

Corso di Laurea in Ingegneria Informatica



Corso di Reti di Calcolatori

Roberto Canonico (roberto.canonico@unina.it)

Giorgio Ventre (giorgio.ventre@unina.it)

Protocolli applicativi: FTP ed SMTP

**I lucidi presentati al corso sono uno strumento didattico
che NON sostituisce i testi indicati nel programma del corso**



Nota di Copyright

Questo insieme di trasparenze è stato ideato e realizzato dai ricercatori del Gruppo di Ricerca COMICS del Dipartimento di Informatica e Sistemistica dell'Università di Napoli Federico II. Esse possono essere impiegate liberamente per fini didattici esclusivamente senza fini di lucro, a meno di un esplicito consenso scritto degli Autori. Nell'uso dovranno essere esplicitamente riportati la fonte e gli Autori. Gli Autori non sono responsabili per eventuali imprecisioni contenute in tali trasparenze né per eventuali problemi, danni o malfunzionamenti derivanti dal loro uso o applicazione.

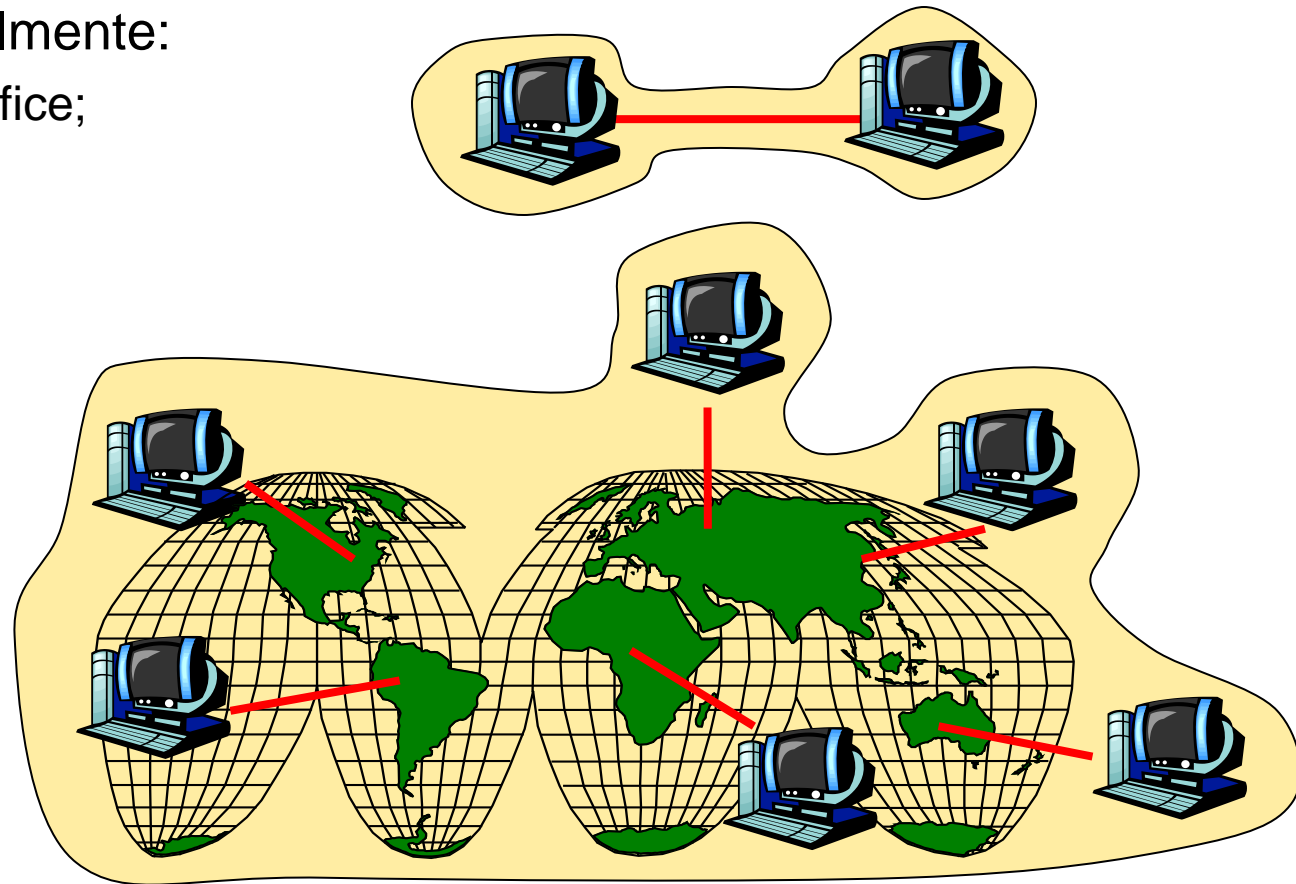
Autori:

Simon Pietro Romano, Antonio Pescapè, Stefano Avallone,
Marcello Esposito, Roberto Canonico, Giorgio Ventre



File Transfer Protocol (FTP)

- Internet oggi si presenta come una rete ad estensione globale che connette molti milioni di macchine sparse su tutto il globo.
- Spesso sorge l'esigenza di copiare un file da una macchina ad un'altra per poterlo utilizzare localmente:
 - un documento di Office;
 - un file eseguibile;
 - un file di testo;
 - etc...
- Ciò può accadere sia tra macchine molto distanti tra di loro che tra macchine direttamente connesse, presenti nello stesso locale.



