



Livello applicazione:  
La posta elettronica ed i protocolli SMTP e POP3  
Trasferimento di file ed il protocollo FTP

Ing. Nadia Rinaldo

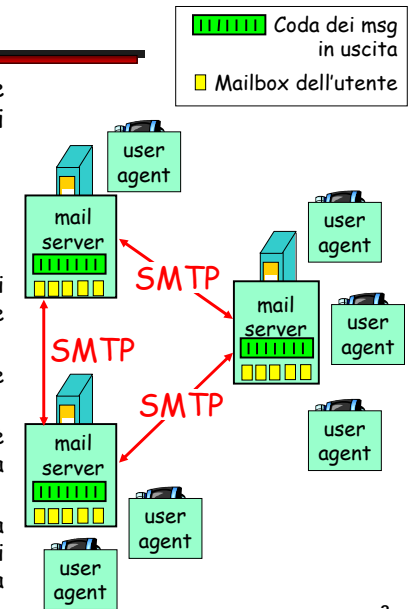
1



## La posta elettronica

Il sistema di posta elettronica è realizzato mediante tre componenti fondamentali:

- Gli **user agent**
  - I **mail server**
  - Il **protocollo SMTP**
- Gli **user agent** sono impiegati dagli utenti per leggere la posta, comporre e richiedere la spedizione di nuovi messaggi
- Eudora, MS Outlook, Netscape Messenger
- Il messaggio costruito dallo user agent è inviato al **mail server** dove è accodato nella coda dei messaggi in uscita
- La lettura di un messaggio è realizzata usando uno user agent e recuperando i messaggi da una coda (**mailbox**) specifica dell'utente gestita dal mail server



2



## Le code di ingresso e di uscita (1)

- I mail server costituiscono il nucleo del sistema di posta elettronica
- Un mail server gestisce i messaggi di un insieme di utenti
- Ogni utente possiede una coda di messaggi riservata (detta **mailbox**) sul proprio mail server
- Una mailbox memorizza i messaggi che asincronamente arrivano ad uno specifico utente
- I messaggi rimangono nella mailbox finchè non sono rimossi esplicitamente dallo user agent a seguito di operazioni di lettura o di cancellazione (messaggi non ancora letti)
- Un mail server gestisce anche una **coda di uscita** nella quale sono presenti tutti i messaggi consegnati dagli user agent degli utenti e che devono essere consegnati ad altri utenti sulla rete (messaggi non ancora spediti)
- Un mail server gestisce **più mailbox** ed **un'unica coda di uscita**

3



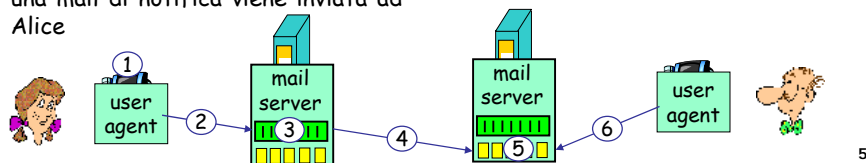
## Le code di ingresso e di uscita (2)

- Nella coda di uscita è possibile che vi siano messaggi di utenti che non possiedono una mailbox su quel mail server
- In tal caso il mail server estrae i messaggi dalla coda di uscita e tenta di spedirli ai mail server degli utenti destinatari
  - Se la spedizione non riesce, il messaggio viene nuovamente accodato per essere spedito successivamente
  - Dopo un numero (fissato in fase di configurazione) di tentativi falliti, il mail server rimuove il messaggio dalla coda di uscita ed invia un messaggio di avviso all'utente che aveva tentato la spedizione del messaggio
- Il protocollo SMTP è il protocollo utilizzato per inviare un messaggio verso un mail server
- In una interazione SMTP il client è l'entità che spedisce il messaggio, mentre il server è l'entità che riceve il messaggio
  - Il client esegue sul mail server del mittente ed il server esegue sul mail server del ricevente

