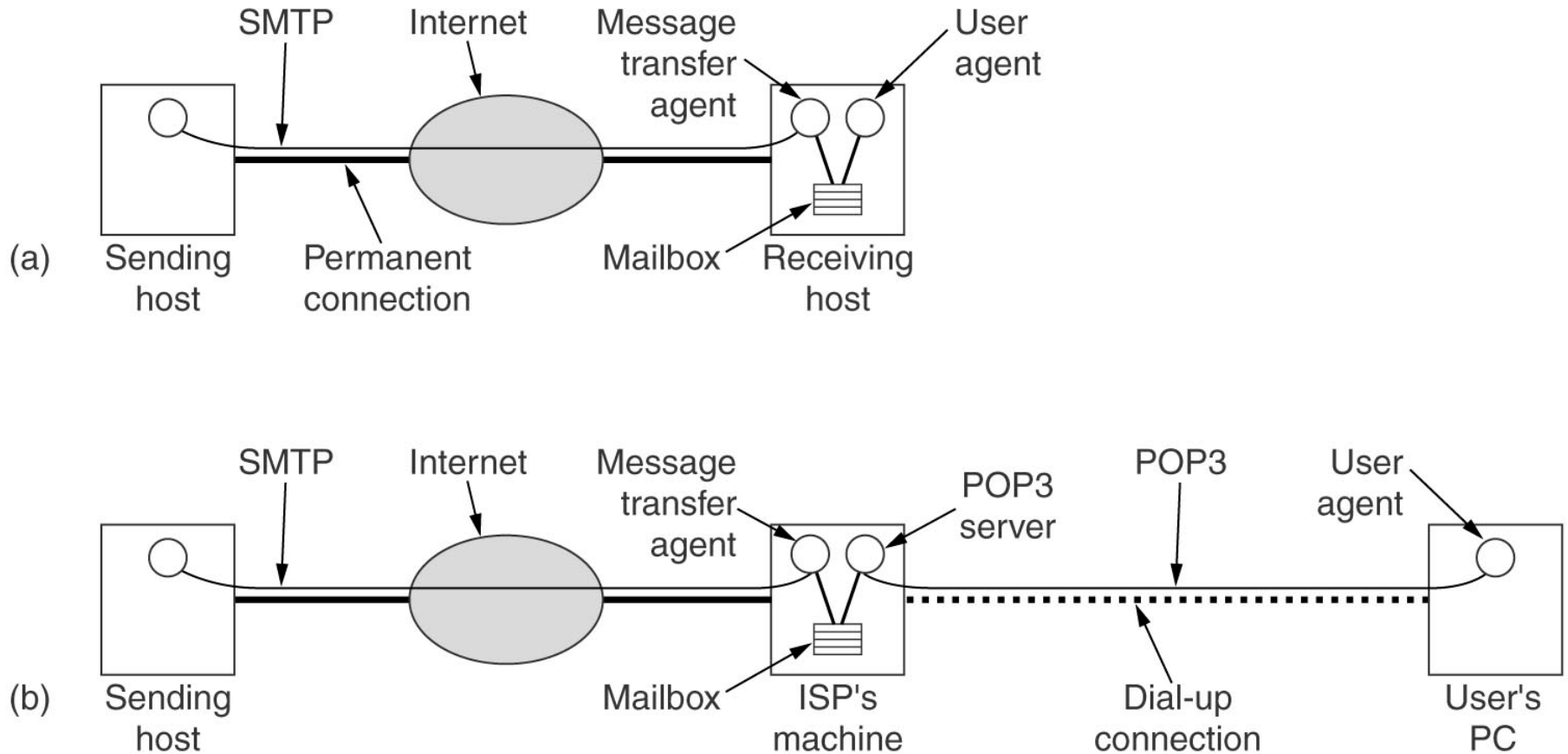
# Letture dei mail



- Come accedo ai mail ricevuti?
  - Una volta tutti gli utenti lavoravano nella macchina che anche riceveva i mail
  - In questo caso il mio MUA accede direttamente alle mailbox nel filesystem locale
  - Altrimenti, se accedo remotamente da un PC, smartphone o comunque da un altro host, devo usare protocolli come POP o IMAP o una forma di webmail (es: hotmail, gmail, yahoo)



# Invio e ricezione





# POP3



- Descritto in RFC 1939
  - Il mail reader chiama il POP server sulla porta TCP 110
  - Dopo la connessione il protocollo attraversa la sequenza Autorizzazione, Transazione, Update
  - In questo modo l'utente si collega all'Internet Service Provider, si scarica i mail e li legge con calma localmente
  - In teoria il protocollo permette di leggere un messaggio specifico e non solo tutti insieme
  - Inoltre il messaggio può essere lasciato nella mailbox che non deve essere necessariamente cancellata