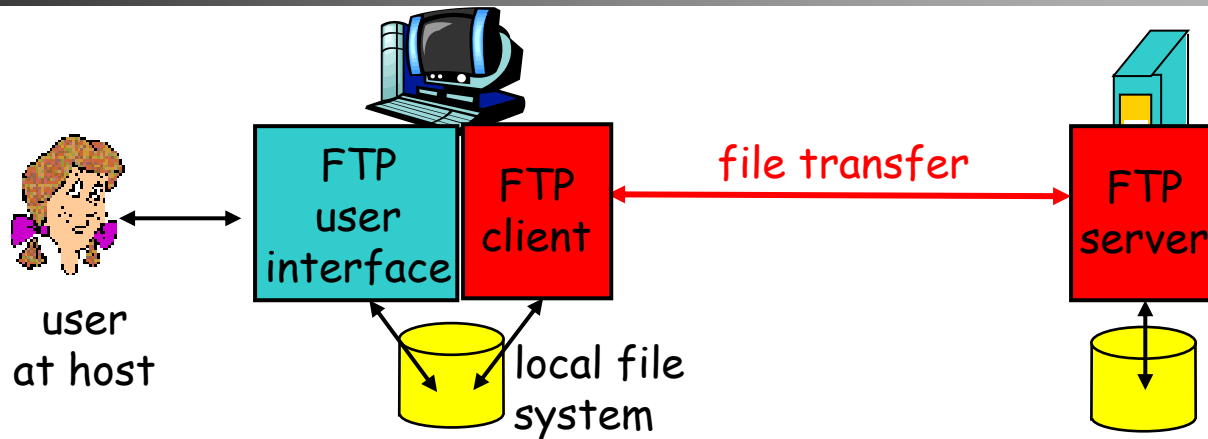


Il protocollo FTP

- Un apposito protocollo è stato definito a questo scopo.
- Si chiama File Transfer Protocol (FTP)
- Attraverso di esso è possibile trasferire uno o più files di qualsiasi tipo tra due macchine.
- Il protocollo FTP è descritto in RFC959
- Lavora utilizzando due connessioni: una dati e una controllo (*out of band*)



Come funziona FTP

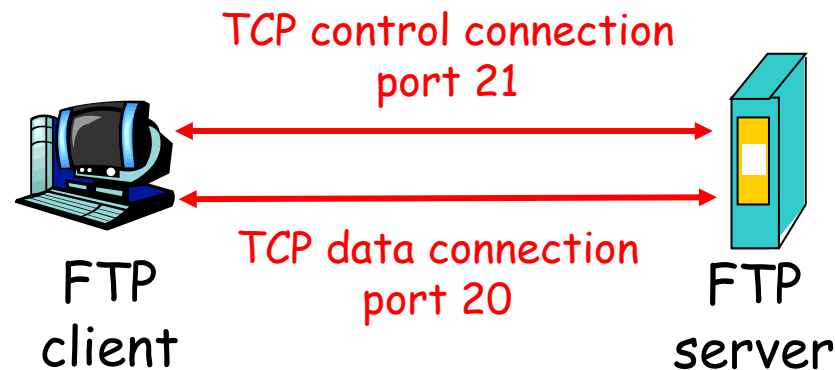


- Trasferisce files da o verso una macchina remota
- Usa il modello client/server
 - **client**: è l'entità che dà luogo al trasferimento (sia in un senso che nell'altro)
 - **server**: è l'entità remota che è in continua attesa di connessioni FTP da parte di altre entità
- ftp server: numero di porto 21



Le connessioni di una sessione FTP

- Il client ftp contatta il server ftp al porto 21;
- Vengono aperte due connessioni parallele:
 - **Controllo**: scambio di comandi, messaggi di risposta tra il client e il server (*controllo “out of band” (fuori banda)*)
 - **Dati**: file che fluiscono dal client al server o viceversa
- Un server ftp mantiene uno stato, per es.:
 - la directory corrente;
 - i dati dell'autenticazione.





Scambio delle informazioni

- I comandi vengono inviati come testo ASCII sulla connessione di controllo
- Anche le risposte sono costituite da testo ASCII
- *NOTA: il testo ASCII è una sequenza di caratteri testuali stampabili*

Esempi di comandi e codici del protocollo



Esempi di comandi:

- **USER** *username*
- **PASS** *password*
- **LIST**
restituisce la lista dei files
presenti nella directory
corrente
- **GET** *filename*
preleva il file dalla macchina
remota
- **PUT** *filename*
invia il file alla macchina
remota

Esempi di codici di stato:

- 331 Username OK,
password required
- 125 data connection
already open;
transfer starting
- 425 Can't open data
connection
- 452 Error writing
file



Cosa è un server FTP

- Non è possibile per un client stabilire una connessione FTP verso una qualsiasi macchina
- Il tipo di paradigma adottato (client/server) presuppone infatti che il server debba essere stato opportunamente configurato per accettare connessioni
- Solitamente, per questioni di sicurezza, le macchine non sono configurate per accettare connessioni di tipo FTP; se si tenta di stabilire una connessione verso una macchina non abilitata la sessione fallisce e nessun trasferimento risulta possibile
- I client FTP sono invece disponibili pressoché su tutti i sistemi operativi

