

Università di Genova
Facoltà di Ingegneria

Livello di Applicazione in Internet

4. Posta Elettronica **(*email*)**

Prof. Raffaele Bolla
Ing. Matteo Repetto

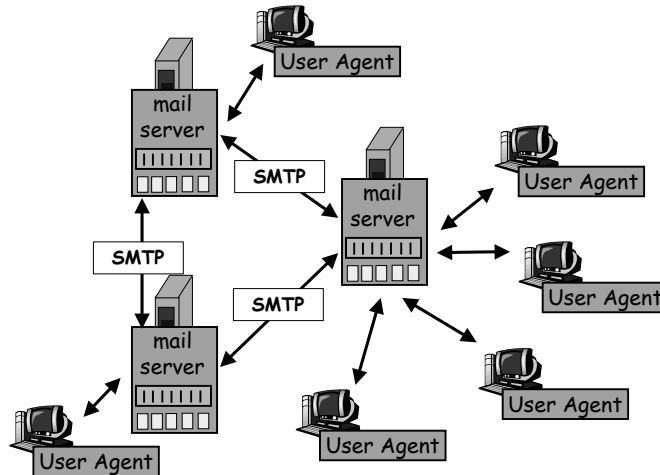


Posta elettronica

- L'*e-mail*, o posta elettronica, è una delle applicazioni di rete attualmente più diffuse.
- L'applicazione di posta elettronica consente la visione/trasferimento asincrono di messaggi contenenti testo, *hyperlink*, immagini, suoni e video.
- L'asincronicità del servizio consiste nel fatto che
 - il messaggio è consegnato e posto in una casella di posta (*mailbox*) associata al destinatario, senza che questi debba essere presente.
 - Il destinatario può accedere e leggere la propria “posta” in qualunque momento possa e lo desidera.
- Pur essendo simile al servizio di “posta ordinaria”, la posta elettronica è molto più veloce (su Internet quasi in tempo reale), più flessibile e molto meno costosa.

4.2

Il servizio di posta elettronica: Schema di principio



4.3

I componenti del sistema

- Server di posta (*Mail Server*):
 - ricevono i messaggi da consegnare;
 - smistano la posta nelle *mailbox* degli utenti locali;
 - inviano i messaggi dei propri utenti al server di destinazione corretto;
 - consentono agli utenti di accedere alle proprie *mailbox*;
 - Es. di applicativi: *sendmail*, *exim*, *qmail*, *postfix*, *exchange*.
- *User Agent*:
 - forniscono un'interfaccia agli utenti per le operazioni legate ai messaggi (lettura, scrittura, ...);
 - interagiscono con i server di posta per inviare/ricevere messaggi e gestire le *mailbox*;
 - es. di applicativi: *mail*, *pine*, *elm*, *Eudora*, *Outlook*, *Netscape Messenger*.

4.4

Architettura del sistema di posta elettronica

- L'utente utilizza il proprio *user agent* per collegarsi ad un *Mail Server* allo scopo di:
 - accedere alla propria *mailbox*;
 - inviare messaggi.
- In linea di principio, un *Mail Server* potrebbe anche risiedere sul PC locale dell'utente
 - In questo caso, però, il PC dovrebbe rimanere sempre acceso e collegato alla rete, in modo da poter ricevere in qualunque momento nuovi messaggi dagli altri server.
- L'utente è individuato da un URL:

<utente>@<server> o **<utente>@<dominio>**

 - il server che fornisce il servizio è identificato interrogando il DNS responsabile del dominio in questione;
 - lo stesso server può fornire il servizio di posta per domini diversi.

4.5

Servizi e protocolli di posta elettronica

- La posta elettronica fornisce due tipologie di servizio agli utenti:
 - Il trasferimento vero e proprio dei messaggi;
 - L'accesso remoto alle *mailbox* degli utenti.
- I protocolli utilizzati per fornire queste tipologie di servizio sono rappresentati da:
 - **SMTP** (Simple Mail Transfer Protocol): definisce le modalità di trasferimento del messaggio fra Server di posta.
 - **POP** (Post Office Protocol), **IMAP** (Internet Message Access Protocol): due distinti protocolli che definiscono le modalità di accesso da parte degli *user agent* alle proprie *mailbox*.

