

Telnet e i protocolli di posta elettronica

Protocolli di Posta elettronica e telnet

Sommario

Protocolli di Posta elettronica e telnet.....	1
Scopo	2
Materiale occorrente.....	3
SMTP	3
I principali comandi SMTP.....	3
Fasi di una sessione SMTP	4
Uso di telnet per l'invio di posta	4
Esempio di uso di tutti i comandi SMTP durante una sessione.....	5
Invio di una mail testuale a più destinatari	5
Analisi della sessione precedente.....	5
Email ricevute	6
SMTP e MIME	7
Come scrivere un e-mail con HTML	7
Sessione Telnet per inviare una mail con contenuto di tipo HTML.....	7
Mail ricevuta	8
Standard MIME	8
Righe di intestazione MIME	9
Content-Transfer-Encoding:	9
Content-Type.....	10
Messaggi contenenti più parti MIME.....	10
Sessione Telnet per l'invio di email contenente diversi oggetti.....	11
L'email ricevuta.....	12
Protocolli di Accesso alla Mail	13
POP3	13
IMAP	13
HTTP (WebMail).....	13
Connettersi ad un server POP3 usando telnet.....	14
Fase dopo fase: il significato dei comandi previsti dal protocollo.	14
Una sessione di ricezione	17

Questa dispensa ha solo scopo didattico.

Telnet e i protocolli di posta elettronica

Scopo

Apprendere l'uso del protocollo SMTP (per inviare le mail) e POP3 (per ricevere le mail), utilizzando il telnet (protocollo a livello di applicazione per realizzare delle connessioni con macchine in remoto. Dà quindi la possibilità di essere fisicamente sul proprio sistema e lavorare su un altro che può essere nella stanza accanto come dall'altra parte del mondo, la propria macchina (locale) diventa così un terminale remoto: tecnicamente l'operazione è definita *Remote Login*. NB: telnet è un protocollo obsoleto, poiché tutti i comandi vengono inviati lungo la rete in chiaro, i dati possono quindi essere intercettati ecco perchè si preferisce l'uso del protocollo SSH che ha lo stesso scopo del telnet l'intera comunicazione (ovvero sia l'autenticazione che la sessione di lavoro) avviene in maniera cifrata. Utilizza quindi SSL)

Per usufruire del comando Telnet da ambienti linux, unix, mac osx, windows con dos aprire un terminale (ovvero una shell, ovvero una riga di comando, ovvero una finestra dos) e digitare la parola *telnet*.

Poiché telnet è obsoleto, è potrebbe essere stato disabilitato nel S.O. Windows.

Per **abilitarlo** è sufficiente andare in: *Pannello di controllo -> Programmi e funzionalità -> Attivazione e disattivazione delle funzionalità Windows*. Mettere il flag alla voce: "Client Telnet".

Chi volesse utilizzare un telnet più evoluto può scaricare **putty** o ancor meglio **CRT** (quest'ultimo non è software open source)