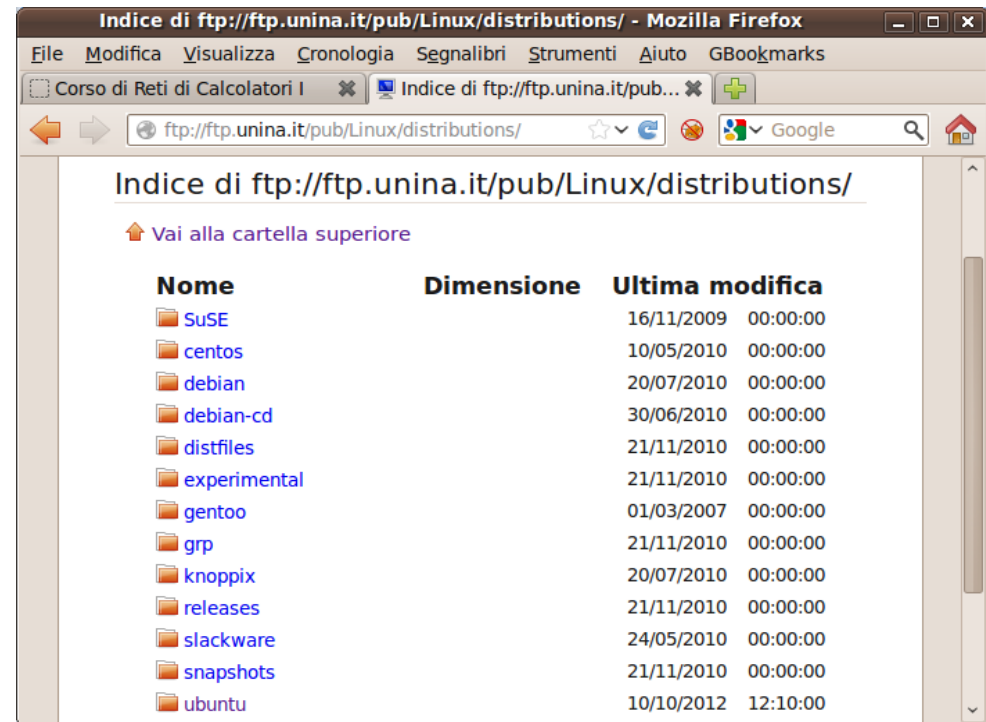
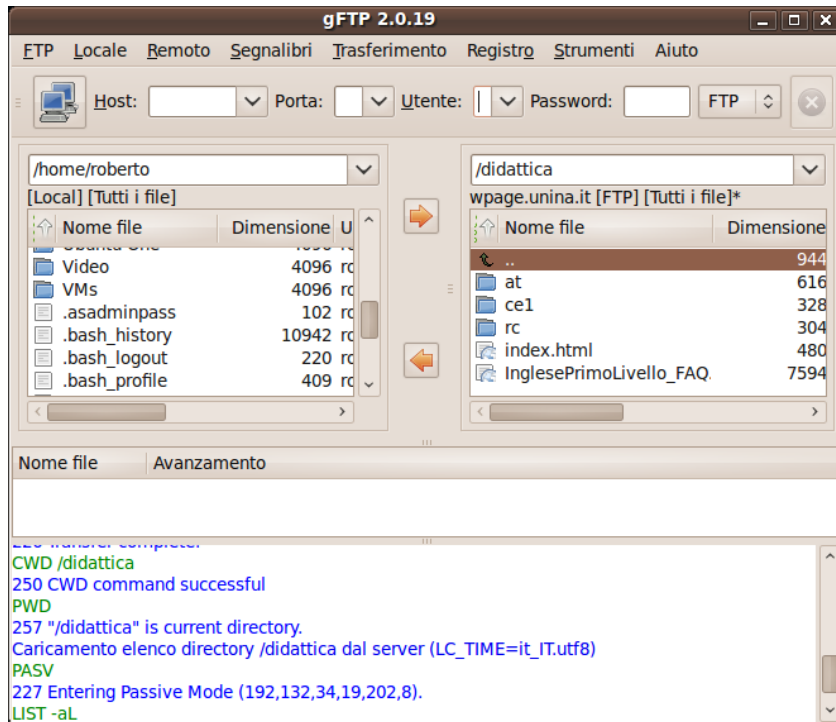


Client ftp da linea di comando

```
roberto@rob-ibmX40: ~
File Modifica Visualizza Terminale Ajuto
roberto@rob-ibmX40:~$ ftp ftp.unina.it
Connected to ftp.unina.it.
220 ftp.unina.it NcFTPD Server (free educational license) ready.
Name (ftp.unina.it:roberto): anonymous
331 Guest login ok, send your complete e-mail address as password.
Password:
230-You are user #4 of 50 simultaneous users allowed.
230-
230 Logged in anonymously.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> hash
Hash mark printing on (1024 bytes/hash mark).
ftp> dir
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
drwxr-xr-x  2 ftpuser  ftpusers   1808 Sep  5  2011 images
-rw-r--r--  1 ftpuser  ftpusers  28319 Sep  6  2011 index.html
-rw-r--r--  1 ftpuser  ftpusers  24855 Jan 14  2011 index2.html
-rw-r--r--  1 ftpuser  ftpusers  28252 Sep  5  2011 indexprova.html
-rw-r--r--  1 ftpuser  ftpusers  28305 Sep  5  2011 indexprova.html-2
drwxr-xr-x 10 ftpuser  ftpusers    360 May  2  2011 pub
226 Listing completed.
ftp> get index2.html
local: index2.html remote: index2.html
200 PORT command successful.
150 Opening BINARY mode data connection for index2.html (24855 bytes).
#####
226 Transfer completed.
24855 bytes received in 0.38 secs (63.9 kB/s)
ftp>
```



Esempi di client ftp grafici





FTP passivo (1/2)

Active FTP

- Il Client apre una connessione di controllo verso l'indirizzo del server sul porto 21 utilizzando un *ephemeral port*
- Il Server apre una connessione dati dal porto 20 verso il client