# Scambio POP3

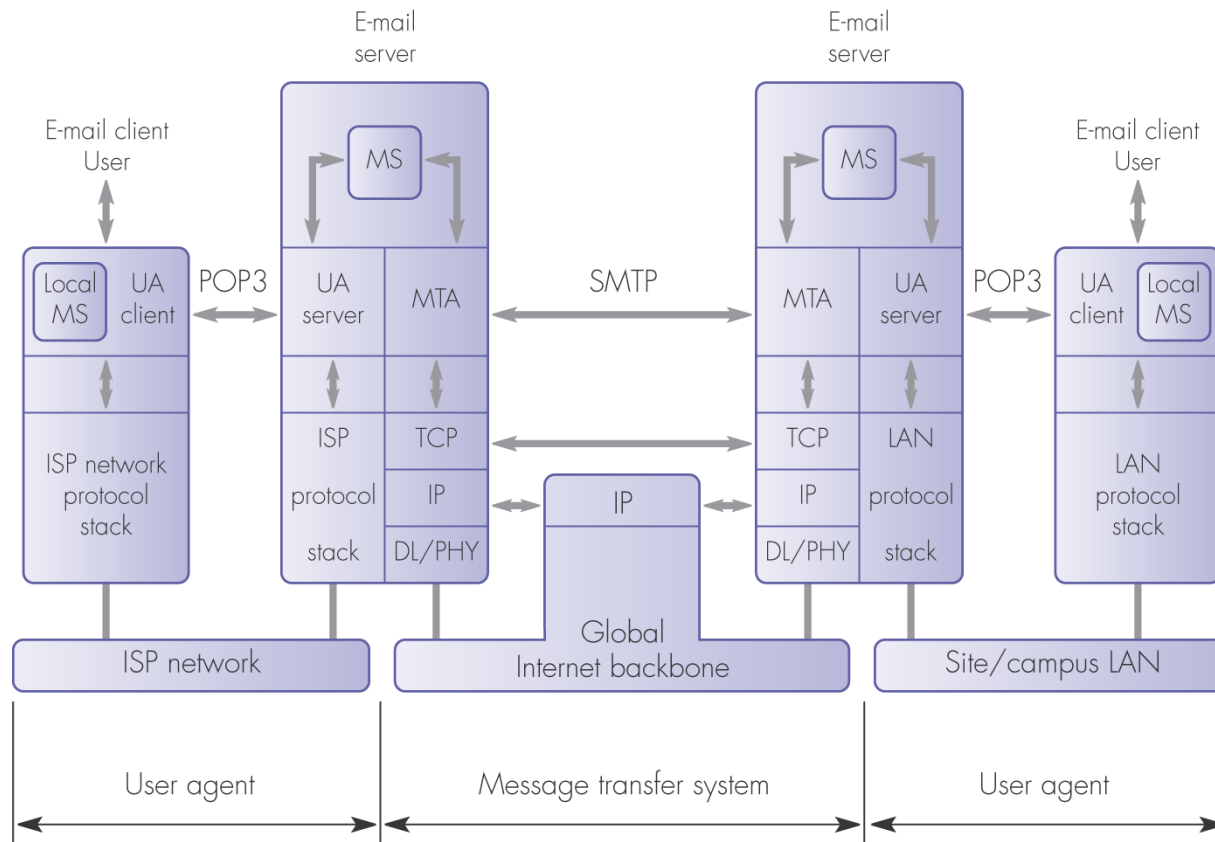


```
S: +OK POP3 server ready
C: USER carolyn
S: +OK
C: PASS vegetables
S: +OK login successful
C: LIST
S: 1 2505
S: 2 14302
S: 3 8122
S: .
C: RETR 1
S: (sends message 1)
C: DELE 1
C: RETR 2
S: (sends message 2)
C: DELE 2
C: RETR 3
S: (sends message 3)
C: DELE 3
C: QUIT
S: +OK POP3 server disconnecting
```

Numero messaggio e lunghezza messaggio



# Percorso completo



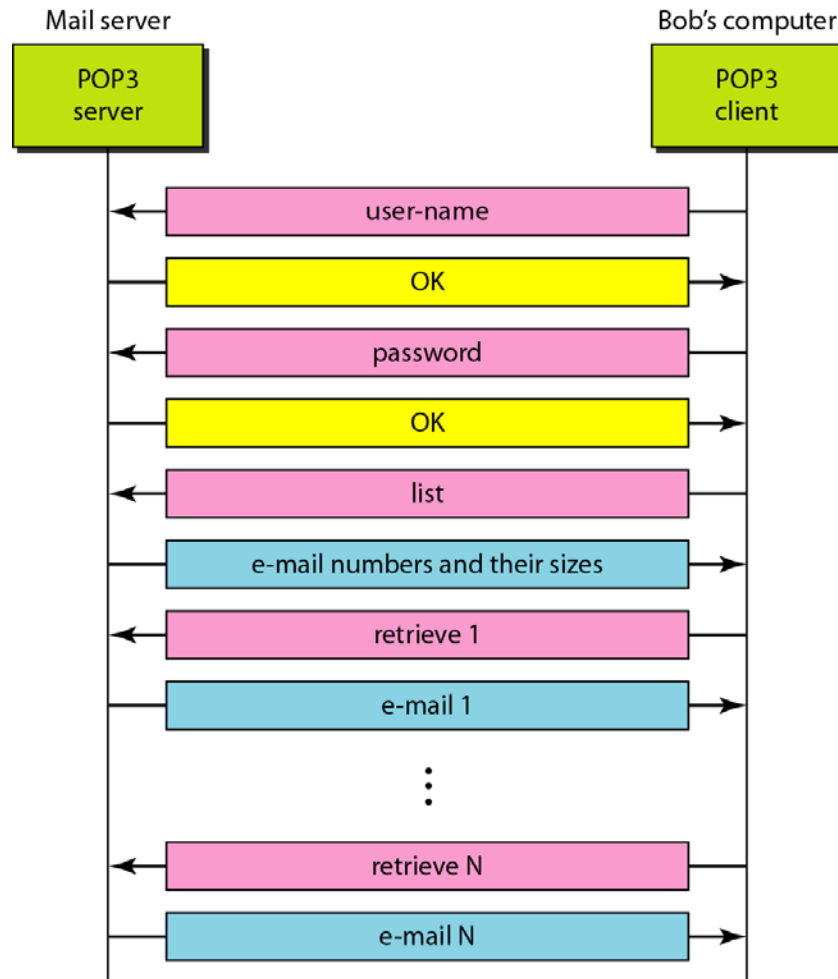
POP = post office protocol  
SMTP = simple mail transfer protocol  
ISP = Internet service provider  
MTA = message transfer agent  
UA = user agent  
DL/PHY = data link/physical layer