4.6

Invio di un messaggio di Posta Elettronica (1)

- Nel momento in cui un utente desidera inviare un messaggio di posta elettronica, le operazioni che consentono di consegnare il messaggio sono le seguenti:
 - Lo *user agent* del mittente richiede il servizio di inoltro al proprio *Mail Server* utilizzando il protocollo SMTP.
 - Tale *Mail Server* consegna il messaggio ricevuto direttamente al server di destinazione, senza utilizzare nessun intermediario.
 - Il *Mail Server* di destinazione riceve il messaggio e, in alternativa:
 - » inserisce il messaggio nella *mailbox* dell'utente ;
 - » inoltra il messaggio verso una diversa destinazione (*forwarding*).

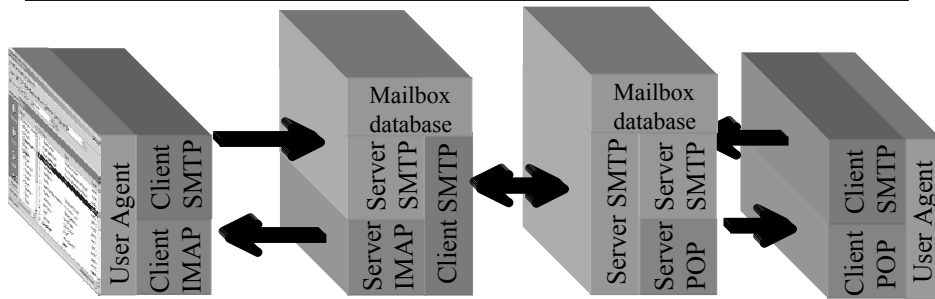
4.7

Invio di un messaggio di Posta Elettronica (2)

- In teoria lo *user agent* potrebbe anche inviare il messaggio direttamente al server dell'utente di destinazione.
- L'utilizzo del proprio *Mail Server* risulta più opportuno per gestire sia i permessi d'accesso, sia eventuali problemi nella consegna del messaggio (destinazione non raggiungibile o malfunzionante):
 - i server SMTP mantengono una coda (*message queue*) dei messaggi da inviare;
 - nel caso in cui la consegna non sia possibile, il messaggio rimane in coda e la consegna viene rimandata;
 - ad intervalli regolari (di solito 30 minuti), il mail server tenta l'invio dei messaggi ancora presenti nella coda;
 - dopo che il messaggio è rimasto in coda per un certo periodo (generalmente alcuni giorni) il server interrompe i tentativi e notifica il fallimento all'utente tramite *e-mail*.

4.8

Ciclo completo di trasferimento di un messaggio di posta



1. L'utente interroga il proprio server per l'invio del messaggio e/o per l'accesso alla propria *mailbox*:
 - La consegna del messaggio al proprio server avviene tramite il protocollo SMTP,
 - La ricezione può avvenire utilizzando il protocollo POP o IMAP.
2. Il messaggio in uscita dal Server viene trasferito, tramite una transazione SMTP, al Server di destinazione e da questi inserito nella *mailbox* dell'utente.
3. L'utente destinatario (quando lo ritiene opportuno) accede alla propria *mailbox* tramite i protocolli POP o IMAP e recupera/gestisce i messaggi in essa presenti.

4.9

SMTP - *Simple Mail Transfer Protocol*

- L'SMTP (*Simple Mail Transfer Protocol*), definito nella RFC 821, è il protocollo che definisce le modalità di dialogo tra le entità coinvolte nello scambio del messaggio.
- Il trasferimento del messaggio avviene tramite un'unica connessione TCP (sulla porta 25) tra il server che invia il messaggio e quello di destinazione.
- L'SMTP non utilizza server intermedi di transito.