Passive FTP

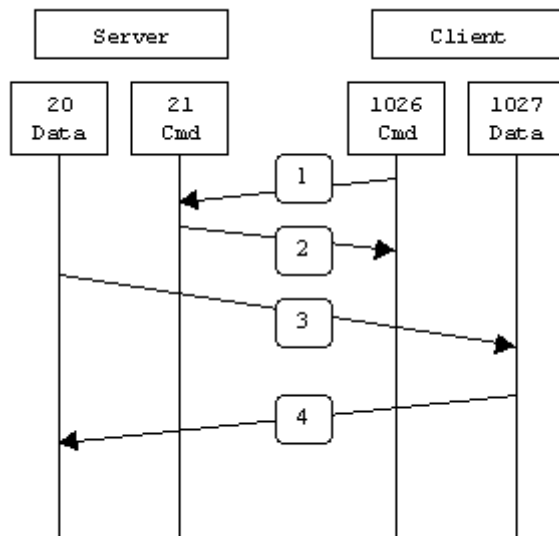
- Il Client apre una connessione di controllo verso l'indirizzo del server sul porto 21 utilizzando un *ephemeral port*
- Il Server sceglie un *ephemeral port* per la connessione dati e la comunica al Client
- Il Client apre la connessione dati sul porto indicato dal server

Nel caso passivo FTP, è il client ad aprire una connessione verso il server, sia per il canale di controllo che per il canale dati.

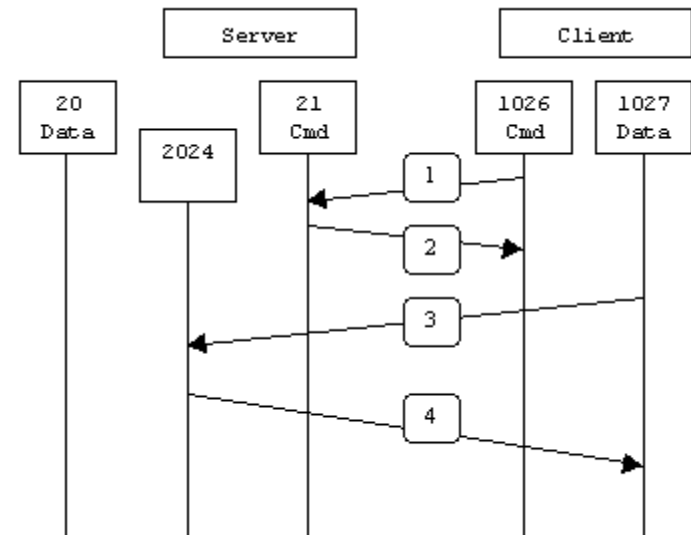


FTP passivo (2/2)

Active FTP



Passive FTP





Esempio di sessione FTP in modo attivo

```
testbox1% ftp -d testbox2
Connected to testbox2.slacksite.com.
220 testbox2.slacksite.com FTP server ready.
Name (testbox2:slacker): slacker
---> USER slacker
331 Password required for slacker.
Password: TmpPass
---> PASS XXXX
230 User slacker logged in.
---> SYST
215 UNIX Type: L8
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
ftp: setsockopt (ignored): Permission denied
---> PORT 192,168,150,80,14,178
200 PORT command successful.
---> LIST
150 Opening ASCII mode data connection for file list.
drwx-----  3 slacker      users          104 Jul 27 01:45 public_html
226 Transfer complete.
ftp> quit
---> QUIT
221 Goodbye.
```

Esempio di sessione FTP in modo passivo



```
testbox1% ftp -d testbox2
Connected to testbox2.slacksite.com.
220 testbox2.slacksite.com FTP server ready.
Name (testbox2:slacker): slacker
---> USER slacker
331 Password required for slacker.
Password: TmpPass
---> PASS XXXX
230 User slacker logged in.
---> SYST
215 UNIX Type: L8
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> passive
Passive mode on.
ftp> ls
ftp: setsockopt (ignored): Permission denied
---> PASV
227 Entering Passive Mode (192,168,150,90,195,149).
---> LIST
150 Opening ASCII mode data connection for file list
drwx----- 3 slacker users 104 Jul 27 01:45 public_html
226 Transfer complete.
ftp> quit
---> QUIT
221 Goodbye.
```



Il protocollo SMTP

- Una volta che una e-mail è stata scritta attraverso l'uso di un programma su un personal computer, è necessario inviarla al destinatario
- Come è noto, il destinatario potrebbe non essere in quel momento disponibile ad accettare messaggi di posta:
 - utente impegnato
 - computer spento
- La posta elettronica sfrutta degli intermediari per il trasferimento delle e-mail tra le parti, alla stregua degli uffici postali che ospitano pacchi nell'attesa che i destinatari passino a ritirarli
- Per trasferire messaggi di posta elettronica tra gli intermediari si utilizza un apposito protocollo
- Si chiama *Simple Mail Transfer Protocol*, definito in RFC821