MS = message store (contains a queue of mail to be sent and an IN mailbox for each of its local users)





# Esempio POP3





- Internet Message Access Protocol RFC 2060
  - Al contrario di POP - che prevede che l'utente ogni volta svuoti la mailbox e poi lavori offline - IMAP è pensato per uso online
  - Inoltre è pensato per funzionare con diverse mailbox
  - Lavora di default sulla porta 143



# Confronto POP IMAP



<b>Feature</b>	<b>POP3</b>	<b>IMAP</b>
Where is protocol defined	RFC 1939	RFC 2060
TCP port used	110	143
Where is e-mail stored	User's PC	Server
Where is e-mail read	Off-line	On-line
Connect time required	Little	Much
Use of server resources	Minimal	Extensive
Multiple mailboxes	No	Yes
Who backs up mailboxes	User	ISP
Good for mobile users	No	Yes
User control over downloading	Little	Great
Partial message downloads	No	Yes
Are disk quotas a problem	No	Could be in time
Simple to implement	Yes	No
Widespread support	Yes	Growing



# FTP



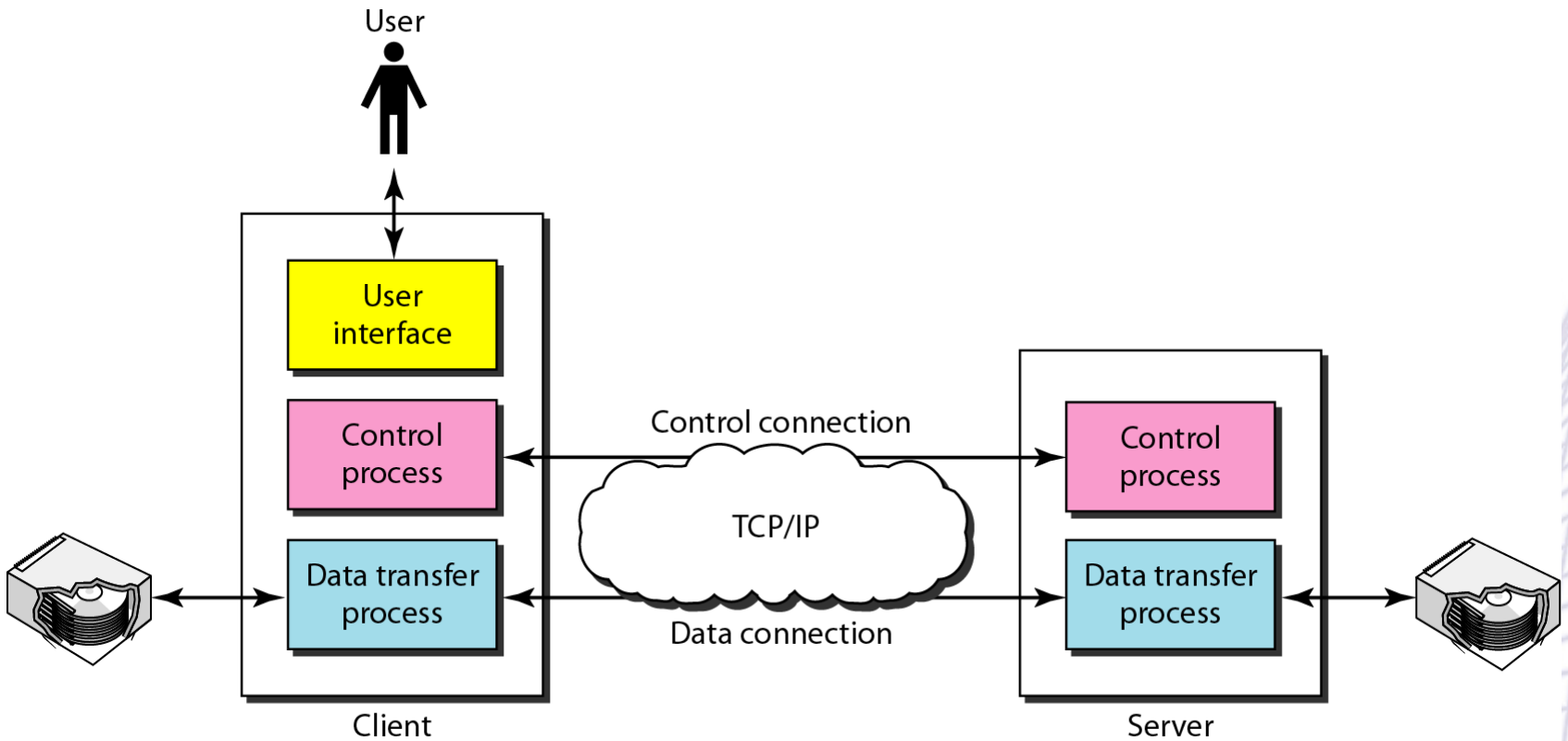
- **File Transfer Protocol**
- Prima di http si usava solo questo protocollo. C'erano dei server con accesso anonimo con grandi liste di altri server e dei file presenti su questi server
- Migliaia di file di vario contenuto si potevano recuperare poi accedendo a mano e navigando all'intero dei server
- L'accesso normale sarebbe via username e password ma alcuni server accettano come username "**anonymous**" e come password qualcosa che possa assomigliare ad un indirizzo e-mail. In questo caso viene dato accesso ad un sottoinsieme read-only del filesystem del server