4.10

SMTP *Simple Mail Transfer Protocol*

- I messaggi (intesi come PDU a livello di applicazione) scambiati sono di tipo testuale.
- Poiché l'SMTP è stato definito nei primi anni '80, presenta alcune caratteristiche ormai obsolete e limitative, come ad esempio, l'uso della codifica in codice ASCII a 7 bit come formato di testo.
- L'SMTP utilizza connessioni TCP permanenti: quindi ognuna di esse può essere utilizzata per il trasferimento di più messaggi tra due *mail server*.

4.11

SMTP

- L'SMTP è caratterizzato da una fase di *handshake*, dove il Client e il Server si scambiano, prima del trasferimento del vero e proprio messaggio, diverse informazioni.
- Generalmente il Client SMTP invia messaggi strutturati come segue:

Command	SP	Argument	CRLF
---------	----	----------	------

- *Command* (Comando) è il tipo di informazione o l'operazione che il Server SMTP deve compiere o ricevere, ad esempio:
 - » HELO, MAIL FROM, RCPT TO, DATA, QUIT.
- *Argument* (Argomento) è l'eventuale informazione aggiuntiva richiesta dall'istruzione di *command*.
- SP = "spazio", CRLF = Carriage Return Line Feed.

4.12

SMTP

- Il server SMTP generalmente risponde al Client con messaggi così strutturati:

Code	SP	Comment (OPT)	CRLF
------	----	---------------	------

- Dove:
 - *Code* (Codice): è un codice che identifica l'esito del messaggio inviato dal Client o lo stato del Server;
 - *Comment* (Commento): è un campo opzionale dove il Server può fornire un messaggio esplicativo del codice.

4.13

SMTP : esempio di comunicazione Client-Server (1)

- Esempio di comunicazione SMTP fra il Client zeus.com.dist.unige.it e il Server quercia.reti.dist.unige.it :
(C: messaggio inviato dal Client, S: risposta dal Server, in giallo in campo *command*, in azzurro il campo *argument*)
 - » S: 220 quercia.reti.dist.unige.it
 - » C: HELO zeus.com.dist.unige.it
 - » S: 250 Hello zeus.com.dist.unige.it, pleased to meet you
 - » C: MAIL FROM: <bianchi@com.dist.unige.it>
 - » S: 250 bianchi@com.dist.unige.it... Sender OK
 - » C: RCPT TO: <rossi@reti.dist.unige.it>
 - » S: 250 rossi@reti.dist.unige.it... Recipient OK

4.14

SMTP: esempio di comunicazione Client-Server (2)

```

»C: DATA
»S: 354 Enter mail, end with "." on a line by itself
»C: Ciao vecchio mio.      -- messaggio --
»C: Come stai ?           -- messaggio --
»C: .                      -- messaggio --
»S: 250 Message accepted for delivery
»C: QUIT
»S: 221 quercia.reti.dist.unige.it closing connection

```

4.15

Formato del messaggio

- Il formato del messaggio di posta elettronica è definito nella specifica RFC 822.
- Il messaggio risulta composto da due parti:
 - **intestazione**, consiste in una parola chiave seguita dai relativi argomenti:
 - » i campi sono definiti dall’RFC 822;
 - » consente di includere informazioni accessorie al messaggio: nome del mittente (*From*), data di invio (*Date*), priorità (*Priority*), server SMTP attraversati, ecc.;
 - » alcune informazioni sono inserite durante la composizione del messaggio, altre dai server SMTP attraversati dallo stesso.
 - **contenuto** del messaggio o *entity-body*:
 - » contiene il messaggio vero e proprio utilizzando la codifica ASCII a 7 bit.

4.16

POP3 - *Post Office Protocol vers. 3*

- Il POP3 è definito nella RFC 1939.
- Il POP3 è un protocollo di accesso estremamente semplice ma con funzionalità limitate.
- Gli *user agent* implementano il lato *client* di questo protocollo, mentre i mail server implementano quello *server*.
- Il POP3 utilizza come protocollo di trasporto il TCP. Le connessioni sono sempre aperte dal client POP3 sulla porta numero 110 del server.

