

PARTE 12

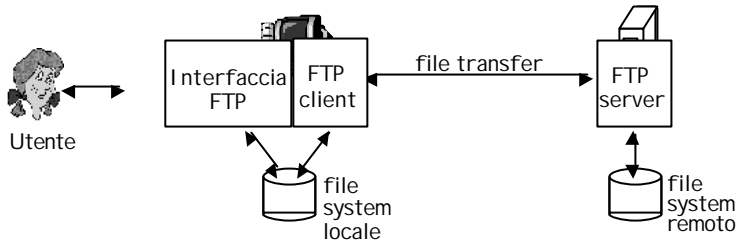
LIVELLO APPLICAZIONI

- Altri protocolli “storici”

Parte 12

Modulo 1: Trasferimento remoto file

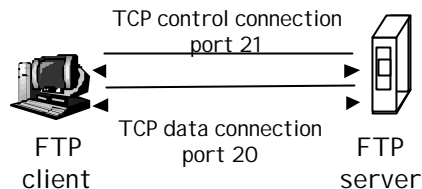
File Transfer Protocol



- Trasferimento di file da/verso un host remoto: FTP [RFC959]
- Paradigma client/server
 - **client**: inizia il trasferimento (sia verso/da remoto)
 - **server**: host remoto
- *Well known ports* dell'FTP del server:
 - **porta 21**: controllo
 - **porta 20**: dati
- Protocollo di trasporto TCP

File Transfer Protocol (cont.)

- Il client FTP client contatta il server FTP sulla porta 21, specificando TCP come protocollo di trasporto
- Vengono aperte **due** connessioni TCP parallele:
 - **connessione per controllo**: scambio dei comandi (put, get), delle risposte tra client e server
 - **connessione per dati**: file di dati da/verso il server (una connessione dati per ogni file)



- Il server FTP mantiene lo “stato”:
 - directory corrente
 - autenticazione precedente

Comandi e risposte FTP

Alcuni comandi:

- inviati come testo ASCII sulla connessione di controllo
- più di 50 comandi
- **USER *username***
- **PASS *password***
- **LIST** restituisce la lista dei file nella directory corrente (inviata su una nuova connessione dati)
- **RETR *filename***: prendi (get) il file dalla directory corrente
- **STOR *filename***: memorizza (put) il file nella directory corrente dell'host remoto

Alcuni codici di risposta:

- codice dello 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**

Parte 12

Modulo 2: (Trasferimento) posta elettronica

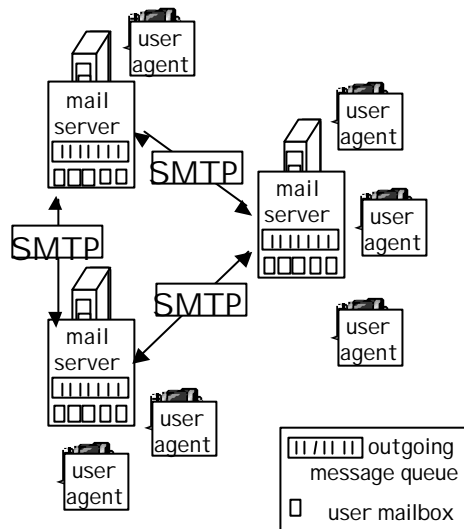
Posta elettronica

- Tre componenti principali:

- **user agent**
- **mail server**
- **Simple Mail Transfer Protocol (SMTP)**

- **User Agent**

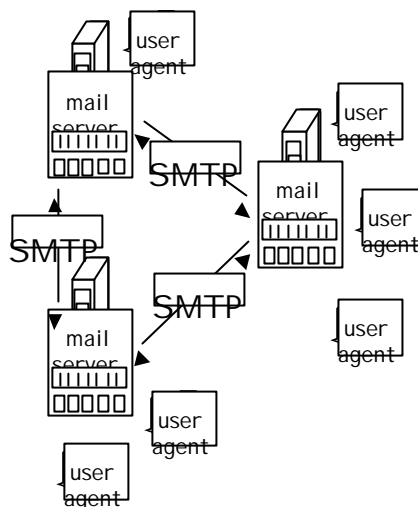
- “lettore della posta”
- composizione, editing, lettura messaggi di posta
- esempi: Eudora, pine, elm, Netscape Messenger, Outlook, Express
- messaggi in ingresso/uscita memorizzati sul server