4

## Recapito di un messaggio di posta elettronica

- (1) Alice compone un messaggio e lo inoltra al suo Mail Server
- (2) Il Mail Server di Alice dispone il messaggio nella coda di messaggi in uscita
- (3) Il Mail Server di Alice apre una connessione SMTP con il Mail Server di Bob ed inoltra il messaggio
- Se il contatto fallisce, l'invio è ripetuto ogni trenta minuti
- Se l'invio fallisce per diversi giorni, una mail di notifica viene inviata ad Alice
- (5) Mail Server di Bob riceve il messaggio dal Mail Server di Alice e lo salva nella Mailbox di Bob
- (6) Bob accede alla propria mailbox specificando Username e Password
- I messaggi possono essere trasferiti dalla mailbox all'host da cui Bob ha acceduto alla mailbox e/o lasciati sul server
- Bob legge il messaggio di Alice



5

## Il protocollo SMTP (1)

- Sebbene esistano altri protocolli per la gestione della posta elettronica, il protocollo SMTP (Simple Mail Transfer Protocol) è quello più importante (RFC 821) (anni '80)
- Come l'HTTP, l'SMTP impiega al livello del trasporto il protocollo TCP per avere garanzie di affidabilità (la porta di default è 25)
- Un mail server può fungere sia da server che da client di interazioni SMTP
  - client quando spedisce, server quando riceve
- Di conseguenza l'SMTP impiega un'unica applicazione che funziona da client e da server a seconda dei casi (componenti client e server integrate)
- Se il mittente ed il destinatario sono connessi allo stesso servizio di trasporto, la connessione è diretta tra il mail server del mittente e quello del destinatario (non sono usati mail server intermedi)
- Se il mittente ed il destinatario non sono connessi allo stesso servizio di trasporto, i messaggi attraversano server intermedi, detti **relay SMTP server** che fungono da gateway

6

## Il protocollo SMTP (2)

- I messaggi inviati con il protocollo SMTP sono in formato testuale, ovvero stringhe ASCII a 7 bit (ogni linea non deve superare 1000 caratteri)
- Le informazioni multimediali sono trasferite usando una duplice conversione: da binario a carattere e viceversa
- Un SMTP client apre una connessione TCP verso un SMTP server
- Una volta che la connessione è stata stabilita la comunicazione avviene in tre fasi:
  - **Handshaking** (saluto)
    - Il client ed il mittente si scambiano informazioni di handshaking
  - **Trasferimento di uno o più messaggi** (connessione permanente)
    - Il client indica l'indirizzo e-mail del mittente e di colui che riceverà il messaggio
    - Se il destinatario esiste sul mail server contattato, inizia la trasmissione effettiva del messaggio sul canale TCP affidabile, altrimenti viene rilasciata la connessione
    - Il client impiega **connessioni persistenti**: un client chiude la connessione solo dopo aver inviato tutti i messaggi in coda destinati al server con cui è stabilita la connessione
  - **Chiusura**

7

## Interazione SMTP

- Interazione mediante comandi/risposte
    - Comando: testo ASCII
    - Risposta: codice di stato e frase
- ```

S: 220 mail.unisannio.it
C: HELO mail.unisannio.it
S: 250 Hello mail.unisannio.it , pleased to meet you
C: MAIL FROM: <alice@unisannio.it>
S: 250 alice@unisannio.it ... Sender ok
C: RCPT TO: <bob@unisannio.it>
S: 250 bob@unisannio.it ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: ciao!
C: .
S: 250 Message queued for delivery
C: QUIT
S: 221 closing connection. Good bye
    
```

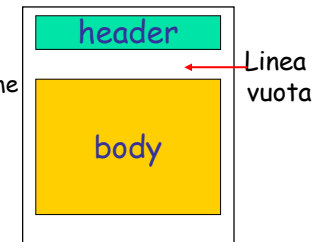
8

- telnet servername 25
- Attendi 220 risposta dal server
- Inserisci HELO, MAIL FROM, RCPT TO, DATA, QUIT
- Questi comandi permettono di inviare e-mail senza usare uno User Agent!
- Inserire HELP per ottenere una lista dei comandi accettati