4.17

POP3

- A connessione TCP avvenuta, il POP3 agisce in tre fasi distinte:
 - **Autorizzazione:** l'utente invia la *login* (*username* + *password*) al Server.
 - **Transazione:** l'utente “scarica” i messaggi, li può contrassegnare per l'eliminazione (un *mail reader* può essere configurato per *download-and-delete* o per *download-and-keep*) e può ottenere delle statistiche sulla posta.
 - **Aggiornamento:** questa fase avviene dopo che il Client ha inviato il comando “quit”; concludendo la transazione il Server elimina tutti i messaggi che erano stati contrassegnati per l'eliminazione.

4.18

POP3

- Le transazioni Client-Server POP3 avvengono con messaggi in codifica ASCII, e appaiono solitamente come nel seguente esempio:

```

- S: +OK POP3 server ready
- C: user alice
- S: +OK
- C: pass hungry
- S: +OK user successfully logged on
- C: list
- S: 1 498 - (id e dimensione dell'email) -
- S: 2 912
- S: .
- C: retr 1

```

Autorizzazione

Transazione

4.19

POP3

```

- 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

```

Transazione

Aggiornamento

- In questo esempio il Client chiede al Server di elencare le dimensioni di ciascun messaggio memorizzato nella mailbox. Successivamente recupera ed elimina ciascun messaggio dal Server.

4.20

POP3

- La sintassi dei comandi, come *list*, *retr*, *dele* e *quit*, e i relativi codici di risposta utilizzati nelle transazioni Client-Server POP3 è definita nella RFC 1939.

4.21

IMAP - *Internet Mail Access Protocol*

- L'**IMAP** (*Internet Mail Access Protocol*), definito dalla RFC 2060, è un protocollo di accesso ai Server di posta elettronica con molte più caratteristiche e, quindi, più complesso del POP3.
- I server IMAP usano connessioni TCP sulla porta 143.
- L'IMAP consente agli utenti di creare le proprie cartelle di posta sul server e di gestirle come se fossero locali.
- In particolare consente di:
 - creare e mantenere cartelle di messaggi all'interno della *mailbox* sul server;
 - ricercare messaggi secondo criteri specifici nelle cartelle remote.

4.22

IMAP

- Una delle cause che rendono l'IMAP molto più complesso del POP3 è data dalla necessità, da parte del Server, di memorizzare e aggiornare una gerarchia di cartelle per ogni utente.
- Un'altra importante caratteristica di questo protocollo è data dalla possibilità di ottenere anche solo alcune parti (componenti) dei messaggi.
- Per esempio, uno *user agent* che accede al *Mail Server* utilizzando IMAP è in grado di estrapolare la sola intestazione dei messaggi o una singola componente di un messaggio MIME "multipart".

