

# **RETI DI CALCOLATORI**

## **Autunno 2018**

**docente: Laura Ricci**

**Lezione 16:**

# **SERVIZI DI POSTA ELETTRONICA**

**14/12/2018**

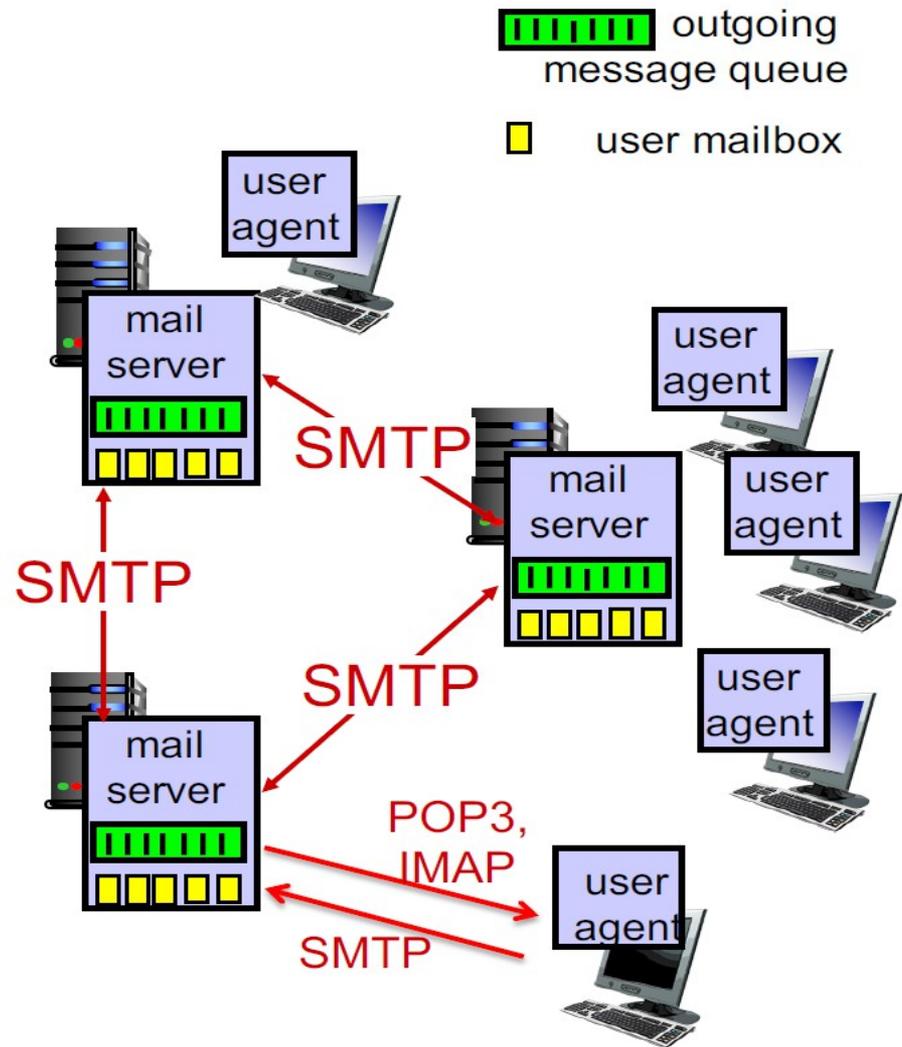
Forouzan

- paragrafo 2.3.3

- Simple Mail Transfer Protocol (SMTP) RFC 5321
- Post Office Protocol (POP3) RFC 1939
- Internet Mail Control Protocol (IMAP) RFC 3501