# Tipica connessione FTP

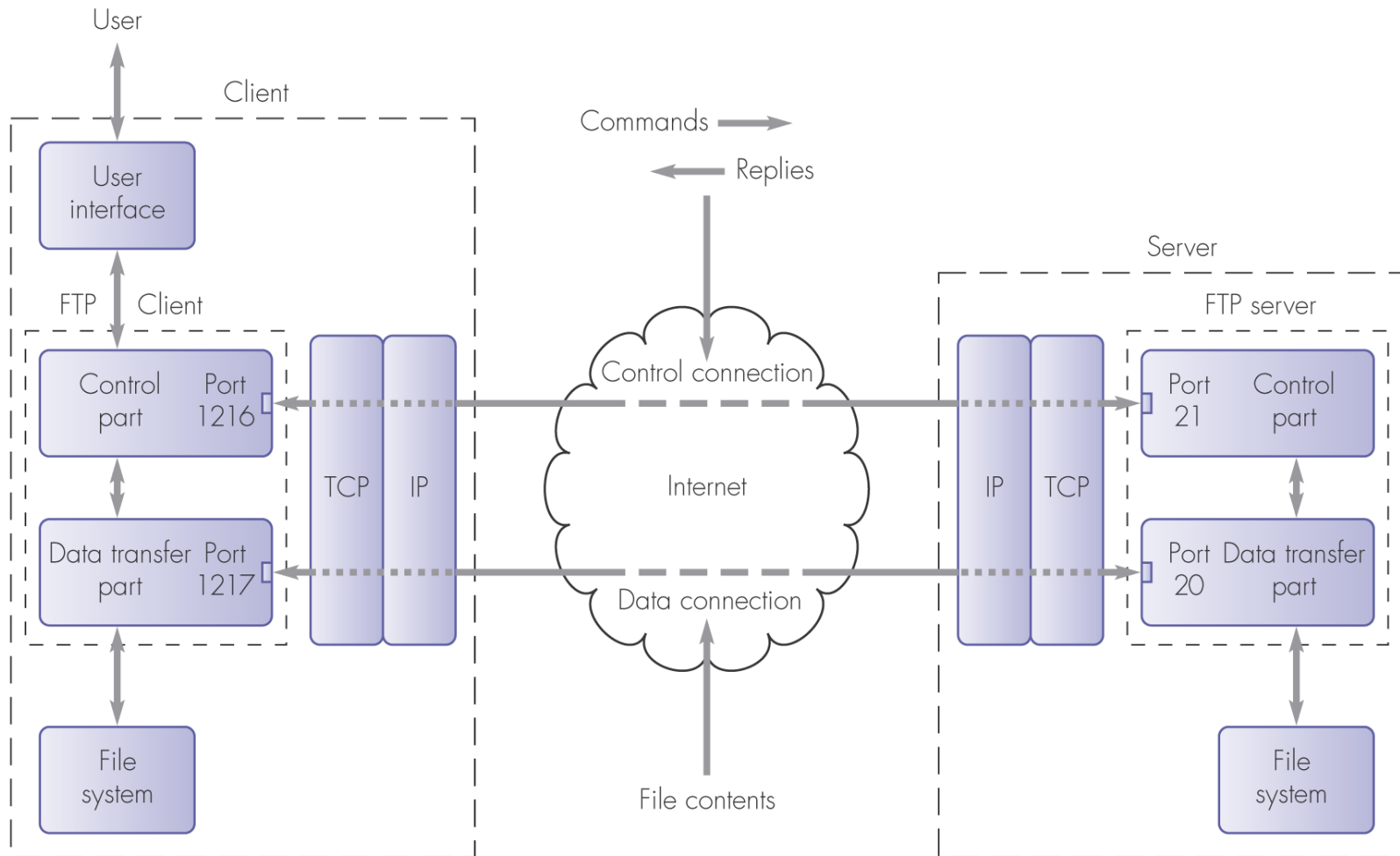


Notare le due connessioni, dati e controllo, separate





# Protocolli usati





# Active vs Passive

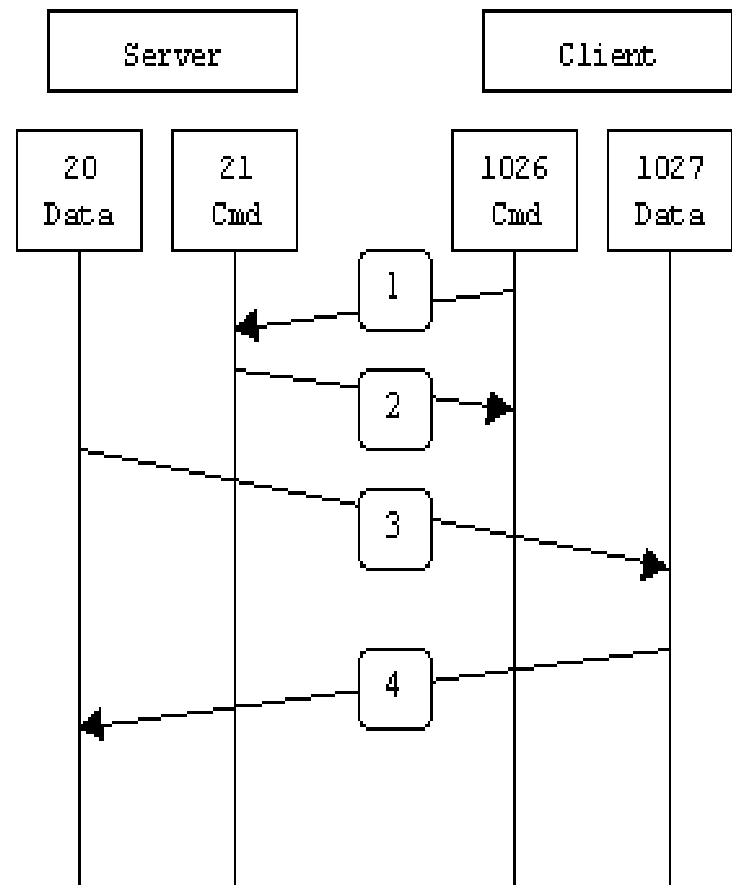


- FTP in modo inusuale utilizza due porte
  - Una porta “data”
  - Una porta “comand” o “control”
  - Solitamente 21 per “command” e 20 per “data”
  - Invece a seconda del modo “attivo” o “passivo” | non sempre la 20 fa da porta “data”