- Entrambi i protocolli sono impiegati per trasferire file ma ...
  - HTTP è un protocollo di tipo PULL
    - E' il client a richiedere al server gli oggetti
  - SMTP è un protocollo di tipo PUSH
    - Il client spedisce i file, i dati viaggiano dal client al server
- SMTP trasferisce file in formato ASCII a 7 bit
  - HTTP non impone questo vincolo
- La terminazione di un messaggio è individuata dalla presenza dei seguenti caratteri **CRLF.CRLF**
- La trasmissione diretta di byte di contenuti multimediali può causare dei problemi
  - E' possibile incorrere nella presenza CRLF.CRLF nel contenuto binario di un messaggio
  - La soluzione è quella di trasformare il contenuto binario in codice ASCII a 7 bit convertendo alcuni caratteri tra cui CRLF.CRLF

- In HTTP la fine del trasferimento di un oggetto è individuata in modo diverso:
  - Con connessioni non persistenti -> chiusura della connessione TCP
  - Con connessioni persistenti -> campo Content-length presente in ogni messaggio di risposta
- In HTTP il testo e le immagini viaggiano in messaggi separati mentre in SMTP essi sono contenuti all'interno di un unico file

- RFC 822: standard per il formato dei messaggi inviati:
- **header**
  - Singola linea di testo ASCII contenente il nome del campo ed un valore separato da :
    - **To:** (indirizzo/i destinatario/i primario/i)
    - **Cc:** (indirizzo/i destinatario /i secondario /i)
    - **Bcc:** (indirizzo per copie carbone trasparenti)
    - **From:** (chi ha creato il messaggio)
    - **Subject:** (breve descrizione del contenuto del messaggio)
    - **Reply-to:** (indirizzo al quale inviare la risposta)
    - **Received:** (aggiunta da ogni server lungo il percorso)
    - **Message-Id:** (identificativo univoco del messaggio)
    - Es. Reply-to [ranaldo@unisannio.it](mailto:ranaldo@unisannio.it)  
Received: from ... by ... <orario>
- **body:** Il "messaggio" vero e proprio, solo caratteri ASCII



*Diversi dai comandi smtp!*

## Messaggio ricevuto (1)

```
Received: from mail.domain.it by mail.unisannio.it; 6
  Oct 2003
From: bob@domain.it
To: alice@unisannio.it
Subject: Ciao!
Reply-to: bob@domain.it
```

- Received indicano i mail server che hanno recapitato il messaggio
- Più linee "Received" se il messaggio è stato inoltrato da più server SMTP lungo il percorso da mittente a destinatario

13

## Il formato MIME (1)

- Per inviare contenuti diversi dal testo ASCII a 7 bit, lo user agent deve includere nel messaggio intestazioni aggiuntive
  - Messaggi in linguaggi con accenti (Francia, Germania, Italia)
  - Messaggi in linguaggi con alfabeti non latini (Ebraico, Russo)
  - Messaggi in linguaggi senza alfabeti (Cinese, Giapponese)
  - Messaggi che non contengono testo (audio, immagini)
- Tali estensioni sono definite negli RFC dal 2045 al 2049 e si chiamano estensioni **MIME (Multipurpose Internet Mail Extensions)**
- MIME è uno standard per l'invio dei dati multimediali e composti attraverso la posta elettronica di Internet
  - E' divenuto una tecnica ampiamente utilizzata per descrivere i contenuti di un file
- L'idea di MIME è quella di continuare ad utilizzare il formato RFC 822 introducendo delle regole di codifica aggiuntive per i messaggi non ASCII
- Questa scelta consente di conservare i mail server e i protocolli
  - Le modifiche sono apportate agli user agent