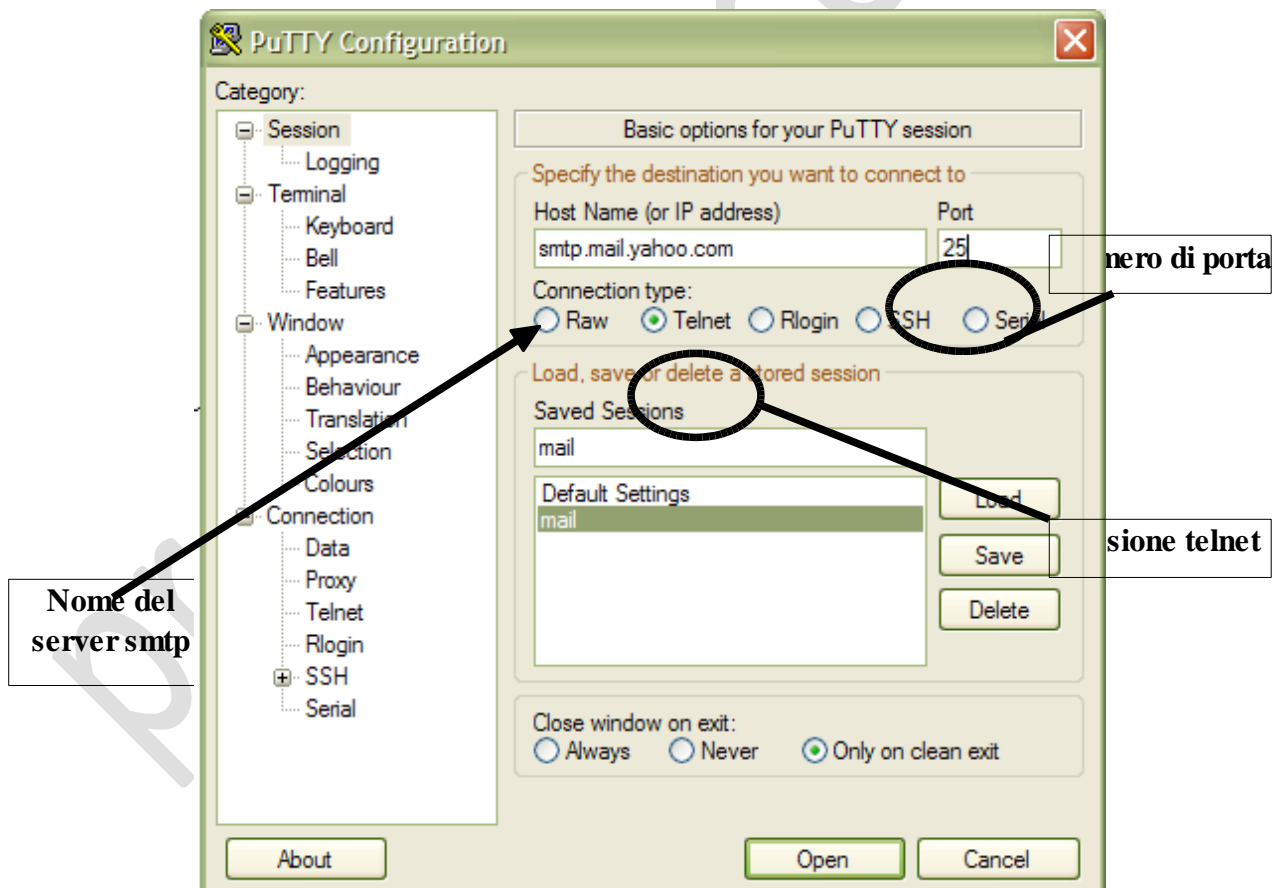


figura 1: Sessione Telnet sulla porta 25 con PUTTY

Telnet e i protocolli di posta elettronica

Materiale occorrente

pc ^_-
connessione ad Internet
un client telnet
account di posta elettronica
un programma per la codifica in us-ascii (spesso indicato come mime64)
indirizzi dei server di posta - (di seguito un link all'uopo)
<http://www.sandrodiremigio.com/documenti/tutorials/pop3smtpnntp.htm>

SMTP

Simple Mail Transfer Protocol (SMTP) è il protocollo utilizzato per trasmettere messaggi di posta elettronica tra due host. SMTP utilizza il protocollo di trasporto **TCP**, ed in particolare un SMTP server rimane costantemente in ascolto sulla **porta 25**. Il server SMTP si occupa poi di trasferire i messaggi nelle caselle di posta (*mailbox*) dei destinatari.

La sintassi dei comandi è case-insensitive, ed è composta da istruzioni seguite da uno o più parametri terminate da un CRLF (Invio).

Il protocollo è descritto nella **RFC 821**, ma lavora in stretta collaborazione con altri standard come la **RFC 822** che descrive la sintassi degli headers della mail, la **RFC 1049** che definisce le strutture dati per interpretare correttamente il contenuto delle mail

Lo standard definito dalla **RFC 821**, ha però diversi limiti riguardanti per esempio la dimensioni dei messaggi oppure la trasmissione di mail e impone il vincolo al solo del set di caratteri US-ASCII (costituito dalle 26 lettere dell'alfabeto minuscole e maiuscole, 10 cifre decimali il simbolo + e lo /) È stato necessario, quindi, estendere il protocollo con lo standard **MIME** (Multipurpose Internet Mail Extensions). definito dalle RFC 2045, 2046, 2047, 2048 e 2049, che permette di definire come deve essere interpretato il contenuto di un messaggio e-mail. In questo modo un messaggio può essere opportunamente codificato per contenere anche informazioni diverse dai caratteri ASCII standard.