4.23

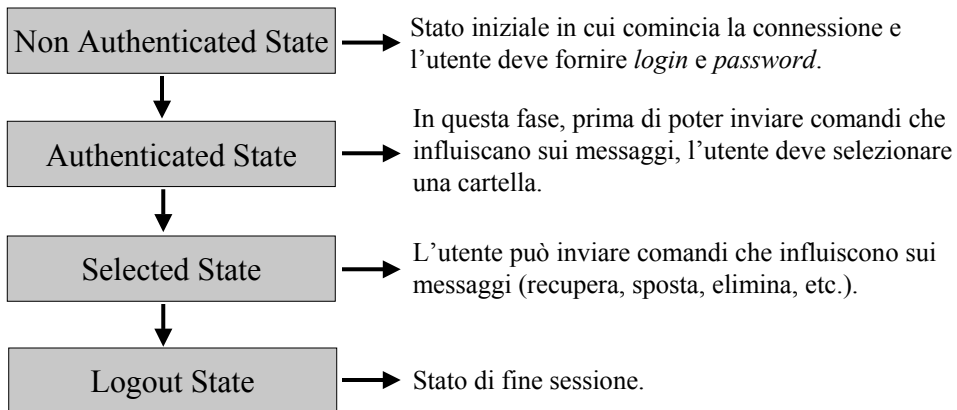
IMAP

- Una sessione IMAP consiste di:
 - una connessione client-server,
 - una fase di *handshake*,
 - interazioni Client-Server,
 - chiusura della connessione.
- Le interazioni IMAP sono strutturalmente simili ma più complesse di quelle POP (comando del Client + risposta del Server).

4.24

IMAP

- Alle fasi della sessione *Client-Server* IMAP è legata una sequenza composta da quattro differenti stati:



4.25

HTTP

- Oggi molti utenti utilizzano il servizio e-mail tramite Web, ossia il Mail Server viene realizzato estendendo opportunamente le funzionalità di un *Web-Server* tramite script o programmi CGI (*Common Gateway Interface*).
- In questo modo è possibile usare come *User Agent* un normale *Web-Browser* e accedere alla propria *mailbox* attraverso transazioni HTTP.
- In questo caso, quindi, il *Web-Browser* opera come interfaccia ed usa l'HTTP (invece che MIME o POP) per comunicare con un *Mail-Web-Server*, che crea pagine dinamiche tramite le quali gli utenti gestiscono la posta.
- Il *Mail-Web-Server* usa comunque SMTP per comunicare con gli altri Mail-Server (Web o meno).

4.26

MIME - *Multipurpose Internet Mail Extension*

- Il MIME (*Multipurpose Internet Mail Extension*) è un'estensione della struttura RFC 822 che elimina alcune limitazioni presenti nello schema SMTP/RFC 822:
 - SMTP non può trasferire direttamente file eseguibili o comunque binari.
 - SMTP non può trasmettere i dati di un testo che comprenda caratteri di una lingua nazionale specifica (presenti in codici a 8 bit o superiori).
 - I Server SMTP possono rifiutare messaggi che superano una certa dimensione.
 - Alcune realizzazioni di SMTP non aderiscono completamente allo standard definito nella RFC 821.

4.27

MIME- *Multipurpose Internet Mail Extension*

- La specifica MIME comprende le seguenti caratteristiche:
 - Sono stati definiti 5 nuovi campi nell'intestazione del messaggio oltre a quelli già presenti ed ereditati dalla RFC 822.
 - Sono stati definiti nuovi formati di contenuto che supportano la multimedialità nella posta elettronica.
 - Sono state definite le codifiche di trasferimento che abilitano la conversione di ogni formato in una forma protetta da eventuali alterazioni introdotte dal sistema di posta.

4.28

MIME: I campi dell'intestazione

- I 5 nuovi campi di intestazione introdotti dal MIME sono:
 - **MIME-Version**: deve assumere un valore 1, che indica che il messaggio è conforme agli RFC.
 - **Content-Type**: Descrive il tipo di dati contenuti nell'*entity-body*.
 - **Content-Transfer-Encoding**: Indica il tipo di codifica utilizzata per rappresentare l'*entity-body* in un modo compatibile al trasporto SMTP.
 - **Content-ID**: Serve ad identificare univocamente le entità MIME in diversi contesti.
 - **Content-Description**: Una descrizione in testo non codificato dell'oggetto contenuto nell'*entity-body*.

4.29

MIME: L'intestazione *Content-Type*

- L'RFC 2046 definisce il formato dell'intestazione *Content-Type*:
 - Content-Type: type/subtype; parameters**
 - **Type**: indica il tipo generico di dati (immagine, ...);
 - **Subtype**: indica il formato per quel tipo di dati (JPEG, ...);
 - **Parameters**: sono i parametri facoltativi:
 - Specificano altri aspetti del contenuto; molti hanno senso solo se associati ad solo ad una singola specifica coppia *type/subtype*.
 - I nuovi *set* di *type/subtype* vanno registrati presso lo IANA (Internet Assigned Numbers Authority).
 - Il processo di registrazione è definito nella RFC 2048.