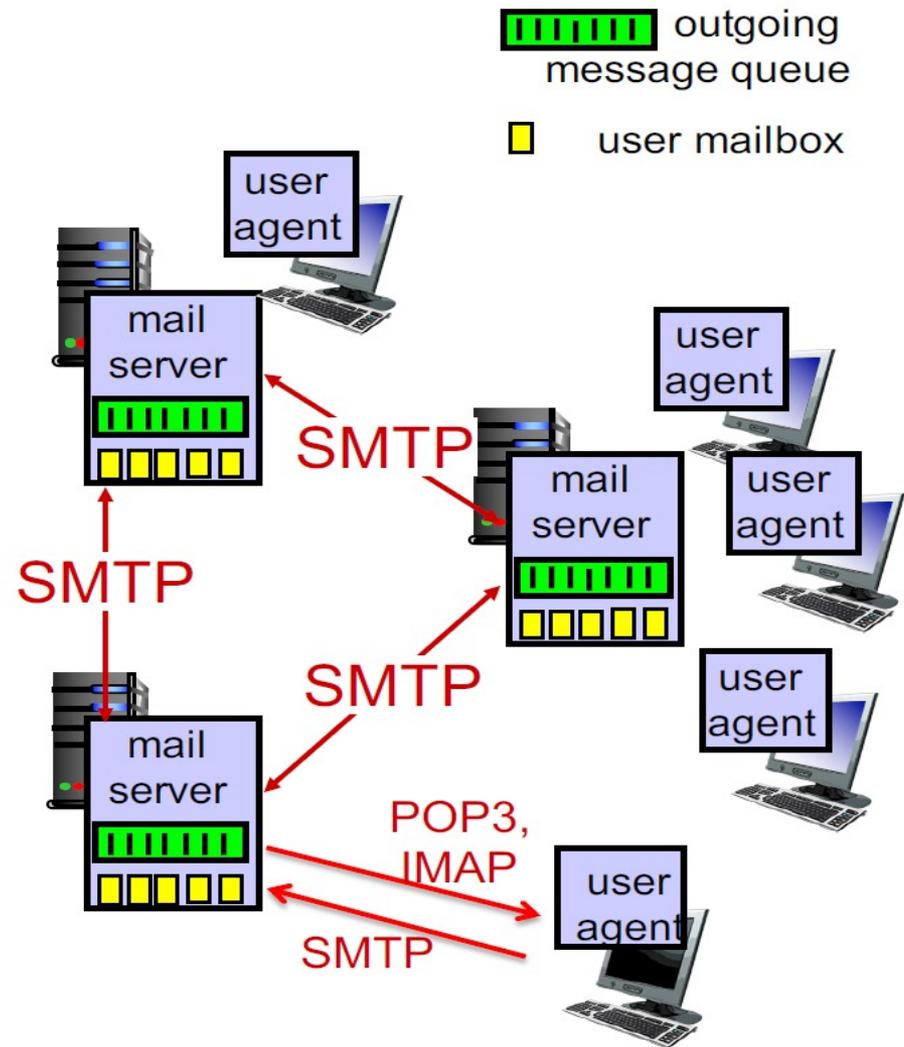
# SERVIZI DI POSTA ELETTRONICA

- Client d'utente: User Agent (OutLook, Thunderbird, etc.)
- Mail Server
- Simple Mail Transfer Protocol SMTP: per trasferire email dal client d'utente fino al mail server del destinatario
- protocolli di accesso al mail server: per “scaricare” email dal proprio mail server (POP3, IMAP)