I principali comandi SMTP

HELO: Identifica il client SMTP al server SMTP;
EHLO: E' possibile usare anche questo comando per identificarsi, se il server supporta le SMTP Service Extensions risponderà in modo positivo altrimenti con un errore di tipo 500 (*Syntax Error*);
MAIL FROM: <indirizzo mittente>: Indicata la mailbox del mittente del messaggio;
RCPT TO: <indirizzo destinatario> : Indica la mailbox del destinatario (*Recipient*). E' possibile specificare attraverso molteplici RCPT TO diversi destinatari;
DATA: Indica al server che quanto digitato successivamente saranno i dati del messaggio di posta;
RESET: Annulla i comandi (*Reset*) precedentemente inviati nella sessione SMTP corrente;
VRFY <stringa>: Chiede al server se la stringa di testo immessa rappresenta un nome utente presente ed in tal caso visualizza l'intero indirizzo;
HELP: Visualizza i comandi disponibili sul server;
NOOP: Non esegue nessuna operazione restituisce solo un messaggio 250 (Ok) se il server risponde;
QUIT: Termina la sessione SMTP corrente;

Telnet e i protocolli di posta elettronica

Fasi di una sessione SMTP

Una sessione SMTP attraversa almeno sei fasi:

- 1 Il client SMTP contatta il server sulla porta TCP 25. Se questo è in ascolto e la connessione è accettata risponde con un messaggio 220 (Ready);
- 2 Il client chiede di stabilire la sessione SMTP inviando il comando HELO seguito dal FQDN (Fully Qualified Domain Name). Se il server accetta risponde con un messaggio 250 (Ok);
- 3 Il client indica il proprio indirizzo tramite il comando MAIL FROM: <indirizzo mittente>. Il server risponde con 250 (Ok) per ogni destinatario accettato;
- 4 Successivamente il client indica al server i **destinatari** del messaggio tramite RCPT TO: <indirizzo destinatario> ed il server risponde **per ogni destinatario** accettato un codice 250 (Ok);
- 5 Il client comunica al server l'intenzione di scrivere il corpo del messaggio con DATA. Il server risponde con un codice 354 e indica come marcare il termine del messaggio. I campi come Date, Subject, To, Cc, bcc, From vanno inseriti tra i dati della mail;
- 6 Completato il messaggio da scrivere tramite "." il server memorizza la mail. A questo punto è possibile, scrivere un nuovo messaggio oppure inviare il comando QUIT, dopo il quale il server invia i messaggi e risponde con un codice 221 (Closing) e la connessione TCP viene terminata;

Ad oggi quasi tutti i server SMTP per motivi di sicurezza supportano le SMTP Service Extensions definite nella RFC 1425 che obbligano l'autenticazione dell'utente (comando EHLO).

Uso di telnet per l'invio di posta

Allora per stabilire la connessione TCP con il nostro server basta mandare in esecuzione il client Telnet prescelto inserendo il nome e il numero della porta del servizio desiderato.

Le porte permettono di accedere a servizi diversi allo stesso server.

Elenchiamo i servizi più comuni:

Trasferimento file (ftp) porta 21
Telnet porta 23
Invio della posta (smtp) porta 25
Pagine web (http) porta 80
Ricezione posta (pop) porta 110
Ricezione posta (imap) porta 143
A noi serve la 25 naturalmente...

La sintassi per connettersi ad un servizio qualsiasi è la seguente:

```
telnet provider.it porta
per il nostro percorso utilizzerò l'account di posta elettronica indirizzoAperdere@ymail.com
che si trova nel server smtp di indirizzo smtp.mail.yahoo.com, per cui io devo stabilire una
connessione TCP con tale server sulla porta 25, il comando diventa perciò:
```

```
telnet smtp.mail.yahoo.com 25
```

Telnet e i protocolli di posta elettronica

Esempio di uso di tutti i comandi SMTP durante una sessione

```
220 smtp108.plus.mail.re1.yahoo.com ESMTP
helo smtp.mail.yahoo.com
rset
250 flushed
helo smtp.mail.yahoo.com
250 smtp108.plus.mail.re1.yahoo.com
noop
250 ok
```