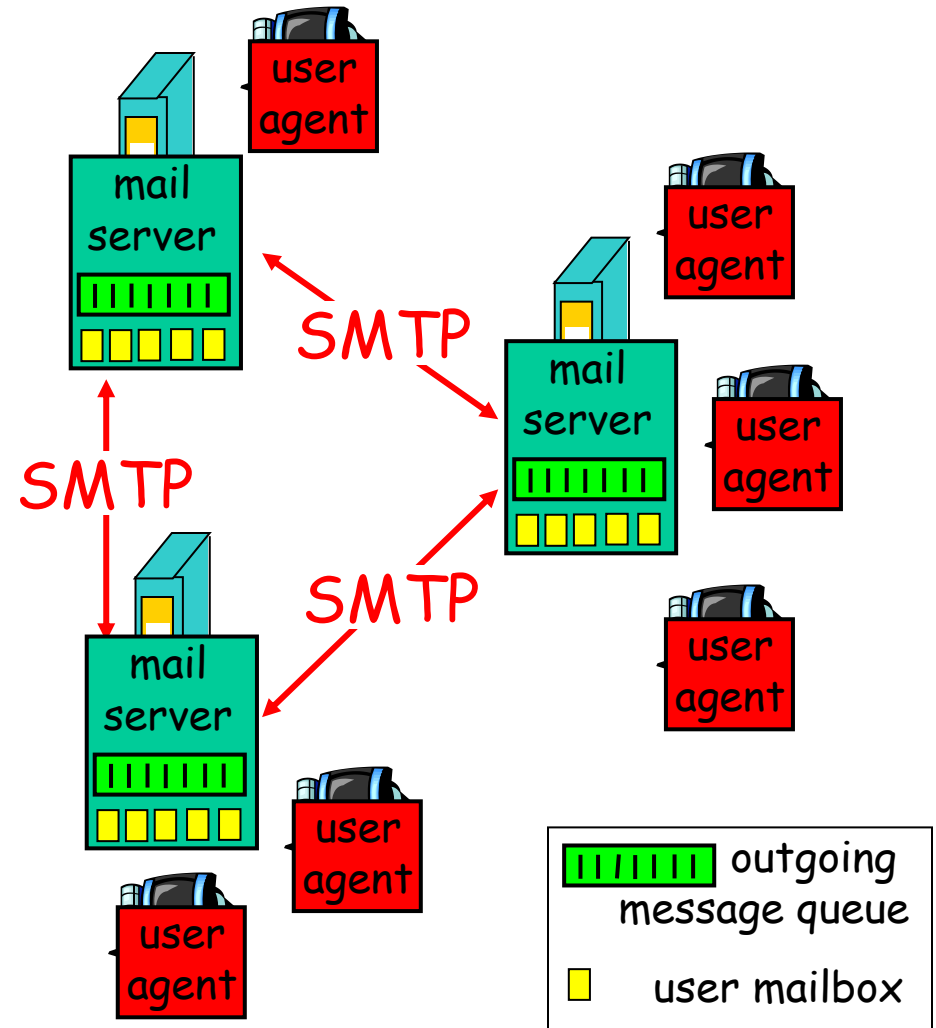
Le entità in gioco

Tre entità principali:

- user agents
- mail servers
- protocollo SMTP

User Agent

- anche detto mail reader
- composizione, modifica, lettura di messaggi
- es.: Eudora, Outlook, Mozilla Thunderbird, Evolution, Kmail...
- messaggi in uscita ed in entrata immagazzinati sul server
- Il protocollo SMTP viene utilizzato anche tra user-agent e server durante l'invio di una mail

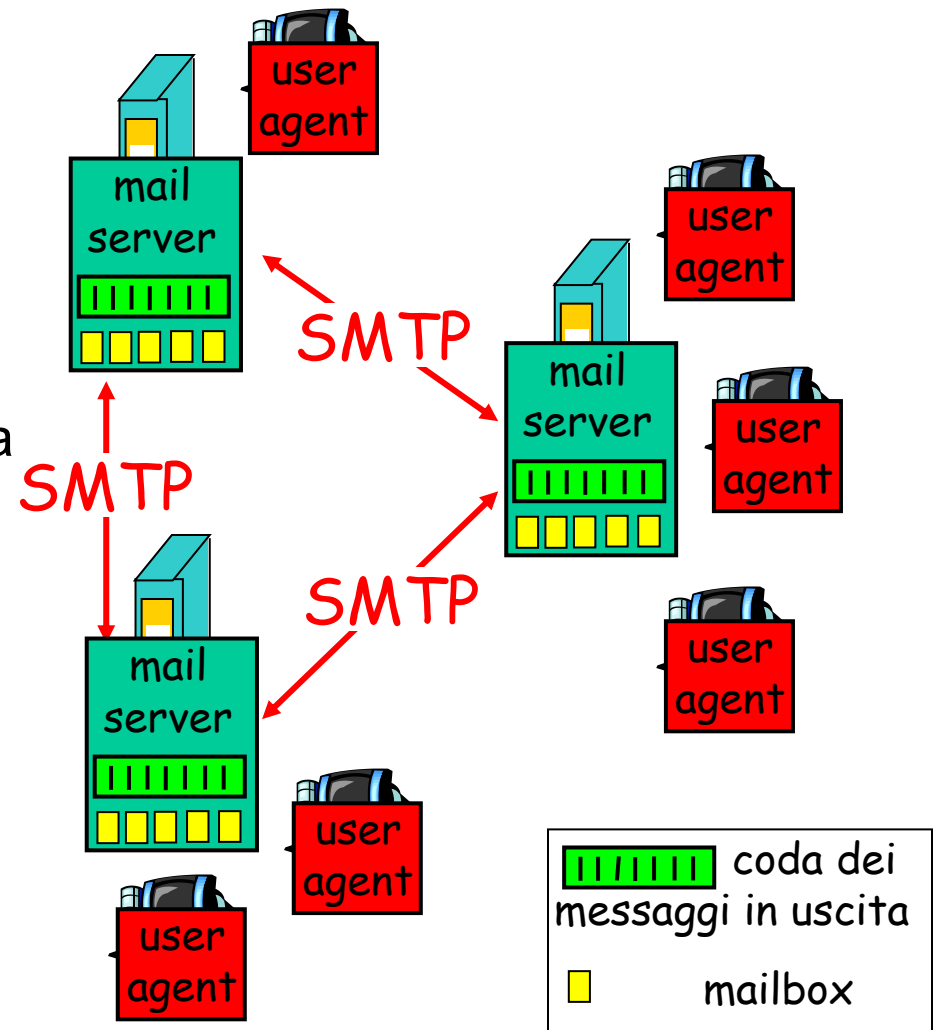




I mail server

Mail Server

- **mailbox** contenente messaggi in entrata (non letti) per l'utente
- **coda dei messaggi in uscita** contenente i messaggi non ancora recapitati
- **protocollo SMTP** per l'interazione tra due mail server.
 - “client”: mail server mittente
 - “server”: mail server destinatario
- Un “mail server” funge in momenti diversi da client o da server a seconda del ruolo che ricopre nello scambio del messaggio