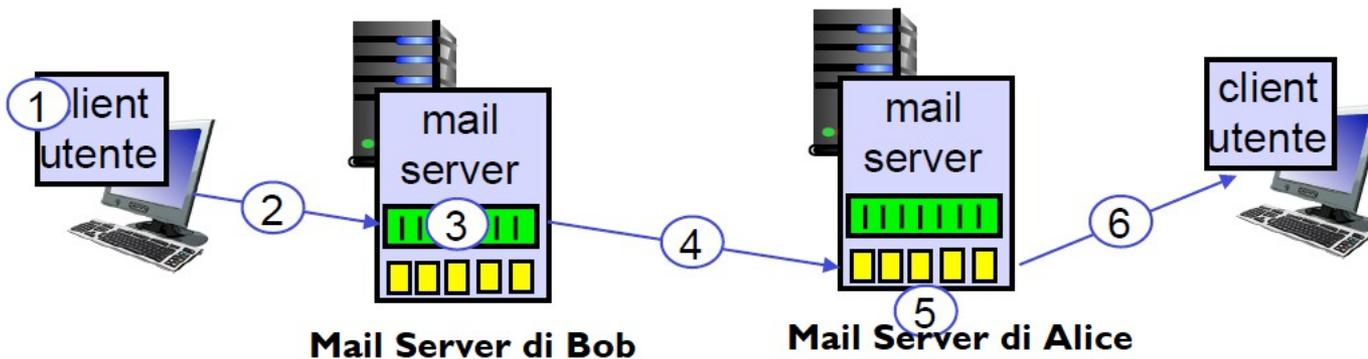
# I MAIL SERVER

- i mail server contengono per ogni client controllato:
  - una coda di email in ingresso (mailbox)
  - una coda di email in uscita
- i mail server
  - ricevono le mail in uscita da tutti i client d'utente che "controllano"
  - ricevono da altri mail server tutte le mail destinate ai client d'utente controllati
- i mail server "parlano"
  - **SMTP** con altri mail server e con i client d'utente in uplink
  - **POP3/IMAP** con i client d'utente in downlink



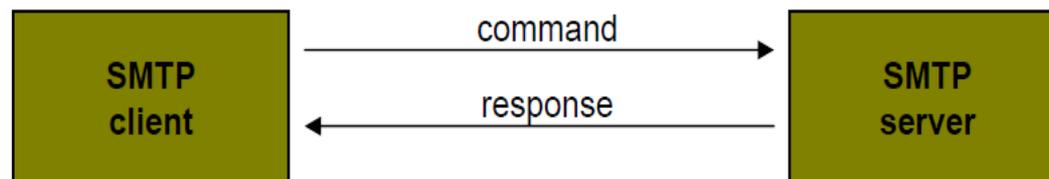
# ESEMPIO DI TRASFERIMENTO SMTP

1. Bob compone una email destinata ad Alice: [Alice@miomailserver.com](mailto:Alice@miomailserver.com)
2. Il client d'utente di Bob invia la mail al proprio mail server
3. Il mail server di Bob si comporta come client SMTP ed apre una connessione TCP (porta 25) con il mail server di Alice
4. Il client SMTP (mail server di Bob) invia la email sulla connessione TCP
5. Il mail server di Alice memorizza la mail nella mailbox di Alice
6. Alice (in modo asincrono) usa il proprio client d'utente per leggere la mail



# SMTP

- E' un protocollo applicativo **client-server** che utilizza TCP: apre una connessione TCP con la porta 25 del mail server del destinatario
  - TCP handshaking
  - setting della connessione SMTP
  - trasferimento del messaggio
  - terminazione della connessione SMTP
  - chiude la connessione TCP
- l'interazione tra client SMTP e server SMTP è di tipo **comando/risposta**
- i comandi vengono inviati dal client SMTP al server SMTP: ogni comando consiste in una keyword seguita da uno o più argomenti
- le risposte sono inviate dal server SMTP al client SMTP. Ogni risposta è un numero di 3 bit, che può essere seguito da altra informazione testuale



# COMANDI SMTP

Keyword	Argument(s)
<b>HELLO</b>	Sender's host name.
<b>MAIL FROM</b>	Sender of the message.
<b>RCPT TO</b>	Intended recipient of the message.
<b>DATA</b>	Body of the mail.
<b>QUIT</b>	meaning: terminate message
<b>RSET</b>	meaning: abort the current mail transactions
<b>YRFY</b>	Name of recipient to be verified.
<b>NOOP</b>	meaning: check the status of recipient
<b>EXPN</b>	Mailing list to be expanded.
<b>HELP</b>	Command name.
<b>SEND FROM</b>	Intended recipient of the message.
<b>SMOL FROM</b>	Intended recipient of the message.
<b>SMAL FROM</b>	Intended recipient of the message.