Invio di una mail testuale a più destinatari

```
ehlo
250-smtp108.plus.mail.re1.yahoo.com
250-AUTH LOGIN PLAIN XYMCOOKIE
250-PIPELINING
250 8BITMIME
auth login
334 VXNlcm5hbWU6
aW5kaXJpenpvQXB1cmRlcmVAeW1haWwuY29t
334 UGFzc3dvcmQ6
password codificata
235 ok, go ahead (#2.0.0)
mail from: <indirizzoAperdere@ymail.com>
250 ok
vrify flyct@libero.it
252 send some mail, i'll try my best
rcpt to: <lascuolainrete@gmail.com>
250 ok
rcpt to: <indirizzoInesistente@hostfalso.net>
250 ok
rcpt to: <flyct@libero.it>
250 ok
data
354 go ahead
from: sconosciuto
to: a te no?
subject: stiamo giocando

quanto digitato dopo la linea data e prima della riga vuota costituisce
l'header
del messaggio di posta elettronica, questo è il corpo del messaggio, la mail
ha
fine introducendo una nuova riga che contiene un punto.
.
250 ok 1233081474 qp 6445
```

Analisi della sessione precedente

In grassetto trovate tutti i comandi SMTP usati (mediante il client SMTP) durante la connessione con il lato server SMTP dell'host smtp.mail.yahoo.com, i comandi da ehlo fino alla risposta del server **235 ok, go ahead (#2.0.0)** servono per autenticarsi.

Il comando **rcpt to: <indirizzodestinatario@mailservername.tld>** comunica al server un destinatario

Telnet e i protocolli di posta elettronica

per il messaggio che scriveremo subito dopo il comando **data**, in questo esempio i destinatari del messaggio sono 3 e precisamente : lascuolainrete@gmail.com , indirizzoInesistente@hostfalso.net, flyct@libero.it.

Il messaggio di posta elettronica è contenuto tra le righe **data** e la riga con il punto .
In particolare i comandi **from** e **to** sono obbligatori per qualsiasi client di posta elettronica e in questo caso specifico avreste pure il campo **cc**: con gli indirizzi degli ultimi 2 destinatari.
Nel nostro caso, i dati fittizi di questo esempio saranno visualizzati dai nostri destinatari come mostrato in seguito (fig. 2).

Il secondo indirizzo è chiaramente inesistente, ragion per cui, il mio server smtp mi comunicherà mediante un messaggio che non è riuscito ad inviare tale mail (fig. 3).

Il corpo del messaggio si scrive subito dopo una CRLF (Carriage Return /Line Feed) e finirà con un punto.