14

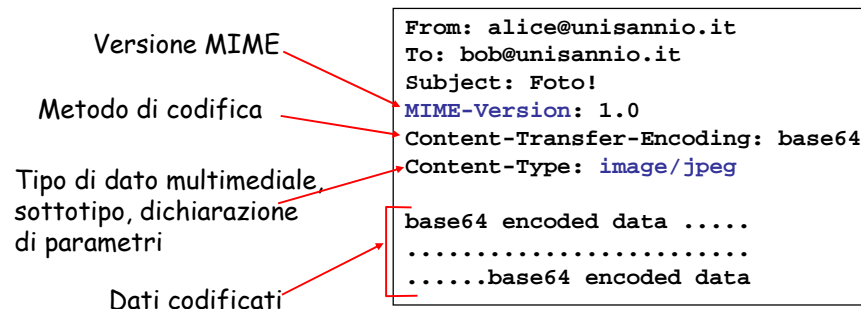
## Il formato MIME (2)

- Le due principali intestazioni aggiuntive per specificare il tipo di contenuto sono:
  - **Content-type**: permette allo user agent del client che riceve il messaggio di elaborare opportunamente il contenuto del messaggio
    - quale applicazione deve interpretarlo
  - **Content-Transfer-Encoding**: indica che il contenuto non ASCII del messaggio è stato codificato e secondo quale codifica
    - alla ricezione il messaggio deve essere decodificato nel formato non ASCII originale
- Altre intestazioni
  - **Content-Description**: Stringa che descrive il contenuto del messaggio
  - **Content-id**: Identifica il contenuto (stesso formato di Message-id)

15

## Esempi

- **Base64** è una delle varie tecniche di codifica standardizzate in MIME per la conversione in un formato ASCII a 7 bit accettabile
  - Gruppi di 24 bit sono suddivisi in unità da 6 bit, ognuno codificato in ASCII ('A' per 0, 'B' per 1, etc. poi le dieci cifre ed infine + e / per 62 e 63)
- **Quoted-printable content-transfer-encoding**
  - Per convertire un messaggio a 8 bit (caratteri non inglesi) in ASCII a 7 bit
  - Ogni carattere con codice ASCII > 127 è codificato con = ed il valore del codice con due cifre esadecimali



16

### Text

- Esempi di sottotipi: plain, html

### Image

- Esempi di sottotipi : jpeg, gif

### Audio

- Esempi di sottotipi : basic, 32kadpcm (32 kbps coding)

• MIME è estendibile, e i nuovi tipi e sottotipi sono registrati presso un apposito registro centrale

### Video

- Esempi di sottotipi: mpeg, quicktime

### Application

- Dati che devono essere processati da un'applicazione prima di essere "visibili"
- Esempi di sottotipi: msword, octet-stream (file binario qualsiasi)

### Message

- Per incapsulare un messaggio in un altro
  - Per il forwarding

- Un tipo importante del formato MIME è il **multipart**
  - E' impiegato per consentire la trasmissione di oggetti diversi all'interno di un stesso messaggio di posta elettronica
- Il sottotipo più comune è **mixed**: il messaggio contiene parti indipendenti
  - Content-Type: multipart/mixed
- E' necessario un criterio per individuare
  - I **confini di ogni messaggio** (caratteri **boundary**: una stringa qualsiasi)
  - Come codificare i messaggi non ASCII
  - Il tipo di contenuto di ogni messaggio