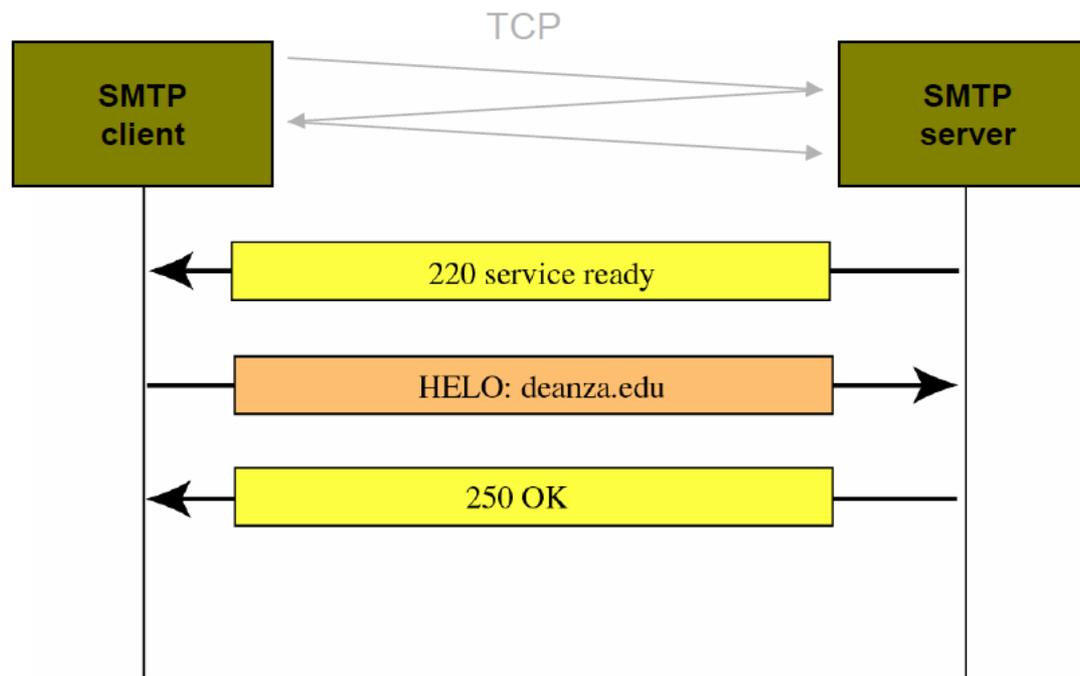
# RISPOSTE SMTP

- codice di 3 cifre che può essere seguito da ulteriore informazione testuale
  - **2yz** – il comando è stato eseguito correttamente ed è possibile emettere un nuovo comando
  - **3yz** – il comando è accettato, ma il destinatario ha bisogno di ulteriore informazione per completarlo
  - **4yz** – il comando richiesto è respinto, ma la condizione di errore è temporanea
  - **5yz** – il comando richiesto è respinto e deve essere inviato nuovamente