4.30

MIME: L'intestazione *Content-Type*
Type/Subtype

<i>Type</i>	<i>Subtype</i>	<i>Type</i>	<i>Subtype</i>
text	plain	multipart	mixed
	HTML		
image	jpeg		parallel
	gif		alternative
video	mpeg		digest
audio	basic		RFC822
	32kadpcm		
application	Postscript Adobe	message	partial
	Dati binari a 8 bit		external body

4.31

MIME: L'intestazione *Content-Type*
Il contenuto dell'Entity-body (1)

- In presenza di un *entity-body* di tipo ***text/plain*** non è necessario alcun software particolare per decifrare il testo.
 - In questo caso il campo *parameters* può essere utilizzato per specificare il set di caratteri da utilizzare o utilizzato:
 - » Content-Type: ***text/plain; charset="ISO-8859-1"***
 - » Content-Type: ***text/plain; charset="us-ascii"***
- I tipi ***image***, ***video*** e ***audio*** ovviamente indicano che l'*entity-body* contiene rispettivamente un'immagine, un video o un file audio.

4.32

MIME: L'intestazione *Content-Type*

Il contenuto dell'Entity-body (2)

- Il tipo *application* fa riferimento a tipologie di dati binari da non interpretare o ad informazioni che devono essere elaborate da una applicazione specifica.
- Il tipo *multipart* indica che il corpo del messaggio contiene diverse parti indipendenti.
 - Il campo *Content-Type* dell'intestazione include un parametro, detto *Boundary*, che indica il limite tra le diverse parti dell'*entity-body*.
Content-Type: multipart/mixed; Boundary="StartOfNextPart"
 - Nel testo le diverse parti sono distinte con una linea di tipo
`--StartOfNextPart`
 - La fine del messaggio è indicata con una riga
`--StartOfNextPart--`

4.33

MIME: L'intestazione *Content-Type*

Il contenuto dell'Entity-body (3)

- Ci sono 4 sottotipi del tipo *multipart* e ognuno di essi ha la stessa sintassi generale:
 - Il sottotipo *multipart/mixed* è impiegato quando ci sono diverse parti indipendenti che devono essere collocate secondo un ordine particolare;
 - Il sottotipo *multipart/parallel* è impiegato quando l'ordine delle parti non è significativo;
 - Nel sottotipo *multipart/alternative* le varie parti sono diverse rappresentazioni delle stesse informazioni da utilizzare in alternativa;
 - Il sottotipo *multipart/digest* è impiegato quando quando ognuna delle parti è un messaggio RFC 822 con relative intestazioni.

4.34

MIME: L'intestazione *Content-Type*

Il contenuto dell'Entity-body (4)

- Il tipo *message* consente di realizzare diverse funzionalità a seconda del *sub-type* ad esso seguente:
 - **RFC822**: l'*entity-body* è esso stesso un messaggio (RFC822 o MIME) incapsulato.
 - **Partial**: usato per la frammentazione di messaggi di grosse dimensioni.
 - » I parametri specificano le informazioni necessarie per ricostruire il messaggio originario a destinazione.
 - **External-Body**: contiene un puntatore ad un oggetto che si trova altrove.
 - » I parametri specificano la locazione dell'oggetto da includere.

4.35

MIME: L'intestazione *Context-Transfer-Encoding*

- Lo standard MIME definisce attualmente due metodi per la codifica dei dati:
 - *quoted-printable*;
 - *base64*.
- In realtà, il campo *Context-Transfer-Encoding* può assumere ulteriori 4 valori:
 - **7bit**: i dati sono rappresentati da brevi linee di caratteri ASCII;
 - **8bit**: possono essere presenti caratteri ASCII codificati a 8 bit (bit più significativo a 1);
 - **binary**: oltre a essere presenti caratteri non-ASCII le linee potrebbero non essere brevi quanto necessario per il trasporto tramite SMTP;
 - **x-token**: utilizzato per codifiche proprietarie.
- I valori *7bit*, *8bit* e *binary* indicano che non è stata applicata nessuna codifica specifica (da MIME), ma forniscono alcune indicazioni di massima sulla natura dei dati.