Caratteristiche di SMTP (1)

- Usa il protocollo TCP (porto 25) per consegnare in modo affidabile messaggi dal client al server
- Trasferimento diretto dal server mittente al server destinatario
- Tre fasi durante il trasferimento via SMTP:
 - handshaking (“stretta di mano”)
 - trasferimento del messaggio
 - chiusura della connessione
- Interazione comando/risposta (command/response)
 - **comandi:** testo ASCII
 - **risposta:** codice di stato e descrizione (facoltativa)
- Messaggi codificati con caratteri ASCII a 7-bit



Caratteristiche di SMTP (2)

- Usa una connessione persistente
- Richiede che il messaggio, comprensivo del contenuto, sia codificato in caratteri ASCII a 7 bit
- Alcune combinazioni di caratteri non sono ammesse (p.es., `CRLF.CRLF`). Quando queste combinazioni si presentano il messaggio deve essere opportunamente codificato.
- SMTP usa `CRLF.CRLF` per determinare la fine di un messaggio



Esempio di interazione client → server

```
S: 220 hamburger.edu
C: HELO crepes.fr
S: 250 Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr... Sender ok
C: RCPT TO: <bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Do you like ketchup?
C:   How about pickles?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 hamburger.edu closing connection
```



SMTP in pratica: uso con telnet (1/2)

- **telnet servername 25**
- Si osservi il codice 220 di risposta dal server
- Si inseriscano i comandi `HELO`, `MAIL FROM`, `RCPT TO`, `DATA`, `QUIT`
- In questo modo è possibile inviare un'e-mail senza servirsi dello user agent



SMTP in pratica: uso con telnet (2/2)

```
C:\ Prompt dei comandi
220 smtp1.unina.it ESMTP Sendmail 8.14.0/8.14.0; Wed, 14 Oct 2009 09:10:36 +0200
HELO crepes.fr

250 smtp1.unina.it Hello host57-57-dynamic.53-79-r.retail.telecomitalia.it [79.5
3.57.57], pleased to meet you
500 5.5.1 Command unrecognized: ""
MAIL FROM: <alice@crepes.fr>
250 2.1.0 <alice@crepes.fr>... Sender ok
RCPT TO: <rcanonic@unina.it>
250 2.1.5 <rcanonic@unina.it>... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
ciao
.
250 2.0.0 n9E7AaxZ016057 Message accepted for delivery
QUIT
221 2.0.0 smtp1.unina.it closing connection

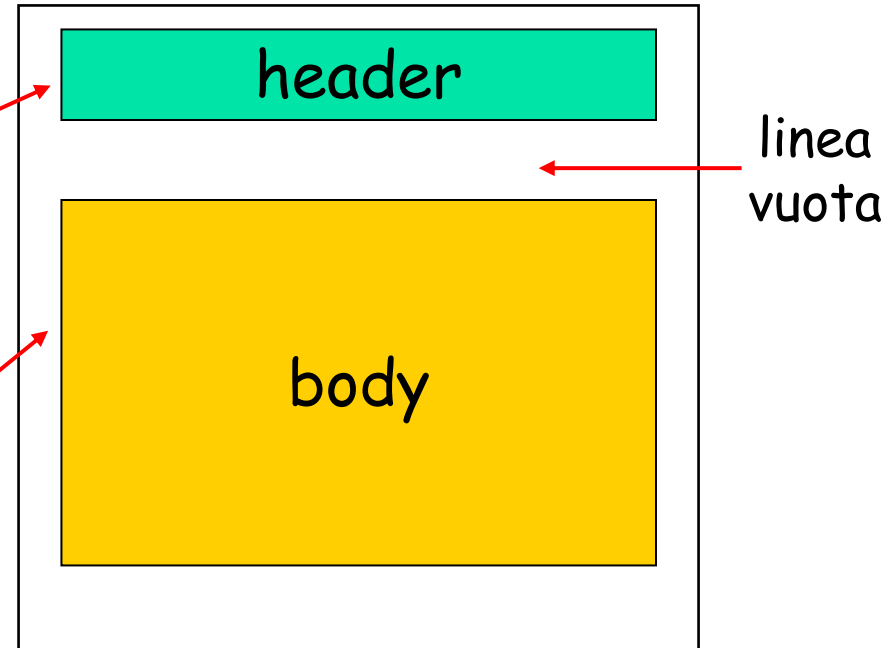
Connessione all'host perduta.

C:\Documents and Settings\User>_
```