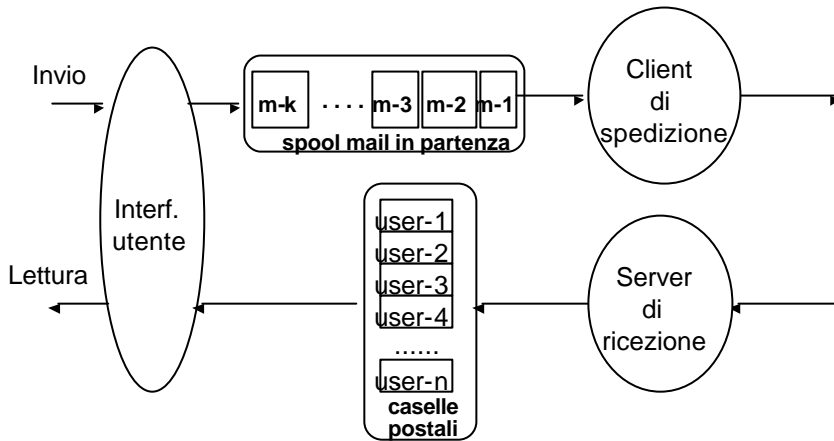
Mail server

- **Mail Server**

- **mailbox** contenente i messaggi in ingresso (ancora da leggere) dell'utente
- **coda di messaggi** di posta in uscita (da inviare)
- **protocollo SMTP** tra mail server per inviare messaggi di posta



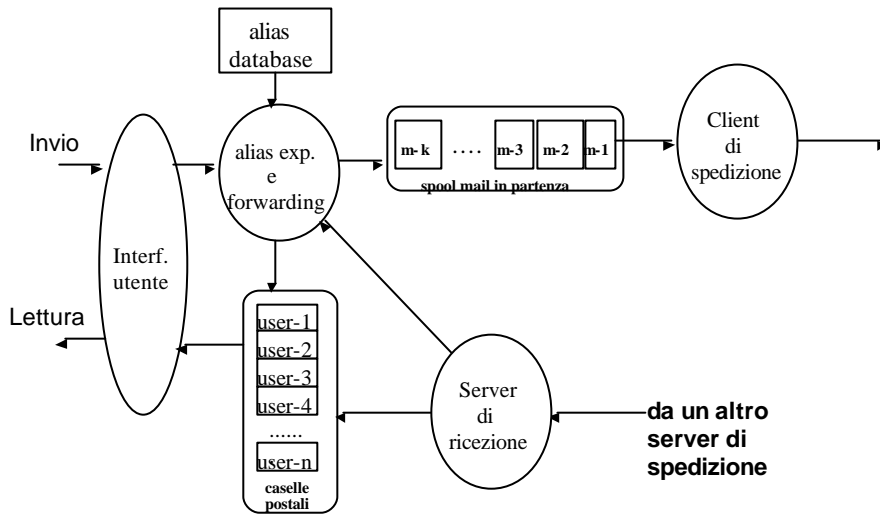
Architettura di mail



Destinazioni e mailbox

- La destinazione è una macchina il cui nome è caratterizzato come “mail-exchange object”
- Il nome della mailbox, normalmente corrisponde ad uno username, ma può corrispondere ad un “alias”
- Una mail può anche arrivare da un utente interno (quindi non arriva tramite una connessione dall'esterno)
- Una mail, sia dall'esterno sia dall'interno, può essere inoltrata ad un'altra destinazione (*mail forwarding*)

Architettura completa



Simple Mail Transfer Protocol

SMTP [RFC 821] [RFC 822]

- protocollo tra mail server
- paradigma client/server, in cui:
 - **client**: mail server del mittente
 - **server**: mail server del destinatario
 - **client e server in esecuzione su ogni mail server**