Email ricevute

Di seguito l'email ricevuta dall'indirizzo lascuolainrete@gmail.com

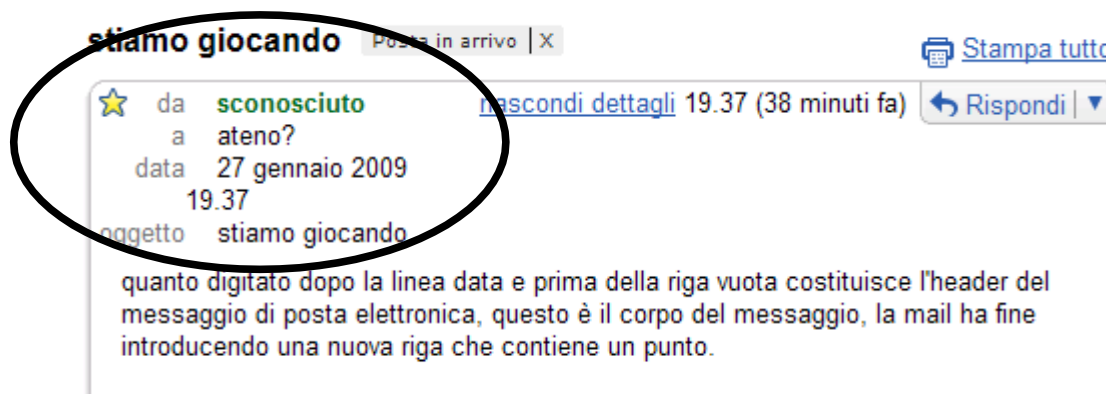


Figura 2: Email ricevuta - nota i campi a e da

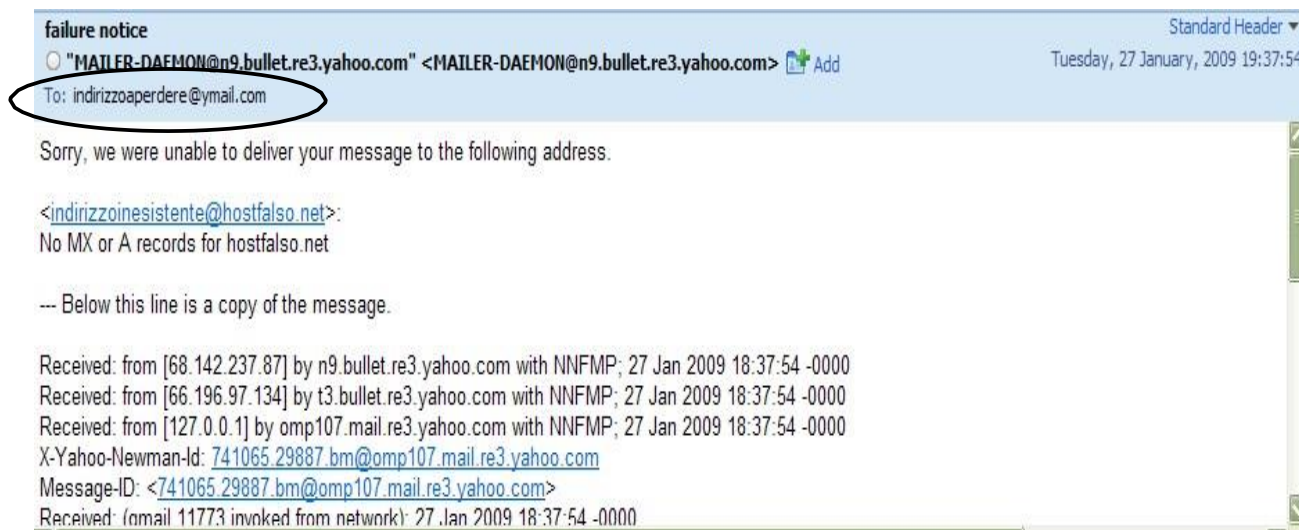


Figura 3: Email che notifica la mancata ricezione del messaggio- nota il campo To:

Da notare che il server SMTP di yahoo aggiunge correttamente il campo to: anche se il campo from: del destinatario è sconosciuto. PERCHÈ????

SMTP e MIME

Come scrivere un e-mail con HTML

Stabiliamo una connessione TCP con nostro server sulla porta 25 come fatto in precedenza e copiamo il nostro messaggio precedentemente preparato in un file.

Tale file deve contenere le righe di intestazione del messaggio di posta

```
Received: by mioprovider.it id AB65332 with SMTP;Thu 28 Jan 2009 00:00:00
+0200
Message-ID:
To:
Date:
Subject: Oggetto dell'e-mail
MIME-Version: 1.0
```

la riga Received:by... a mioprovider dovete scrivere lo stesso specificato con helo

Quel AB65332 si può cambiare come vi pare basta che sia una qualcosa di simile, è possibile anche modificare la data secondo il formato indicato nell'esempio (lasciare +0200), nella riga **Date:**

Ricordate che la riga **Received: by mioprovider.it id....** viene aggiunta al messaggio recapitato, dal lato server del server SMTP contenente la mailbox del destinatario.

è ovvio che l'indirizzo e-mail specificato in From deve coincidere a quello specificato in rcpt to:... ma come visto nell'esempio di poco fa, per far apparire sul client di posta il nome collegato all'e-mail piuttosto che l'indirizzo vero è proprio basta scrivere

From: Biancaneve

Apparirà come mittente la scritta Biancaneve. Stessa cosa per il destinatario con To:....

Vediamo ora come scrivere un e-mail con HTML.

Per utilizzare i tag HTML all'interno di un e-mail di questo tipo, è sufficiente aggiungere le seguenti righe appena dopo MIME-Version: 1.0

```
Content-Type: text/html; charset=us-ascii
Content-Transfer-Encoding: 7bit
```

Lasciare una riga vuota e inserite i tag HTML .