Formato del messaggio SMTP

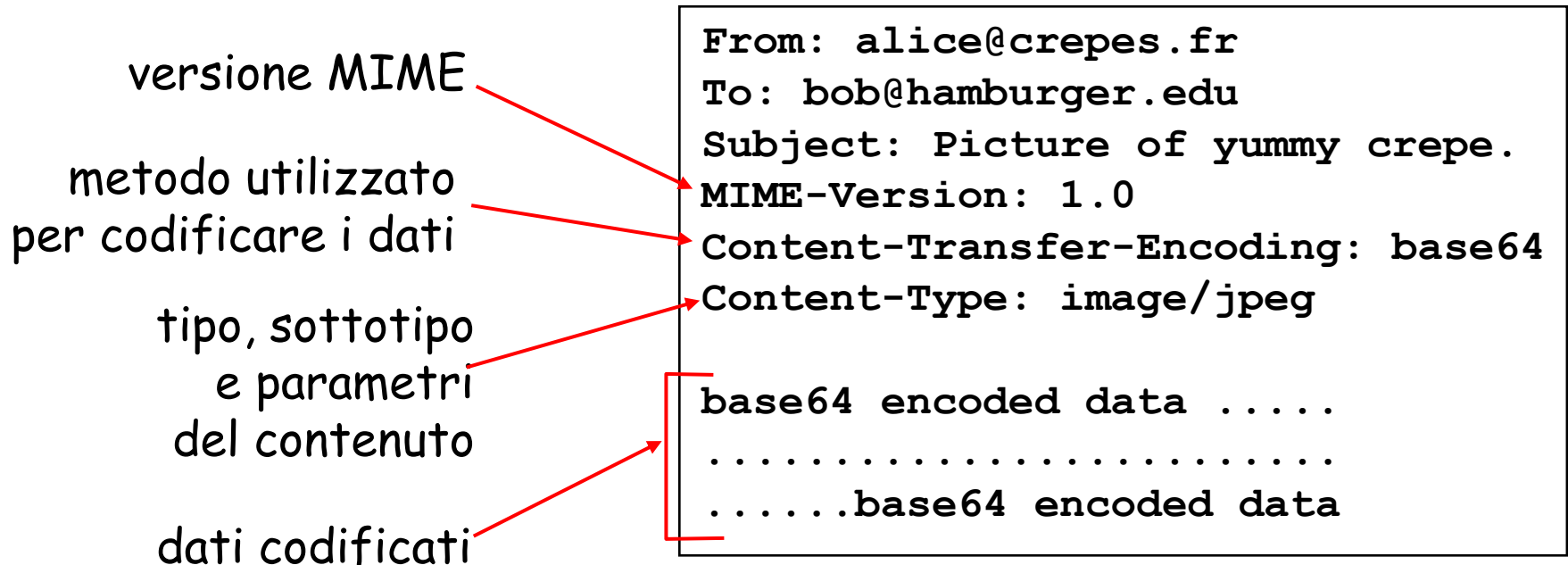
- Linee di intestazione (header):
 - To:
 - From:
 - Subject:
 - ...*differenti da comandi smtp!*
- corpo (body):
 - il “messaggio” vero e proprio
 - solo caratteri ASCII





L'estensione MIME

- MIME: Multipurpose Internet Mail Extensions, RFC 2045-2056
- righe aggiuntive nell'intestazione informano della presenza di un body MIME





Messaggi SMTP: un esempio (sniffer Ethereal)

14 5.022557 143.225.229.172 192.132.34.73 SMTP Response: 334 ENTER MAIL, END WITH . ON A LINE BY

14 5.022557 143.225.229.172 192.132.34.73 SMTP Message Body

Internet Protocol, Src Addr: 143.225.229.172 (143.225.229.172), Dst Addr: 192.132.34.73 (192.132.34.73)
Transmission Control Protocol, Src Port: 3419 (3419), Dst Port: smtp (25), Seq: 1829128306, Ack: 2648952608, Len: 1
Simple Mail Transfer Protocol

Message: Message-ID: <4055A84A.9060100@unina.it>\r\n
Message: Date: Mon, 15 Mar 2004 13:57:46 +0100\r\n
Message: From: Simon Pietro Romano <spromano@unina.it>\r\n
Message: User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.0.2) Gecko/20030208 Netscape/7.02\r\n
Message: X-Accept-Language: en-us, en\r\n
Message: MIME-Version: 1.0\r\n
Message: To: spromano <spromano@unina.it>\r\n
Message: Subject: prova SMTP da Netscape agent\r\n
Message: Content-Type: text/plain; charset=ISO-8859-1; format=flowed\r\n
Message: Content-Transfer-Encoding: 8bit\r\n
Message: \r\n
Message: Ciao.\r\n
Message: \r\n
Message: SP\r\n
Message: \r\n
Message: -- \r\n
Message: _\\|//_\r\n
Message: (0-0)\r\n
Message: ~~~~~000~()~000~~~~~\r\n
Message: Simon Pietro Romano\r\n
Message: Universita' di Napoli Federico II\r\n
Message: Computer Science Department \r\n
Message: Phone: +39 081 7683823 -- Fax: +39 081 7684219\r\n
Message: e-mail: spromano@unina.it\r\n
Message: http://www.grid.unina.it/individual/sprom\r\n
Message: \r\n
Message: idioti. Ci rifletto un istante; e mi scoraggio>>. Magritte.\r\n
Message: oooo\r\n
Message: ~~~~~()~ oooo~~~~~\r\n
Message: \ () \r\n
Message: \) / \r\n
Message: \ / \r\n
Message: \r\n
Message: \r\n

0190 74 0d 0a 43 6f 6e 74 65 6e 74 2d 54 79 70 65 3a t..Content-Type:
01a0 20 74 65 78 74 2f 70 6c 61 69 6e 3b 20 63 68 61 text/plain; cha
01b0 72 73 65 74 3d 49 53 4f 2d 38 38 35 39 2d 31 3b rset=ISO-8859-1;
01c0 20 66 6f 72 6d 61 74 3d 66 6c 6f 77 65 64 0d 0a format=flowed..