Simple Mail Transfer Protocol (*cont.*)

- Usa il protocollo di trasporto TCP per il trasferimento affidabile dei messaggi tra client e server (**porta 25**)
- **Trasferimento diretto: dal mail server del mittente al mail server del destinatario**
- Tre fasi del trasferimento TCP: handshaking, trasferimento, chiusura
- Interazione comando/risposta:
 - comando: testo ASCII
 - risposta: codice di stato e frase
- Il messaggio deve essere in ASCII a 7 bit

Simple Mail Transfer Protocol (*cont.*)

- Il messaggio di mail consiste di due parti:
 - un header che contiene dei campi codificati
 - il body del messaggio che è un testo libero in ASCII a 7 bit
- Il dialogo sender-receiver avviene sulla **porta 25** con connessione TCP ed è costituito da “frasi” in formato testuale comprensibili immediatamente ad un essere umano

Simple Mail Transfer Protocol (cont.)

- SMTP usa connessioni TCP persistenti per trasferire più messaggi in una sola volta
- Alcune stringhe di caratteri non sono permesse nel messaggio (es., CRLF.CRLF). Quindi il messaggio deve essere codificato (di solito in **base 64** oppure **quoted printable**)
- Il server SMTP usa CRLF.CRLF per indicare la fine del messaggio

Simple Mail Transfer Protocol (cont.)

Attivazione della connessione TCP

```
S: 220 mail.ucla.edu
C: HELO mail.unimo.it
S: 250 Hello mail.unimo.it, pleased to meet you
C: MAIL FROM: <alice@mail.unimo.it>
S: 250 alice@ mail.unimo.it ... Sender ok
C: RCPT TO: <bob@mail.ucla.edu>
S: 250 bob@ mail.ucla.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Do you like cs books?
C: How about journals?
C: .
S: 250 Message accepted for delivery
C: QUIT
```

CRLF.CRLF
per indicare la
fine del
messaggio

connection

Chiusura della connessione TCP

Dialogo sender-receiver (lato *sender*)

Nome	Formato	Descrizione
HELO	HELO<sp><domain><crLf>	identificazione del sender
MAIL	MAIL<sp>FROM:<rev.-path><crLf>	identifica il mittente
RCPT	RCPT<sp>TO:<forward-path><crLf>	identifica il destinatario
DATA	DATA<crLf>	inizia la trasmissione
RSET	RSET<crLf>	abortisce la transazione
NOOP	NOOP<crLf>	no operation
QUIT	QUIT<crLf>	chiude la connessione TCP
SEND	SEND<sp>FROM :<rev.-path><crLf>	manda la mail al terminale
SOML	SOML<sp>FROM :<rev.-path><crLf>	manda la mail al terminale se possibile se no alla mailbox
SAML	SAML <sp>FROM :<rev.-path><crLf>	manda la mail al terminale e alla mailbox
VRFY	VRFY<sp><string><crLf>	verifica un user name
EXPN	EXPN <sp><string><crLf>	riporta l'appartenenza ad una mailing list
HELP	HELP<sp>[<string>]<crLf>	invia documentazione di sistema
TURN	TURN<crLf>	scambia i ruoli sender e receiver

Dialogo sender-receiver (lato *receiver*)

Codice	Descrizione (risposta)
211	system status/system help reply
214	<i>messaggio di help</i> (pagina di manuale, per una persona)
220	<domain> service ready
221	<domain> service closing transmission channel
250	Requested mail action ok, completed
251	User not local; will forward to <forward-path>
354	Start mail input; en with<crLf><crLf>
421	<domain>Service not available, losing transmission channel
450	Requested mail action not taken; mailbox unavailable (eg.: busy)
451	Requested action not taken: insufficient system storage
500	Syntax error, command unrecognized
501	Syntax error in parameter or arguments
502	Comand not implemented
503	Bad sequence of commands
504	Command parameter not implemented
550	Requested mail action not taken: mailbox unavailable (not found)
551	User not local; please try <forward path>
552	Requested action aborted: eceeded storage allocation
553	Requested action aborted: mailbox name not allowed (eg.: syntax error)
554	Transaction failed

Consegna mail

- La comunicazione tra il client di spedizione e il server di ricezione avviene su protocollo di trasporto TCP
- Questo garantisce che quando il messaggio è rimosso dallo spool del mittente, è stato recapitato con successo al ricevitore (non ci sono passi intermedi a livelli di mail exchange)
- Esistono invece altri sistemi di mail che prevedono siti intermediari per la mail (*mail gateway*, *mail bridges*, *mail relay*). In questo caso si è garantiti solo fino al primo recapito, poi si perdono le tracce ...