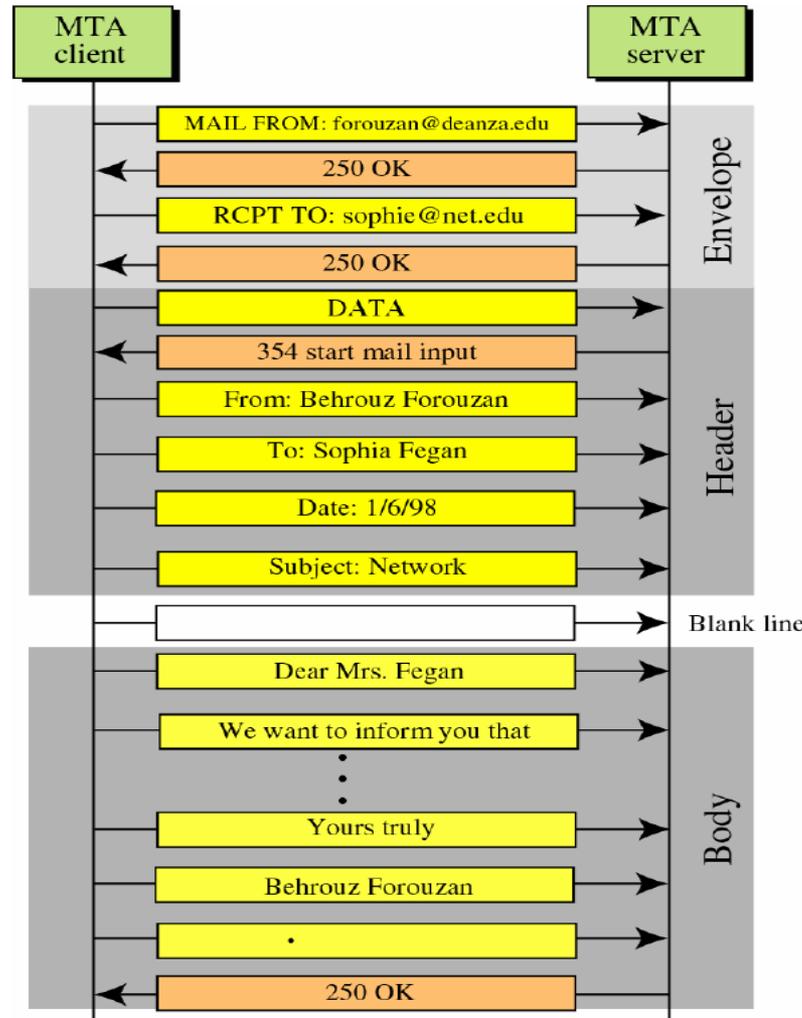
Code	Description
<b>220</b>	Service ready.
<b>221</b>	Service closing transmission channel.
<b>354</b>	Start mail input.
<b>421</b>	Service not available.
<b>450</b>	Mailbox not available.
<b>500</b>	Syntax error; unrecognized command.
<b>501</b>	Syntax error in parameters or arguments.

# SMTP: STABILIRE UNA CONNESSIONE

- il client apre una connessione TCP con il ricevente
- il server invia il codice 220 (service ready) per indicare al client che è pronto a ricevere una mail. Se il server non è pronto, invia il codice 421.
- il client invia il messaggio HELO per identificarsi usando il suo nome di dominio
- il server risponde con il codice 2xx o con un altro codice a seconda della situazione