Filter: [] [Reset] [Apply]



Esempi di tipi MIME

Content-Type: type/subtype; parameters

Text

- sottotipi: **plain**, **html**

Image

- sottotipi: **jpeg**, **gif**

Audio

- sottotipi:
basic (8-bit mu-law encoded), **32kadpcm** (32 kbps coding)

Video

- sottotipi: **mpeg**,
quicktime

Application

- altri dati che devono essere processati da specifiche applicazioni
- sottotipi: **msword**,
octet-stream



Esempio di mail “multiparte”

```
From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Picture of yummy crepe.
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=98766789
```

```
--98766789
```

```
Content-Transfer-Encoding: quoted-printable
Content-Type: text/plain
```

```
Dear Bob,
Please find a picture of a crepe.
```

```
--98766789
```

```
Content-Transfer-Encoding: base64
Content-Type: image/jpeg
```

```
base64 encoded data .....
.....
.....base64 encoded data
--98766789--
```

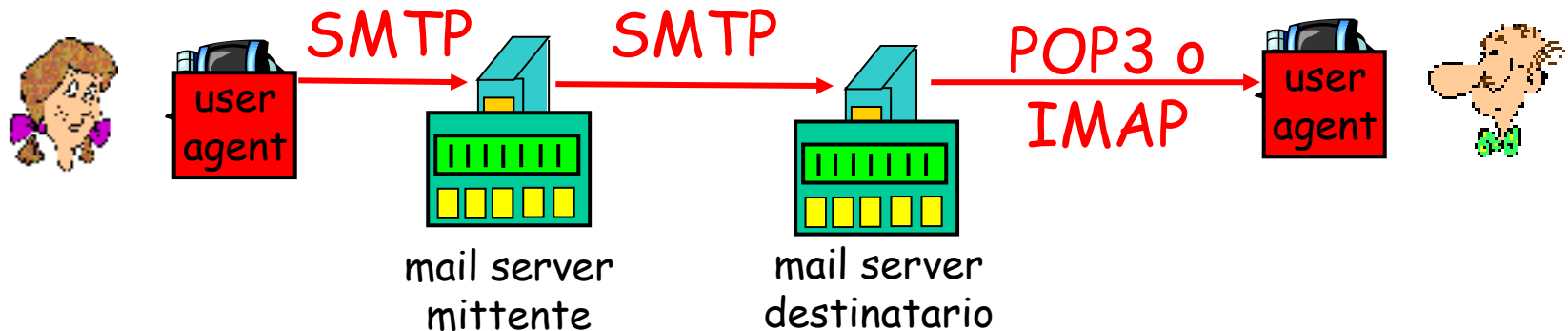
Prelievo della posta: Post Office Protocol (POP3)



- Fino ad ora abbiamo visto come sia possibile trasferire messaggi tra i vari mail server
- Non abbiamo però ancora parlato di come un utente possa, in un momento qualsiasi, accedere alla propria casella di posta elettronica per leggere i propri messaggi
- Per questa operazione è previsto un ulteriore protocollo
- Esso è chiamato POP3 (Post Office Protocol – versione 3) ed è definito in RFC 1939
- Si tratta sempre di un protocollo client server:
 - lo user agent ancora una volta gioca il ruolo di client POP
 - il mail server gioca il ruolo di server POP



La catena dei protocolli per la posta



- SMTP: consegna di messaggi
- Protocolli di accesso alla mail: recupero dei messaggi dai server
 - POP: Post Office Protocol
 - autorizzazione (agent \leftrightarrow server) e download
 - IMAP: Internet Mail Access Protocol [RFC 2060]
 - più complicato e potente
 - manipolazione avanzata dei messaggi sul server
 - HTTP: gmail, Hotmail, Yahoo! Mail, ecc.



Esempio di dialogo POP3

autorizzazione

- comandi del client:
 - user**: specifica la username
 - pass**: specifica la password
- il server risponde
 - +OK
 - ERR

fase di scambio

- comandi del client:
 - list**: visualizza la lista dei messaggi
 - retr**: preleva il messaggio per numero
 - dele**: elimina il messaggio dal server
 - quit**: chiude la sessione

```
S: +OK POP3 server ready
C: user alice
S: +OK
C: pass hungry
S: +OK user successfully logged on

C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: <message 1 contents>
S: .
C: dele 1
C: retr 2
S: <message 1 contents>
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off
```



POP3 in pratica (1/2): telnet cds.unina.it 110

 Telnet cds.unina.it

```
+OK cds.unina.it Cyrus POP3 v2.1.10 server ready
user spromano
+OK Name is a valid mailbox
pass [REDACTED]
+OK Maildrop locked and ready
list_
```

...

Qui c'era la password!



POP3 in pratica (2/2): telnet cds.unina.it 110

```
6292 1867
6293 2683
6294 913
.
retr 6294
+OK Message follows
Return-Path: <spromano@unina.it>
Received: from cds.unina.it ([unix socket])
        by cds.unina.it (Cyrus v2.1.10) with LMTP; Mon, 15 Mar 2004 13:13:33 +0100
X-Sieve: CMU Sieve 2.2
Received: from mail.unina.it (mail.unina.it [192.132.34.73])
        by cds.unina.it (8.12.11/8.12.9) with ESMTP id i2FCDX6G016069
        for <spromano@unina.it>; Mon, 15 Mar 2004 13:13:33 +0100 (CET)
Received: (from root@localhost)
        by mail.unina.it (8.12.11/8.12.11) id i2FCBCfe021235
        for spromano@unina.it; Mon, 15 Mar 2004 13:11:12 +0100
Received: from grid.unina.it ([143.225.229.172])
        by mail.unina.it (8.12.11/8.12.11) with SMTP id i2FC5bZc018823
        for spromano@unina.it; Mon, 15 Mar 2004 13:10:24 +0100
Date: Mon, 15 Mar 2004 13:10:24 +0100
From: spromano@unina.it
Message-Id: <200403151210.i2FC5bZc018823@mail.unina.it>
X-scanner: scanned by Inflex 1.0.12.7

Ciao giovane, come stai?
Fammi sapere asap...
.
```



L'accesso alla posta via WEB

- Molti siti web forniscono accesso alle proprie caselle di posta (gmail, hotmail, Yahoo!, ecc.)
- In questo caso non serve avere uno user agent installato e correttamente configurato per ricevere ed inviare posta.
- È sufficiente disporre di un qualsiasi browser