# Active FTP

1. accesso alla porta 21 del server da qualche porta del client ( $>1023$ )
2. Il server risponde alla porta di controllo del client
3. La porta 20 del server apre una connessione dati ad una porta specificata dal client in precedenza
4. Il client manda ACK alla porta del server







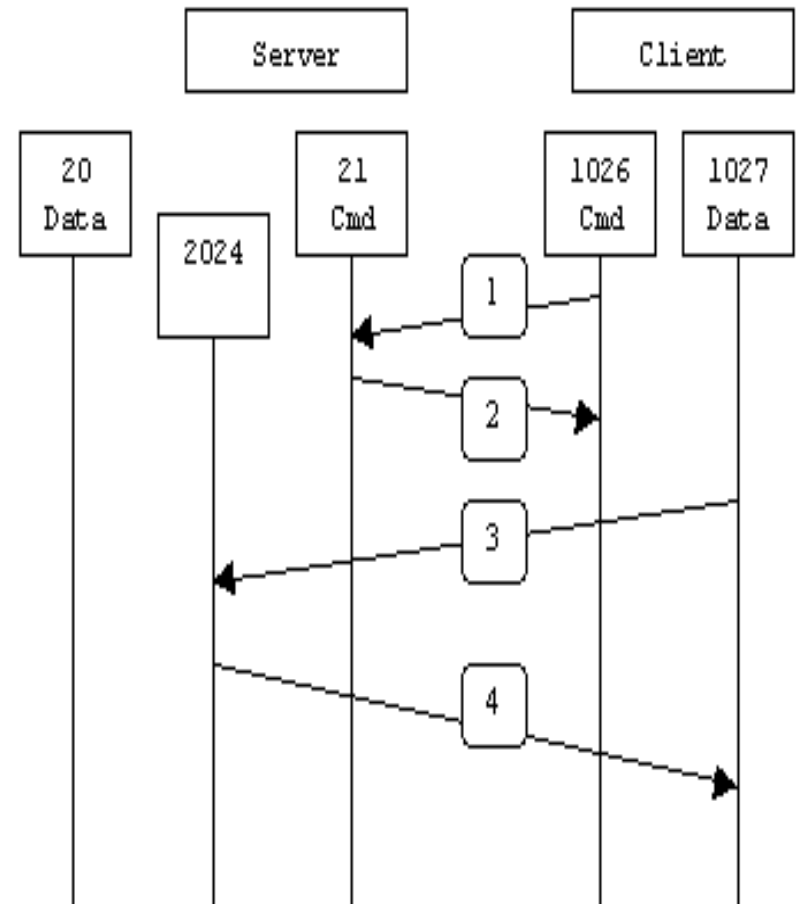
# Passive FTP

Se il client sta dietro un firewall che non lascia connessioni entranti?