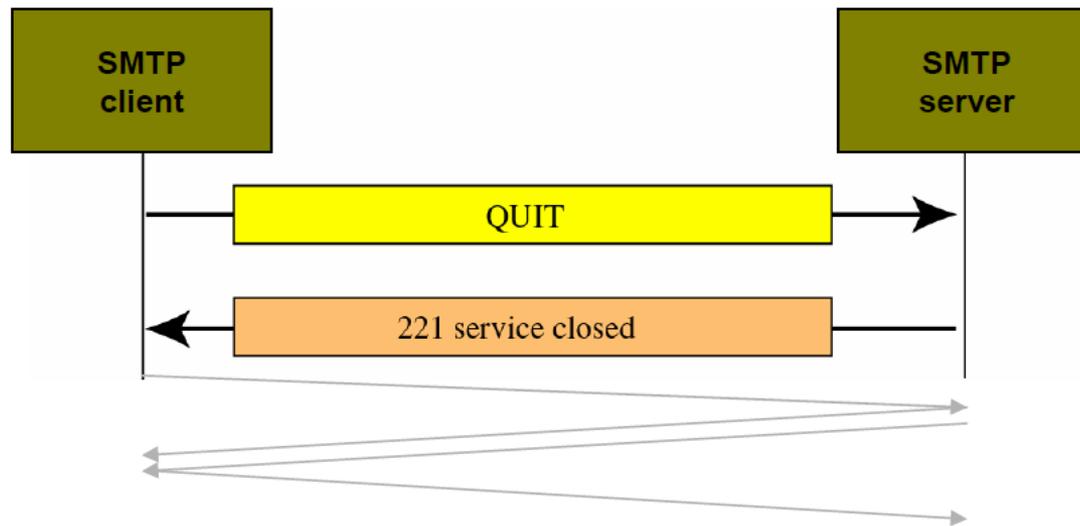


# SMTP: TRASFERIMENTO DEL MESSAGGIO



# TERMINAZIONE CONNESSIONE SMTP

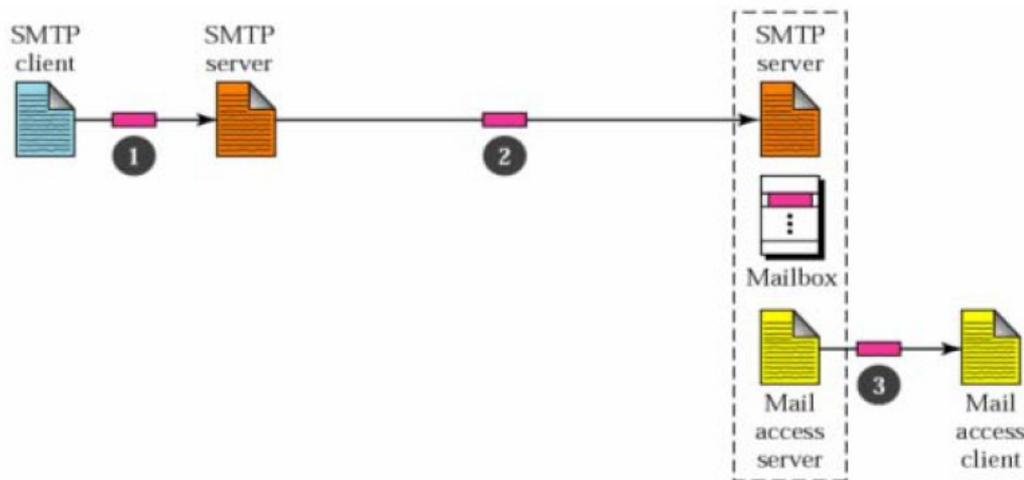
- il client invia il comando QUIT
- il server risponde con il codice 221
- il client inizia la chiusura della connessione TCP



If client has several messages to send, they all can be sent over the same TCP connection.

# SMTP: PUSH PROTOCOL

- SMTP effettua un “push” del messaggio dal mittente al destinatario anche se il destinatario non è disposto, per il momento, a riceverlo.
- la connessione TCP-SMTP è iniziata da parte del mittente, non del destinatario
  - il destinatario può non essere online al momento dell'arrivo della e-mail
- HTTP è un esempio di “pull” protocol: la connessione TCP è iniziata dal client interessato nel ritrovare (pulling) alcuni dati

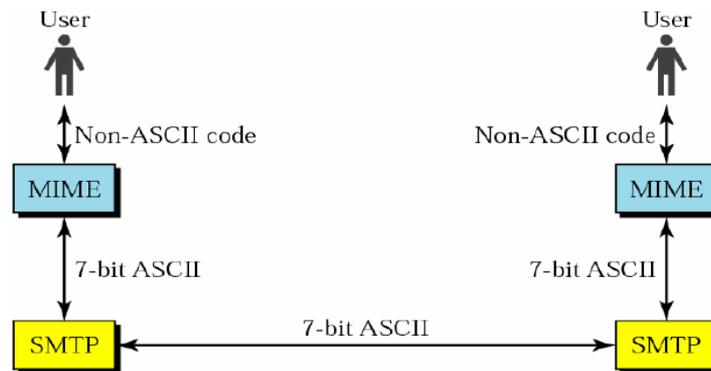


# SMTP: LIMITAZIONI

- i messaggi possono essere inviati solo in formato ASCII a 7-bit
- non possono essere usati per dati audio o video, o per linguaggi non supportato da caratteri ASCII (cinese, giapponese, etc,..)