Sessione Telnet per inviare una mail con contenuto di tipo HTML

Questo è il contenuto della mia sessione per spedire l'email che vedrete nella figura 4.

```
220 smtp106.plus.mail.rel.yahoo.com ESMTP
ehlo
250-smtp106.plus.mail.rel.yahoo.com
250-AUTH LOGIN PLAIN XYMCOOKIE
250-PIPELINING
250 8BITMIME
auth login
334 VXN1cm5hbWU6
aW5kaXJpenpvQXB1cmR1cmVAeW1haWwuY29t
334 UGFzc3dvcmQ6
vostra password codificata in US-Ascii
235 ok, go ahead (#2.0.0)
mail from: <indirizzoAperdere@ymail.com>
250 ok
rcpt to: <lascuolainrete@gmail.com>
250 ok
```


Telnet e i protocolli di posta elettronica

data

354 go ahead

```
Received: by smtp.mail.yahoo.com id AB65332 with SMTP; Thu 28 Jan 2009
00:00:00 +0200
  Message-ID: 3
  To: Cenerentola
  From: Biancaneve
  Date: Thu 28 Jan 2009 00:00:00 +0200
  Subject: Andiamo al ballo
  MIME-Version: 1.0
  Content-Type: text/html; charset=us-ascii
  Content-Transfer-Encoding: 7bit
```

Righe di intestazione

```
<HTML>
<HEAD>
  <TITLE>Email</TITLE>
</HEAD>
<BODY BGCOLOR=#F0F0F0>
<CENTER><H1>Questo &egrave; una fake mail </H1></CENTER>
<FONT SIZE=+2>
Scritta non indentata
<UL>
  Vi state divertendo?<BR>
  mi auguro di si!<P>
</UL>
<UL>
  <LI>papaveri</LI>
  <LI>papere</LI>
</UL>
<CENTER>bye</CENTER>
</FONT>
<HR WIDTH="80%">
</BODY>
</HTML>
```

•
250 ok 1233096854 qp 39028

Mail ricevuta

Ed ecco l'email recapitata dal nostro destinatario:



Figura 4: Email con codice HTML

Standard MIME

Come avete potuto notare, per l'invio di questa email che prevedeva testo in codice differente dall'US-ASCII, abbiamo dovuto introdurre le righe:

Telnet e i protocolli di posta elettronica

```
MIME-Version: 1.0
Content-Type: text/html; charset=us-ascii
Content-Transfer-Encoding: 7bit
```

Righe di intestazione

secondo lo standard MIME che ridefinisce il formato del body descritto nell'RFC822 al fine di includere dati non-testuali e parti strutturate.

Con lo standard MIME è possibile, infatti, inserire in un qualsiasi messaggio di e-mail, oltre al testo, anche files contenenti immagini, segnali audio e video; il software che gestisce la posta non si preoccupa del contenuto del messaggio, è l'utilizzatore finale a preoccuparsi della sua opportuna decodifica in base alle specifiche di tipo inserite nel messaggio stesso. Righe di intestazione MIME

MIME version:

- Identifica la versione dello standard MIME usato nel messaggio.

Content-Transfer-Encoding:

- serve a specificare in che modo avviene la trasformazione delle informazioni stabilite nel campo `Content-type :`, per le esigenze legate al trasferimento del messaggio.

In pratica si tratta di indicare una parola chiave che chiarisca come interpretare il contenuto del messaggio al momento della ricezione.

I valori di `Content-Transfer-Encoding` sono: **7bit, 8bit, binary, quoted-printable, base64.**

7bit, 8bit, binary

Questi valori stanno a significare che nessuna operazione di codifica è stata effettuata sul contenuto del messaggio e, allo stesso tempo, forniscono una indicazione sul tipo di dati contenuti nel messaggio stesso (quindi forniscono un'indicazione sul tipo di codifica che potrebbe rendersi necessaria per trasmettere il messaggio in determinati sistemi di trasmissione).

Il valore "**7bit**" significa che in questo caso i dati possono essere rappresentati in gruppi di sette bit, ognuno dei quali rappresenta un carattere ASCII; questo è anche il valore assunto come default se il campo non viene specificato.

Il valore "**8bit**" significa che possono essere presenti caratteri non appartenenti al set ASCII; cioè, suddividendo il messaggio in linee di 8 bit ciascuna e associando ad ogni linea un carattere ASCII, si possono ottenere delle sequenze di caratteri apparentemente senza significato.

Il valore "**binary**" indica che il contenuto del messaggio è in formato binario (un'immagine, un file audio, ecc.). In generale è impossibile trasmettere messaggi di questo tipo.

quoted-printable

I caratteri che richiedono l'uso di 8 bit, si rappresentano nella forma `=hh`, dove la coppia `hh` rappresenta un numero esadecimale, corrispondente al codice del carattere.