Dobbiamo usare il Passive FTP

1. accesso alla porta 21 del server da qualche porta del client (>1023)
2. Il server risponde alla porta di controllo del client
3. Il server accetta una connessione dati da una porta del client (>1023)
4. Il server manda ACK alla porta del client

Molti browser web supportano solo Passive mode quando accedono a ftp://URL





# Comandi del client



!	cr	macdef	proxy	sendport
\$	delete	mdelete	put	status
account	debug	mdir	pwd	struct
append	dir	mget	quit	sunique
ascii	disconnect	mkdir	quote	tenex
bell	form	mls	recv	trace
binary	get	mode	remotehelp	type
bye	glob	mput	rename	user
case	hash	nmap	reset	verbose
cd	help	ntrans	rmdir	?
cdup	lcd	open	runique	
close	ls	prompt	send	



# Esempio



```
$ ftp
ftp> open
(to) ftp.cs.purdue.edu
Connected to lucan.cs.purdue.edu
220 lucan.cs.purdue.edu FTP server Thu Jun 15 ...READY
Name (ftp.cs.purdue.edu:michelot): anonymous
331 Guest login ok, send your complete e-mail address as password
Password:
230- vario testo di banner
230 Guest login ok, access restrictions apply
ftp> pwd
257 "/" is current directory
ftp> ls
200 PORT command successful.
550 No files found
```



# Esempio



```
ftp> ls pub/Xinu  
200 PORT command successful.  
150 Opening ASCII mode data connection for file list.  
pub/Xinu/CONC.Z  
....  
pub/Xinu/XINUX-VAX.TAR.Z  
226 Transfer complete.  
remote pub/Xinu  
335 bytes received in 0.0025 seconds (139.96 Kbytes/sec)  
ftp> cd pub/Xinu
```



# Esempio



```
ftp> cd pub/Xinu
ftp> get
(remote-file) XINU-PENTIUM.TAR.Z
(local-file) da-cancellare
200 PORT command successful.
150 Opening ASCII mode connection for XINU-PENTIUM.TAR.Z (5017 bytes)
226 Transfer Complete
local: da-cancellare remote:XINU-PENTIUM.TAR.Z
5135 bytes transferred in 0.044 seconds (112.76 Kbytes/s)
....
ftp> close
221- You have transferred 5135 bytes in 1 file
221-Total traffic for this session was 7868 bytes in 2 transfers.
221- Thank you for using the FTP service on lucan.cs.purdue.edu
221 Goodbye
ftp> quit
```



# Esempio ftp



```
$ ftp voyager.deanza.fhda.edu
Connected to voyager.deanza.fhda.edu.
220 (vsFTPd 1.2.1)
530 Please login with USER and PASS.
Name (voyager.deanza.fhda.edu:forouzan): forouzan
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls reports
227 Entering Passive Mode (153,18,17,11,238,169)
150 Here comes the directory listing.
drwxr-xr-x  2 3027      411          4096 Sep 24  2002 business
drwxr-xr-x  2 3027      411          4096 Sep 24  2002 personal
drwxr-xr-x  2 3027      411          4096 Sep 24  2002 school
226 Directory send OK.
ftp> quit
221 Goodbye.
```



# Anonymous ftp



```
$ ftp internic.net
Connected to internic.net
220 Server ready
Name: anonymous
331 Guest login OK, send "guest" as password
Password: guest
```

```
ftp > pwd
257 '/' is current directory
ftp > ls
200 OK
150 Opening ASCII mode
```

```
bin
```

```
...
```

```
...
```

```
...
```

```
ftp > close
221 Goodbye
ftp > quit
```