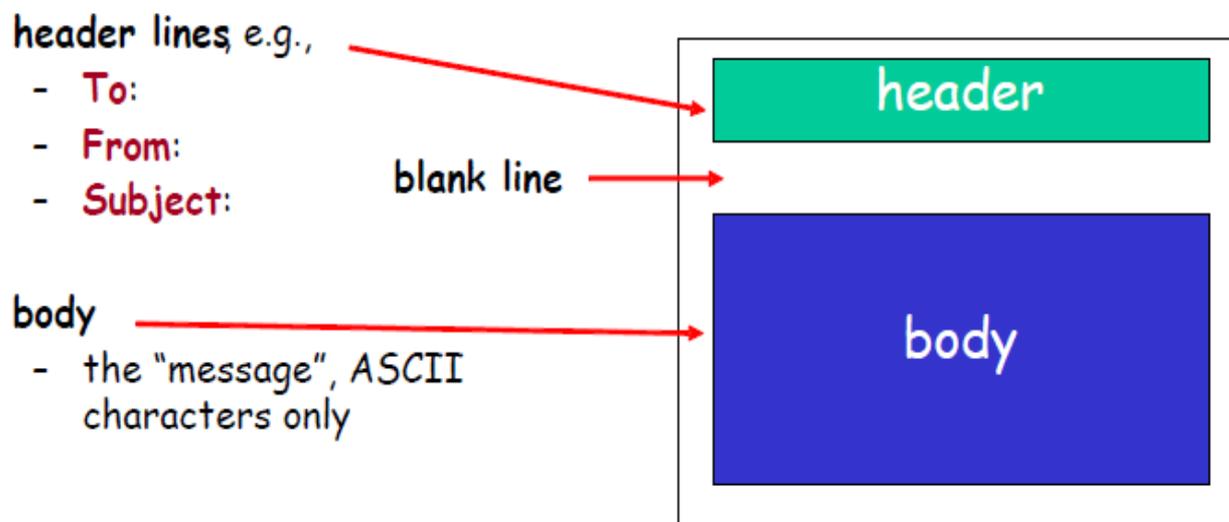
## Multipurpose Internet Mail Extensions (MIME)

- protocollo supplementare che permette di inviare messaggi non ASCII attraverso una connessione SMTP
- non un nuovo protocollo, solo una estensione di SMTP
- mittente: MIME trasforma dati non-ASCII in ASCII e li consegna al client SMTP
- destinatario: il server SMTP riceve i dati ASCII e li consegna al MIME affinché li trasformi nuovamente nel formato originario