Provare SMTP in “altro modo”

telnet servername 25

(servername è il nome del mail server remoto)

- Attendi la risposta 220 dal server
- Inserisci i comandi HELO, MAIL FROM, RCPT TO, DATA, QUIT

→E' possibile inviare una mail senza usare uno user agent!

SMTP vs. HTTP

- HTTP: protocollo pull
- SMTP: protocollo push
- entrambi presentano un'interazione di tipo comando/risposta, ed usano codici di stato
- HTTP: oggetti multipli in un file inviati in connessioni separate
- SMTP: parti multiple di un messaggio inviate in una connessione

Formato del messaggio SMTP

RFC 822 definisce lo standard per il formato del messaggio:

- **Linee di header**, es.:

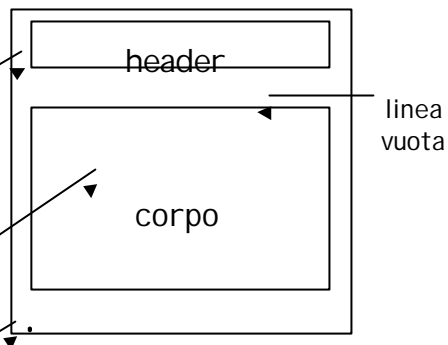
- To:
- From:
- Subject:

Diverse dai comandi SMTP!

- **Corpo**

- il “messaggio”, soltanto in caratteri ASCII

- **Linea contenente solo ‘.’**



Esempio

Subject: Avviso
Date: 12/12/2000, 18:25
From: <direttore@dii.unimo.it>
To: <mc@dii.unimo.it>
Replay-to: segretaria@dii.unimo.it

Caro Professor Colajanni,
volevo comunicarle che.....

- Alcuni campi nell'header sono obbligatori, altri sono opzionali
- Il client di trasmissione interpreta l'header in modo da gestire il dialogo col server corrispondente
- Quindi trasmette il messaggio una riga alla volta

Parte 12

Modulo 3: MIME

Messaggi non testuali

SMTP tratta correttamente soltanto caratteri ASCII a 7 bit

PROBLEMA: Come trasferire un messaggio che contiene altri caratteri o addirittura dati in binario?

Metodi gestiti esplicitamente dall'utente:

- uuencode
- binex
-

Standard de facto attuale:

- **Multipurpose Internet Mail Extension (MIME)**
[RFC 2045, 2056]

MIME

- MIME si interpone tra l'interfaccia utente ed il mailer automatizzando la procedura di conversione
- Introduce un ulteriore livello di incapsulamento
- L'header dello standard permette solo l'invio di messaggi di caratteri ASCII → serve estensione per supporto multimediale
- **Si estende l'header del messaggio con 5 campi specifici del MIME**

Estensione header del MIME

- **MIME-version:** deve avere il valore 1.0 per indicare la conformità alle specifiche delle RFC
- **Content-Type:** descrive i dati nel “body” in modo tale che l’agente ricevente possa scegliere l’applicazione adatta per decodificare i dati
- **Content-Transfer-Encoding:** tipo di trasformazione usata per il body per renderlo trasmissibile in caratteri ASCII a 7 bit
- **Content-ID:** usato per identificare le entità del MIME in contesti multipli
- **Content-Description:** una descrizione testuale dell’oggetto codificato (commento)