# World Wide Web

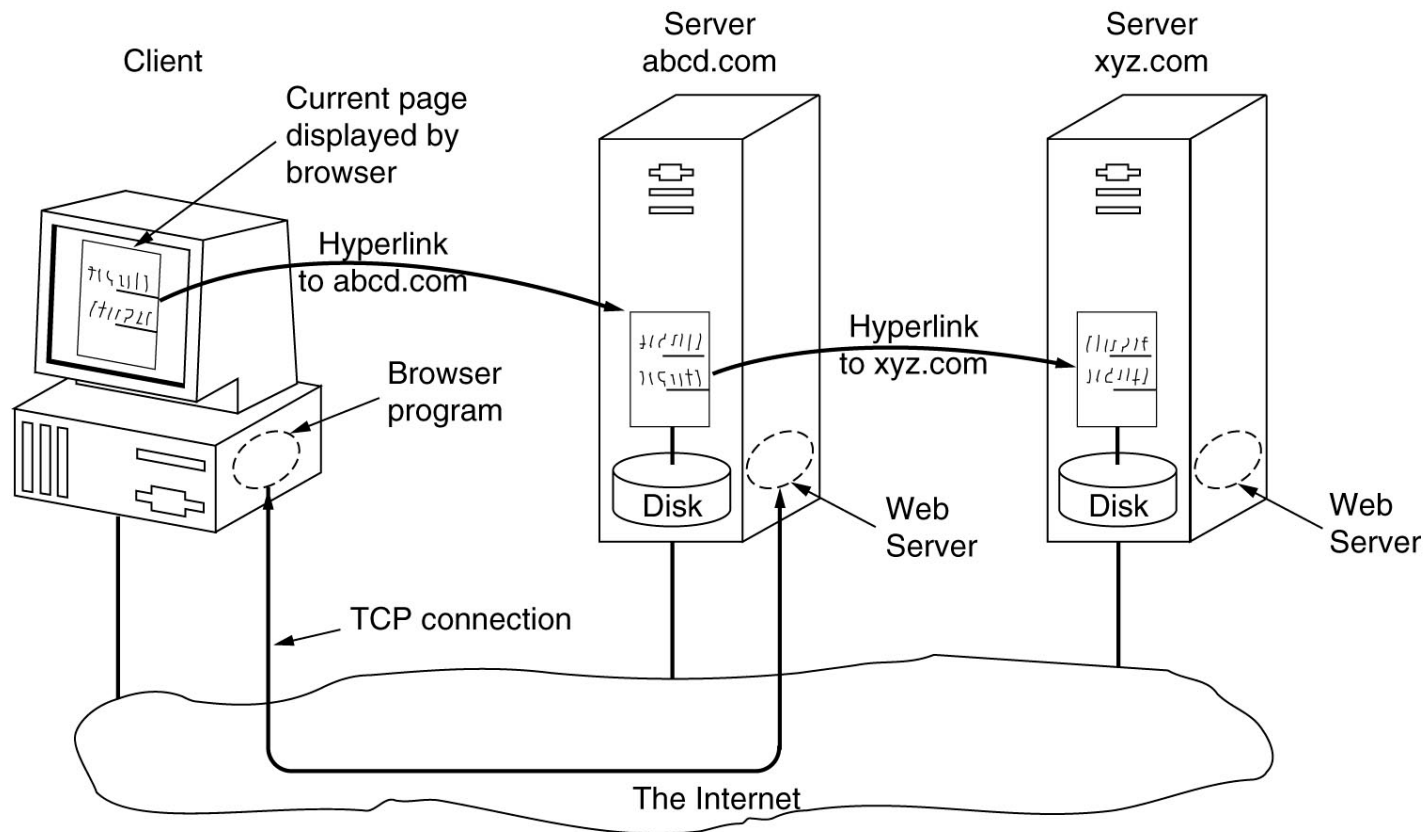


- Architettura client server per accesso a dati ipertestuali distribuiti su diverse macchine
- HTML HyperText Markup Language è il formato con cui vengono descritti gli ipertesti
- HTTP HyperText Transfer Protocol è il protocollo base per accedere a file html
- I collegamenti all'ipertesto sono descritti dalle URL Uniform Resource Locator





# Il modello web





# Client side



- Il browser determina la URL vedendo cosa è stato cliccato (es. **www.pd.infn.it**)
- Il browser chiede al DNS il numero IP corrispondente
- Il DNS risponde **191.34.142.x**
- Il browser si connette alla porta **80** di quell'indirizzo
- Dopo il 3-way handshake il client manda la richiesta del file **/home/index.html**
- Il server risponde mandando il file
- La connessione TCP viene chiusa
- Il browser interpreta il file e lo proietta nell'interfaccia grafica
- Eventualmente il browser richiede le immagini presenti nel file, al server, ad altri server o dalla cache



# URL



- Le URL sono costituite da 3 parti
- **Il metodo (protocollo da usare)** seguito da **://**
- **Il nome del server** che implementa quel metodo e la **porta del daemon**
- **Il nome del file** da chiedere al server
- **Protocol://hostname:port/document-name**



# Alcune URL comuni

Name	Used for	Example
http	Hypertext (HTML)	http://www.cs.vu.nl/~ast/
ftp	FTP	ftp://ftp.cs.vu.nl/pub/minix/README
file	Local file	file:///usr/suzanne/prog.c
news	Newsgroup	news:comp.os.minix
news	News article	news:AA0134223112@cs.utah.edu
gopher	Gopher	gopher://gopher.tc.umn.edu/11/Libraries
mailto	Sending e-mail	mailto:JohnUser@acm.org
telnet	Remote login	telnet://www.w3.org:80

- Altre metodi

- https: http criptato via SSL
- mms: streaming multimediali
- Estensioni proprietarie di alcuni browser
- ftp(s)://<login>:<password>@<ftpserveraddress>:<port>



# Esempio di ipertesto



```
<HTML>
  <HEAD>
    <TITLE>
      text that forms the document title
    </TITLE>
  </HEAD>
  <BODY>
    body of the document appears here
  </BODY>
</HTML>
```

Producono lo  
stesso risultato

```
<HTML><HEAD><TITLE>text that forms
the document
title</TITLE></HEAD><BODY>body of the
document appears here</BODY></HTML>
```