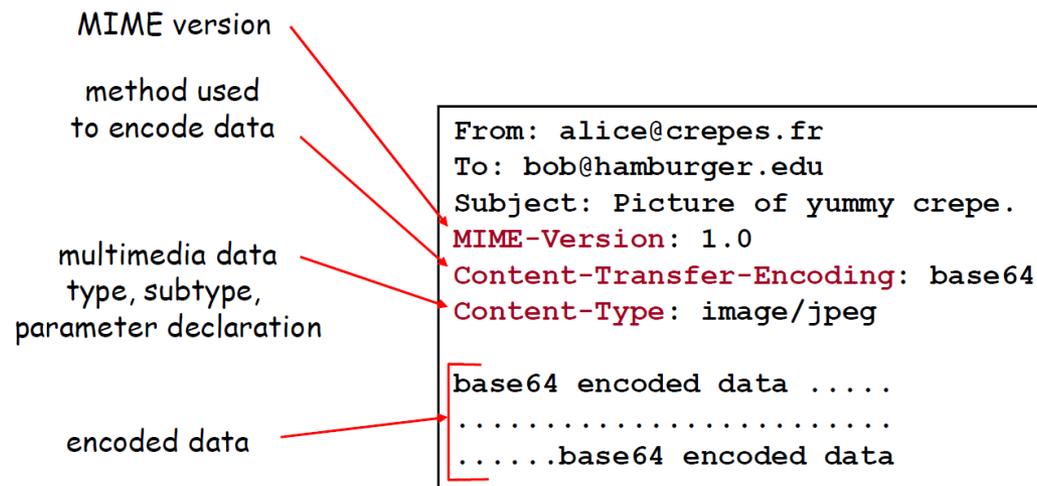
# SMTP: LIMITAZIONI

- RFC 822: Standard utilizzato per costruire messaggi da trasmettere via SMTP
  - messaggi = header + blank line + body
  - Header contiene le informazioni necessarie per la trasmissione e la consegna
  - Body contiene l'oggetto che deve essere consegnato al destinatario



# SMTP: LIMITAZIONI

- RFC 2045 and 2046: estensione a RFC 822: definisce altre MIME-related header-lines richieste per inviare contenuto non-ASCII
  - **MIME-Version**: versione corrente 1.1
  - **Content-Transfer-Encoding**: definisce il metodo per codificare il messaggio (e.g. 7-bit, 8-bit, binary, Base64, Quoted-printable)
  - **Content-Type**: definisce il tipo di dato trasportato nel contenuto
    - (e.g. plain, JPEG, GIF, MPEG): permette al client di prendere l'azione appropriata (ad esempio decomprimere/visualizzare una immagine JPEG)



# IL FORMATO DELLE E-MAIL (RFC 822)

- MIME consente anche il trasferimento di più oggetti come parti di uno stesso messaggio:

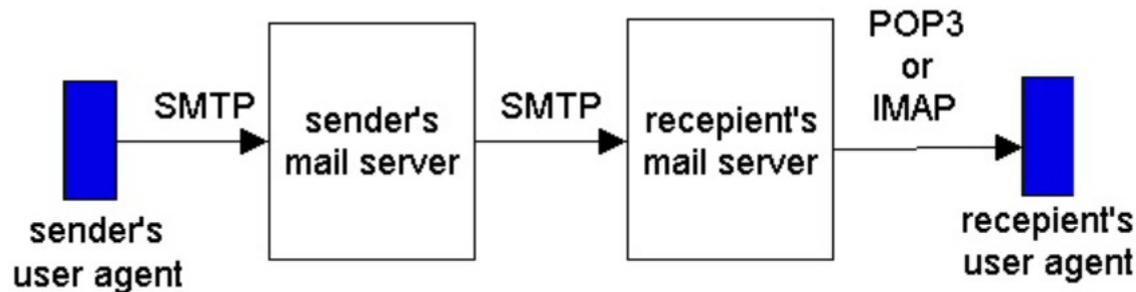
```
From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Picture of yummy crepe with commentary
MIME-Version: 1.0
Content-Type: multipart/mixed; Boundary=StartOfNextPart
--StartOfNextPart
Dear Bob,
Please find a picture of an absolutely scrumptious crepe.

--StartOfNextPart
Content-Transfer-Encoding: base64
Content-Type: image/jpeg
base64 encoded data .....

--StartOfNextPart
Let me know if you would like the recipe.
.
```

# PROTOCOLLI DI ACCESSO ALLE MAIL

- Diversi protocolli sono stati sviluppati per il colloquio tra *user agent* e *server* in fase di lettura dei messaggi presenti nel *mailbox*
  - POP3 (*Post Office Protocol*, RFC 1939): download dei messaggi
  - IMAP (*Internet Mail Access Protocol*, RFC 1730): download dei messaggi, gestione della *mailbox* sul *mail server*



# IL PROTOCOLLO POP

- lo user agent apre una connessione TCP con il mail server sulla porta 110
- POP prevede 3 fasi: autenticazione, transazione, update
- Fase 1: lo user agent invia l'user name e la password per l'autenticazione
- Fase 2: l'utente reperisce i messaggi: può anche ritrovare statistiche o marcare messaggi da cancellare
- Fase 3: l'utente invia un messaggio di QUIT che chiude la sessione POP: il server cancella i messaggi marcati