4.36

MIME: L'intestazione *Context-Transfer-Encoding*
Codifiche di trasferimento

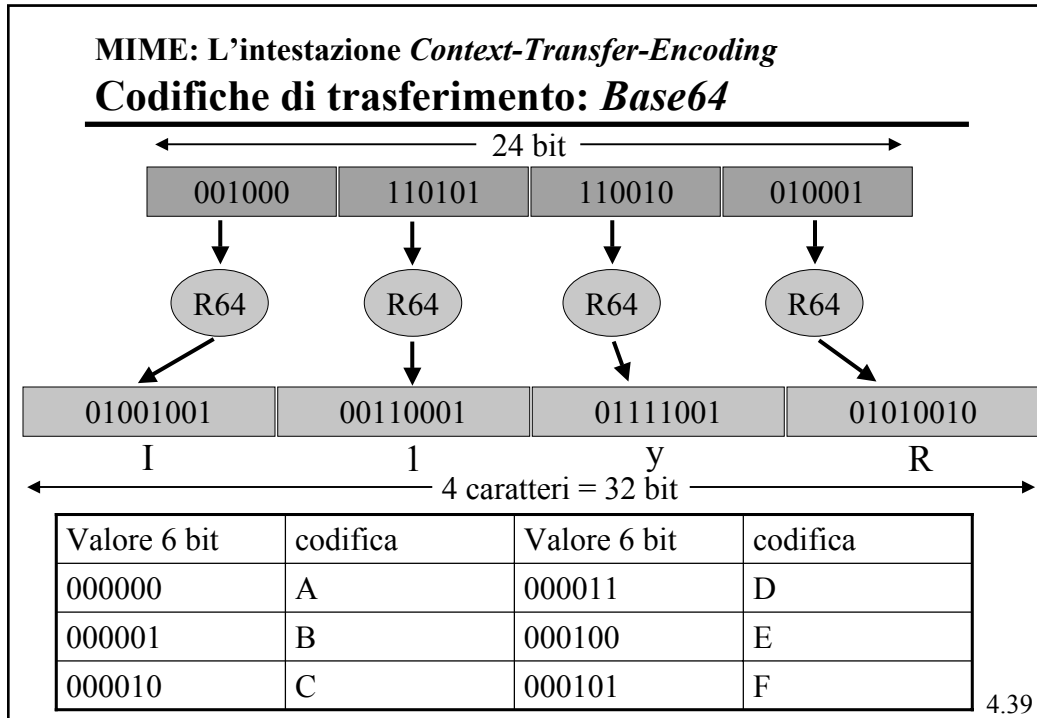
- Il valore *quoted-printable* indica l'uso di una particolare codifica volta a mantenere la leggibilità di testi composti principalmente (in origine) da caratteri ASCII.
- La codifica di trasferimento *base64* è comunemente impiegata per i dati binari generici, in modo da renderli trasparenti alle elaborazioni da parte dei programmi di trasporto della posta elettronica.
- Questa tecnica codifica i dati binari in ingresso in caratteri "visualizzabili".

4.37

MIME: L'intestazione *Context-Transfer-Encoding*
Codifiche di trasferimento: *Base64*

- La codifica *Base64* presenta le seguenti importanti caratteristiche:
 - L'intervallo dei valori forniti dalla funzione di codifica è un insieme di caratteri universalmente rappresentabile.
 - L'insieme dei caratteri è formato da 65 caratteri visualizzabili, uno dei quali serve da riempimento "=". Avendo 64 caratteri disponibili, ognuno di essi viene usato per rappresentare 6 bit del contenuto originario.
 - Nessun carattere di controllo è incluso nell'insieme.
 - Il carattere "-" non è usato perché nel formato RFC 822 ha un significato specifico.