# Esempio



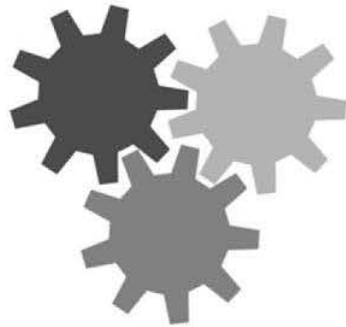
```
<html>
<head> <title> AMALGAMATED WIDGET, INC. </title> </head>
<body> <h1> Welcome to AWI's Home Page </h1>
 <br>
We are so happy that you have chosen to visit <b> Amalgamated Widget's</b>
home page. We hope <i> you </i> will find all the information you need here.
<p>Below we have links to information about our many fine products.
You can order electronically (by WWW), by telephone, or by fax. </p>
<hr>
<h2> Product information </h2>
<ul>
  <li> <a href="http://widget.com/products/big"> Big widgets </a>
  <li> <a href="http://widget.com/products/little"> Little widgets </a>
</ul>
<h2> Telephone numbers </h2>
<ul>
  <li> By telephone: 1-800-WIDGETS
  <li> By fax: 1-415-765-4321
</ul>
</body>
</html>
```



# Esempio



## Welcome to AWI's Home Page



We are so happy that you have chosen to visit **Amalgamated Widget's** home page. We hope *you* will find all the information you need here.

Below we have links to information about our many fine products. You can order electronically (by WWW), by telephone, or by FAX.

---

### Product Information

- [Big widgets](#)
- [Little widgets](#)

### Telephone numbers

- 1-800-WIDGETS
- 1-415-765-4321



# Metodi di richieste HTTP

- HTTP è stato progettato per il web con un occhio per future applicazioni object oriented
- Le operazioni si chiamano metodi (methods)
- Il metodo di gran lunga più richiesto è il **get**
- **GET filename HTTP/1.1**

Method	Description
GET	Request to read a Web page
HEAD	Request to read a Web page's header
PUT	Request to store a Web page
POST	Append to a named resource (e.g., a Web page)
DELETE	Remove the Web page
TRACE	Echo the incoming request
CONNECT	Reserved for future use
OPTIONS	Query certain options





# Risposte del server

- Le 1xx sono usate raramente
- Le 2xx servono in caso di risposte ok
- Le 3xx per redirect (guarda altrove)
- Le 4xx per codici di errore lato client (accesso negato o pagine non esistente)
- Le 5xx per codici di errore lato server, errore nel codice o sovraccarico del server

Code	Meaning	Examples
1xx	Information	100 = server agrees to handle client's request
2xx	Success	200 = request succeeded; 204 = no content present
3xx	Redirection	301 = page moved; 304 = cached page still valid
4xx	Client error	403 = forbidden page; 404 = page not found
5xx	Server error	500 = internal server error; 503 = try again later



# Message header



- La linea di richiesta (per es. la linea con il metodo GET) può essere seguita da linee con informazioni aggiuntive dette **request header** come se avessi dei parametri in una chiamata di funzione
- Anche la risposta può avere dei **response header**



# Esempi di header



Header	Type	Contents
User-Agent	Request	Information about the browser and its platform
Accept	Request	The type of pages the client can handle
Accept-Charset	Request	The character sets that are acceptable to the client
Accept-Encoding	Request	The page encodings the client can handle
Accept-Language	Request	The natural languages the client can handle
Host	Request	The server's DNS name
Authorization	Request	A list of the client's credentials
Cookie	Request	Sends a previously set cookie back to the server
Date	Both	Date and time the message was sent
Upgrade	Both	The protocol the sender wants to switch to
Server	Response	Information about the server
Content-Encoding	Response	How the content is encoded (e.g., gzip)
Content-Language	Response	The natural language used in the page
Content-Length	Response	The page's length in bytes
Content-Type	Response	The page's MIME type
Last-Modified	Response	Time and date the page was last changed
Location	Response	A command to the client to send its request elsewhere
Accept-Ranges	Response	The server will accept byte range requests
Set-Cookie	Response	The server wants the client to save a cookie



# Transazione HTTP



- Esempio di connessione con logging su file

```
telnet www.ietf.org 80 > log
GET /rfc.html
Host: www.ietf.org

close
```

- Dopo la connessione chiedo il file a cui sono interessato e poi c'è il message header Host obbligatorio, anche la linea vuota è obbligatoria
- La close dice al programma telnet di chiudere la connessione
- Vediamo una possibile risposta, le prime tre linee fanno parte del programma telnet e non della risposta del server http

```
Trying 4.17.168.6...
Connected to www.ietf.org.
Escape character is '^]'.
HTTP/1.1 200 OK
Date: Wed, 08 May 2002 22:54:22 GMT
Server: Apache/1.3.20 (Unix) mod_ssl/2.8.4 OpenSSL/0.9.5a
Last-Modified: Mon, 11 Sep 2000 13:56:29 GMT
ETag: "2a79d-c8b-39bce48d"
Accept-Ranges: bytes
Content-Length: 3211
Content-Type: text/html
X-Pad: avoid browser bug
```

## Programma telnet

### Ok, accetto di parlare con te

```
<html>
<head>
<title>IETF RFC Page</title>

<script language="javascript">
function url() {
var x = document.form1.number.value
if (x.length == 1) {x = "000" + x }
if (x.length == 2) {x = "00" + x }
if (x.length == 3) {x = "0" + x }
document.form1.action = "/rfc/rfc" + x + ".txt"
document.form1.submit
}
</script>

</head>
```

Comincia il codice html

Vediamo solo l'header che contiene  
del codice javascript



# Grazie



- Grazie per l'attenzione
- Buon Lavoro