MIME Content type

Tipo	Sotto-tipo	descrizione
Text	Plain	testo semplice
Multipart	Mixed	parti indipendenti ma l'ordine si mantiene
	Parallel	parti indipendenti ma l'ordine non si mantiene
	Alternative	versioni alternative della stessa parte
	Digest	simile a mixed ma il tipo/sottotipo di default è message/rfc822
Message	rfc822	il “body” stesso è un messaggio conforme a RFC822
	Partial	frammento di un body più grande, spezzato
	External-body	contiene un puntatore ad un oggetto che esiste altrove
Image	jpeg	formato JPEG codificato JFIF
	gif	formato GIF
Video	mpeg	formato video MPEG
Audio	Basic	canale singolo 8 bit ISDN mu-law
		codificante un campione a 8KHz
Application	PostScript	file in postscript
	octect-stream	dato binario in byte (8 bit completi)

Esempio

Subject: Avviso
Date: 12/12/2000, 18:25
From: <direttore@dii.unimo.it>
To: <mc@dii.unimo.it>
MIME-version: 1.0
Content-type: multipart/mixed; boundary: boundary11

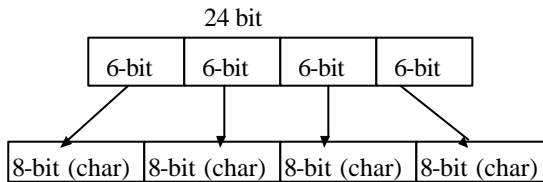
preambolo esplicativo
(ignorato dal MIME, ma usato dalle interfacce utente)
--boundary11
testo implicitamente codificato in ASCII.....
--boundary11
Content-type: text/plain; charset=us-ascii
testo esplicitamente codificato in ascii come testo semplice
--boundary11
epilogo, (anche questo ignorato dal MIME)

MIME message type

message/rfc822: definisce ricorsivamente il contenuto come un altro messaggio completo (incapsulato nel body di una parte)
message/partial: viene usato per gestire la frammentazione di un messaggio in più messaggi:
id: valore comune a tutti frammenti
number: numero d'ordine della sequenza
total: numero totali di parti
message/external-body: usato per accedere a documenti esterni
FTP: seguono i parametri per un comando ftp
TFTP: come sopra
Anon-FTP: ftp con login anonimo
local-file: segue un path-name locale sul ricevitore
AFS: file accessibile via Andrew File System
mail-server: body accessibile inviando una mail a un dato sito

MIME Content transfer encoding

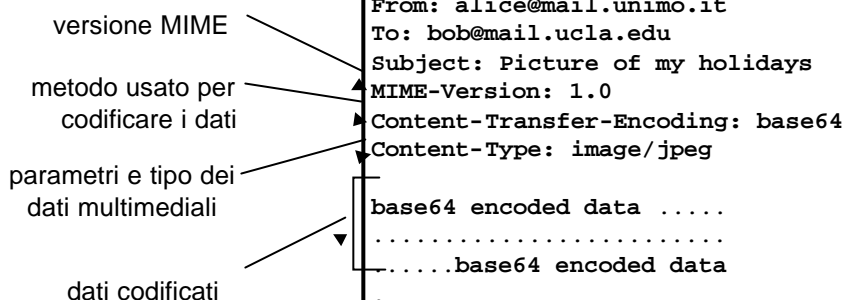
Codifica	Descrizione
7bit	codifica a 7 bit, linee corte con ASCII
8bit	codifica a 8 bit, linee corte ma anche caratteri a 8 bit
binary	file binario, le linee possono essere anche lunghe
quoted-printable	codifica in modo speciale solo i caratteri non ASCII
base64	codifica in radice 64
x-token	forma di codifica non standard



65 caratteri:
64 di codifica
+
1 di padding

Formato del messaggio MIME

Ulteriori linee nell'header del messaggio SMTP dichiarano il tipo di contenuto MIME



Tipi/sottotipi MIME

Content-Type: type/subtype; parameters

Text

- esempio di sottotipi:
plain, html

Video

- esempio di sottotipi:
mpeg, quicktime

Image

- esempio di sottotipi:
jpeg, gif

Audio

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

Application

- altri dati che devono essere processati prima di essere fruibili all'utente
- esempio di sottotipi:
msword, octet-stream