## Il tipo multipart - Esempio (1)

```
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--
```

## Il tipo multipart - Esempio (2)

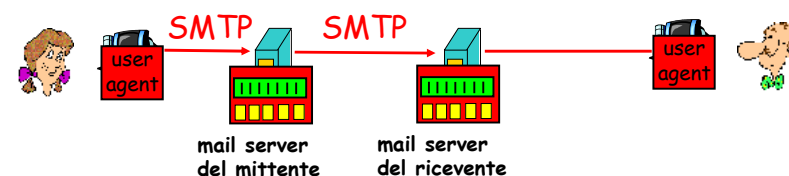
```
...
MIME-Version: 1.0
Subject: Auguri
Content-Type: multipart/mixed;
boundary="-----040807030609040704010005"
Content-Length: 195788
This is a multi-part message in MIME format.
-----040807030609040704010005
Content-Type: text/html; charset=ISO-8859-15
Content-Transfer-Encoding: 7bit
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html> <head>
</head>
<body alink="#000099" bgcolor="#ffffff" link="#000099" text="#000000"
vlink="#990099">
<div align="center"><big><b><font color="#ff0000">TANTI AUGURI di BUON NATALE e
FELICE 2007! da Salvatore</font></b></big><br>
</div> </body> </html>
-----070902060803070706030104
Content-Type: image/jpeg;
name="b101791-big.jpg"
Content-Transfer-Encoding: base64
Content-ID: <part1.06010407.00080702@gmail.com>
/9j/4AAQSkZJRgABAgAAQABAAD/4QDxRXhpZgAASUkqAAgAAAAAFABI BAwABAAAAQAAAD
EBAgAnAAAAcGAAADIBAgUAAAAcQAAABMCAwABAAAAQAAAGmHBAABAAAAhQAAAAAAAA
ABBQ0Qg...
```

## Accesso ai Mail Server: spedizione (1)

- Inizialmente nella rete Internet (Arpanet) tutti i computer erano sempre connessi alla rete
  - Gli utenti lavoravano con macchine in grado di inviare e ricevere e-mail
- Oggi la maggior parte degli utenti utilizzano un ISP per la connessione ad Internet
  - Per la spedizione, un eventuale problema sulla connessione o sul computer del destinatario o sulla connessione non consentirebbe la consegna del messaggio
- Soluzione tradizionale: utente legge direttamente la posta (e invia messaggi) sul mail server
- Gli utenti possono utilizzare gli user agent per spedire e leggere la posta
- L'host su cui è disposto il mail server deve essere sempre attivo
- Invece gli user agent sono connessi alla rete dei mail server solo quando è necessario
  - Uno user agent si connette alla rete all'atto dell'invio di un messaggio
  - Uno user agent si connette alla rete solo quando desidera leggere i messaggi ricevuti

21

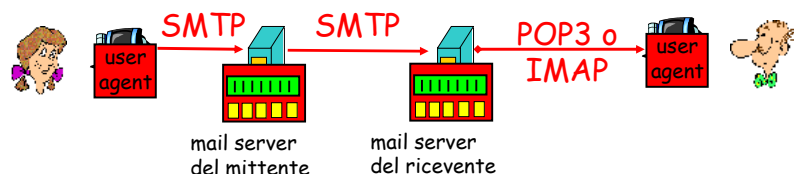
## Accesso ai Mail Server: spedizione (2)



- Gli user agent offrono un insieme di funzionalità aggiuntive, ad esempio permettono di visualizzare file multimediali e di specifiche applicazioni
- Lo user agent viene connesso al mail server dell'ISP a cui è connesso l'utente impiegando il protocollo SMTP (si comporta come un client SMTP)
- Il messaggio è affidato al mail server che può tentare ripetutamente la consegna al destinatario

22

## Accesso ai Mail Server: lettura



- Lo user agent interagisce con il mail server di destinazione impiegando specifici protocolli
  - In questo caso l'interazione è di tipo pull e quindi non può essere impiegato il protocollo SMTP (di tipo push)
  - Lo user agent richiede i messaggi al mail server ed il mail server li invia
  - **POP3 (Post Office Protocol - version 3)**
    - Impiega una connessione TCP sulla porta 110 (RFC 1939)
  - **IMAP (Internet Mail Access Protocol) (RFC 2060)**
    - Maggiori funzionalità