4.38



Esempio di messaggio in ricezione (1)

```
Return-Path: <cyrus@quercia.reti.dist.unige.it>
X-Sieve: cmu-sieve 2.0
Return-Path: <lelus@reti.dist.unige.it>
Received: from bacco (unknown [130.251.8.154])
  by quercia.reti.dist.unige.it (Postfix) with SMTP id 8FC58BC75
  for <matteo@reti.dist.unige.it>; Fri, 21 Mar 2003 12:23:26 +0100 (CET)
From: "Raffaele Bolla" <lelus@reti.dist.unige.it>
To: "Matteo Repetto" <matteo@reti.dist.unige.it>
Subject: l'ultima slide ha l'intestazione MIME!
Date: Fri, 21 Mar 2003 12:27:51 +0100
Message-ID: <001001c2ef9c9e84cf7509a08fb82@com.dist.unige.it>
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="Boundary_(ID_lNI7PoiA/9T4Nh5ShGyDmw)"

--Boundary_(ID_lNI7PoiA/9T4Nh5ShGyDmw)
Content-type: text/plain; format=flowed; charset=us-ascii
Content-transfer-encoding: 7BIT
Testo del messaggio...
--Boundary_(ID_lNI7PoiA/9T4Nh5ShGyDmw)
Content-type: image/gif; name=logo.gif
Content-transfer-encoding: BASE64
Content-disposition: inline; filename=logo.gif
R0lGODlhAA5uAecAAAIcAt7y9hYcApLH4jgKcKGo2QSEml2dxGSyRCoxoIWFhik
vxCKw4JmS6YaGhqEtapaWiJ0mmdpYz19l/pGRmJiXdWWl1pXbEVxgHZeR7YeHvp0
maSSszLzPjNPQhml1FNSWaVVV2KZpZ2bctml11+CGaaY5JZpplnopmmmmuy2aab
b8IZp5xz0lmnnXfimaeee/J5Y0AAOw==
--Boundary_(ID_lNI7PoiA/9T4Nh5ShGyDmw) --
```

4.40

Esempio di messaggio in ricezione (2)

```

Return-path: <enrico@unige.it>
Date: Mon, 24 Feb 2003 17:44:07 +0100
From: enrico <enrico@unige.it>
Subject: Offerta
X-Sender: C0340@mail.unige.it
To: Raffaele Bolla <raffaele.bolla@unige.it>
Message-id: <5_2_0_9_0_20030224174112.019cfd80@box.tin.it>
MIME-version: 1.0
X-Mailer: QUALCOMM Windows Eudora Version 5.2.0.9
Content-type: multipart/mixed; boundary="Boundary_(ID_y5M7R9QVCWgB9amJYydjuA)"
--Boundary_(ID_y5M7R9QVCWgB9amJYydjuA)
Content-type: text/plain; charset=iso-8859-1; format=flowed
Content-transfer-encoding: QUOTED-PRINTABLE
Il Netgear non =E8 pi=F9 disponibile su TechData.
Il costo dell'apparato su Esprinet (www.esprinet.com) =E8 di 200=80 +=
IVA. (+=20
6=80 di spedizione)
ciao
Enrico
--Boundary_(ID_y5M7R9QVCWgB9amJYydjuA)
Content-type: application/msword; name=ADSL.doc;
x-mac-type=42494E41; x-mac-creator=4D535744
Content-transfer-encoding: BASE64
Content-disposition: attachment; filename=ADSL.doc
0M8R4KGxGuEAAAAAAAAAAAAAAAAAAAAAPgADAP7/CQAGAAAAAAAAAAAAAAAABAAAA
TgAAAAAAAAAAEAAUAAAAEAAAD+////AAAAAE0AAAD==
--Boundary_(ID_y5M7R9QVCWgB9amJYydjuA)--

```

4.41