Esempio MIME

Multipart type: per indicare la presenza di oggetti multipli

From: alice@mail.unimo.it
To: bob@mail.ucla.edu
Subject: Picture of my holidays
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=98766789

--98766789
Content-Transfer-Encoding: quoted-printable
Content-Type: text/plain

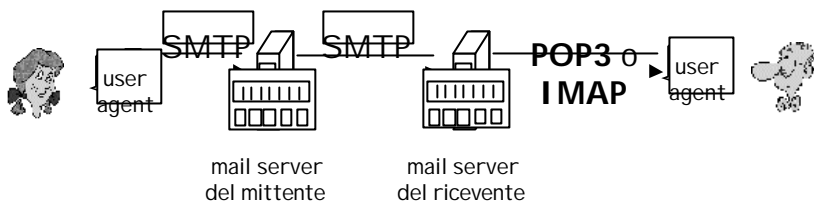
Dear Bob,
Please find a picture of my holidays.

--98766789
Content-Transfer-Encoding: base64
Content-Type: image/jpeg

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

Modulo 4: Accesso alla posta

Protocolli di accesso alla posta



- **SMTP: consegna/memorizzazione al mail server del destinatario**
- **Protocollo di accesso alla posta: consegna dal mail server**
 - **POP: Post Office Protocol** [RFC 1939]
 - autorizzazione (user agent ↔ mail server) e download
 - **IMAP: Internet Mail Access Protocol** [RFC 1730]
 - più caratteristiche (maggiore complessità)
 - manipolazione dei messaggi memorizzati sul server
 - **HTTP**: accesso alla mail tramite Web (Hotmail, Yahoo! Mail, ecc.)

Protocollo POP3

Fasi di una sessione POP3

- **Fase di instaurazione della connessione**
 - user agent apre una connessione TCP con mail server (porta 110)
- **Fase di autorizzazione**
 - user agent invia al mail server la propria login e password
- **Fase di transazione**
 - user agent recupera i messaggi
 - user agent può indicare alcuni messaggi affinché siano cancellati (modalità download-and-delete e download-and-keep)
- **Fase di aggiornamento**
 - dopo il comando **quit** eseguito dal client vengono cancellati dalla mailbox i messaggi indicati dal client

Protocollo POP3 (2)

Fase di autorizzazione

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

Fase di transazione, client:

- **list**: lista i numeri dei messaggi
- **retr**: scarica i messaggi tramite il numero
- **dele**: cancella
- **quit**

```
S: +OK POP3 server ready
C: user alice
S: +OK
C: pass pippo
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
```

Protocollo IMAP

- Più funzionalità e maggiore complessità rispetto al protocollo POP3
- Permette all'utente di modificare la propria mailbox come se fosse locale
 - Es.: creazione di cartelle (*folder*) remote nella mailbox
 - Es., ricerca (*search*) di particolari messaggi nelle cartelle remote
- Il server IMAP deve essere in grado di gestire una gerarchia di mailbox per ogni utente
- Permette all'utente di ottenere alcune parti del messaggio
 - Es., specificare quali attachment “scaricare”
 - Caratteristica utile per connessioni a banda stretta (ad es., dispositivi mobili)

Fasi di una sessione IMAP

- Fase di **instaurazione della connessione**
 - user agent apre una connessione TCP con il mail server
- Fase di **autorizzazione**
 - user agent invia al mail server la propria login e password
- Fase di **transazione**
 - comprende comando client, dati dal server, risultato del server
 - quattro stati del server IMAP:
 - non autenticato
 - autenticato
 - selezionato
 - logout

Accesso alla posta tramite Web

- Servizi di posta elettronica tramite Web (es., Hotmail e Yahoo! Mail)
- Il Web browser è lo user agent
 - comunicazione con mailbox su mail server tramite HTTP
- Permette all'utente di modificare la propria mailbox come se fosse locale (analogamente a IMAP)
 - es., definizione di cartelle (folder) remote nella mailbox
 - es., ricerca di determinati messaggi nelle cartelle remote
- Svantaggio
 - maggiore lentezza