23

## Protocollo POP3 (1)

### Fase di autorizzazione

- Comandi del client:
  - user: nome utente
  - pass: password
- Risposte del server
  - +OK
  - -ERR

### Fase di transazione:

- list: lista numeri e dimens msg
- retr: scarica messaggio in base al numero
- dele: marca per la cancellazione

### Fase di aggiornamento:

- Quit
- Quando il client chiude la connessione i messaggi marcati sono cancellati

```

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 2 contents>
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off
    
```

24

## Protocollo POP3 (2)

- Il protocollo POP3 può essere configurato in modo da lasciare sul server o rimuovere dal server i messaggi scaricati dall'agente
- Scarica ed elimina:**
  - Lo User Agent elimina la posta dalla Mailbox dopo averla scaricata
  - Un utente disperde la posta sui diversi host che usa per accedere alla propria Mailbox
  - Lo User Agent permette di creare cartelle, spostare messaggi, effettuare ricerche nei messaggi
- Scarica e conserva:**
  - Lo User Agent conserva la posta sulla Mailbox
  - L'utente può leggere i messaggi da macchine diverse
  - POP3 non mantiene lo stato tra una sessione ed un'altra, non permette di strutturare i messaggi in directory

25

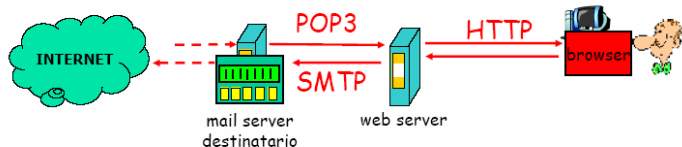
## Protocollo IMAP

- Permette di gestire cartelle di posta remote come se fossero locali
- IMAP deve mantenere una gerarchia di cartelle per ogni utente e informazioni di stato tra una sessione e un'altra
- Permette allo User Agent di scaricare solo parti del messaggio:
  - Intestazione
  - Solo intestazione file MIME multipart
  - Messaggi di dimensione piccola per utenti a banda limitata

26

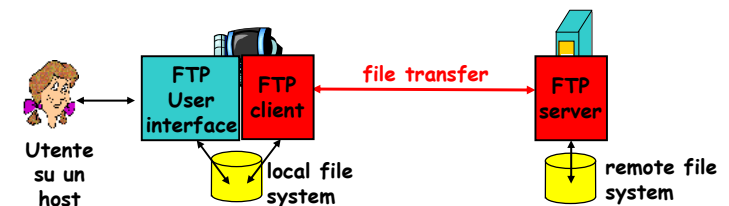
## L'accesso alla posta via Web

- Molti siti web forniscono accesso alle proprie caselle di posta (libero, inwind, tin, 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



27

## FTP: File Transfer Protocol



- Trasferimento file da/verso un host remoto
- Usa il modello client/server
  - client:** parte che richiede il trasferimento (da/verso l' host remoto)
  - server:** host remoto
- ftp: RFC 959
- ftp server: porta 21

28

## FTP: connessioni controllo e dati separate

- Il client contatta il server sulla porta 21, specificando TCP come protocollo di trasporto
- Due connessioni TCP parallele:
  - controllo**: scambio di messaggi di controllo tra client e server.  
"controllo fuori banda"
  - dati**: trasferimento dati da/verso il server
- Entrambi le connessioni aperte dal client
- Il server ftp mantiene informazioni di "stato": directory corrente, autenticazione
- Una nuova connessione per ogni file trasferito



## FTP: comandi e risposte

### Esempi di comandi:

- Inviati come testo ASCII mediante il canale di controllo
- USER** *username*
- PASS** *password*
- LIST** richiede la lista dei file nella directory corrente (ls)
- RETR** <file> richiede (get) un file
- STOR** <file> scarica (put) un file sull' host remoto

### Esempi di codici

- Codice di stato e frase (come in HTTP)
- 331 Username OK, password required
- 125 data connection already open; transfer starting
- 425 Can't open data connection
- 452 Error writing file