Lo scopo principale di questa codifica è di mettere i dati in un formato che difficilmente subirà delle trasformazioni da parte dei vari sistemi che è costretto ad attraversare, prima di giungere a destinazione.

base64

Questo valore significa che sui dati è stata effettuata un'operazione di codifica, detta base64; con questa operazione il messaggio viene trasformato in una sequenza di caratteri appartenenti ad un sottogruppo del set di caratteri ASCII (le lettere

Telnet e i protocolli di posta elettronica

maiuscole da "A" a "Z", quelle minuscole da "a" a "z", I numeri da "0" a "9", il carattere "+" ed il carattere "\").

x-token

Viene usato per specificare uno schema di codifica esterno, non standard, scelto da chi trasmette il messaggio (*token* coinciderà col nome dato a questa codifica); si deve fare attenzione al fatto che questa codifica deve essere nota anche a chi riceve il messaggio, in modo che questo possa essere ricostruito correttamente

Content-Type:

Specifica il tipo ed il sottotipo di dati contenuti nel messaggio (MIME type), in modo che il software che riceve il messaggio possa immediatamente capire come sono codificati i dati ricevuti.

Questo campo ha la forma: **Content-Type: tipo/sottotipo; [parametro]**

dove *tipo* specifica la forma generale dei dati, mentre *sottotipo* specifica il particolare tipo di dati trasmessi. Il campo *parametro* è opzionale.

Esempi di tipi ammessi sono:

text (*possibili sottotipi* ---> css, html, plain, xml...)

audio (*possibili sottotipi* ---> mpeg,mp4, 3gpp, tone ...)

video (*possibili sottotipi* ---> 3gpp, mp4, mpeg, JPEG,BMPEG...)

image (*possibili sottotipi* ---> gif, jpeg, png, tiff..)

application (*possibili sottotipi* ---> javascript, msword, pdf, rtf, xml, zip,...)

multipart (*possibili sottotipi* ---> alternative,digest, parallel, **mixed**).

Tutti i tipi di dati MIME, specificati nel campo *Content-Type*, devono essere registrati presso la IANA (Internet Assigned Numbers Authority); i nuovi tipi di dati, non ancora ufficialmente riconosciuti dalla IANA devono essere indicati con "x-",come ad esempio multipart/x-mixed, oppure application/x-http-cgi.

Messaggi contenenti più parti MIME

Il tipo MIME *multipart* prevede la presenza di più componenti separate, con altrettante intestazioni specifiche. In questo caso si indica comunemente il confine tra una componente e l'altra attraverso una stringa particolare (di solito creata in modo da essere univoca), dichiarata con l'opzione *boundary="stringa"* nel campo *Content-Type* :

Le parti del messaggio sono separate da un limitatore: una stringa univoca detto "Boundary". Ogni boundary è preceduto da due trattini (--). Soltanto l'ultimo boundary è anche terminato dai due trattini, allo scopo di indicare il termine del messaggio.

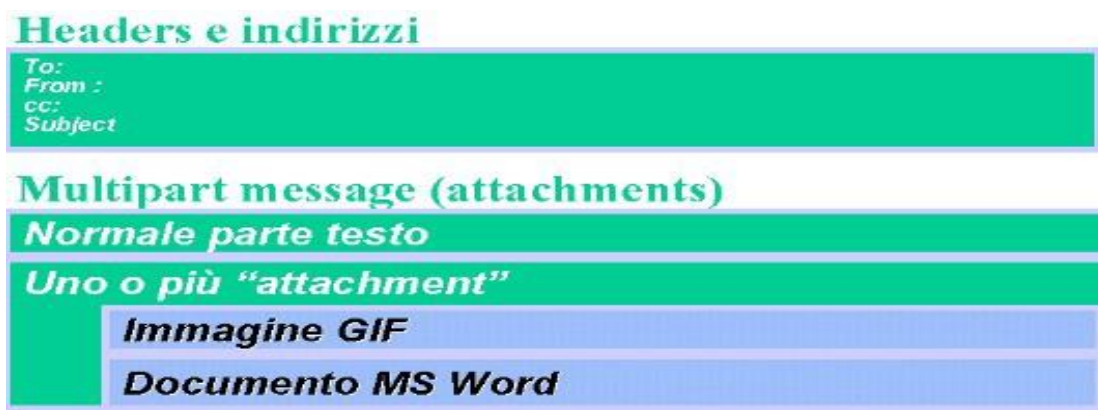
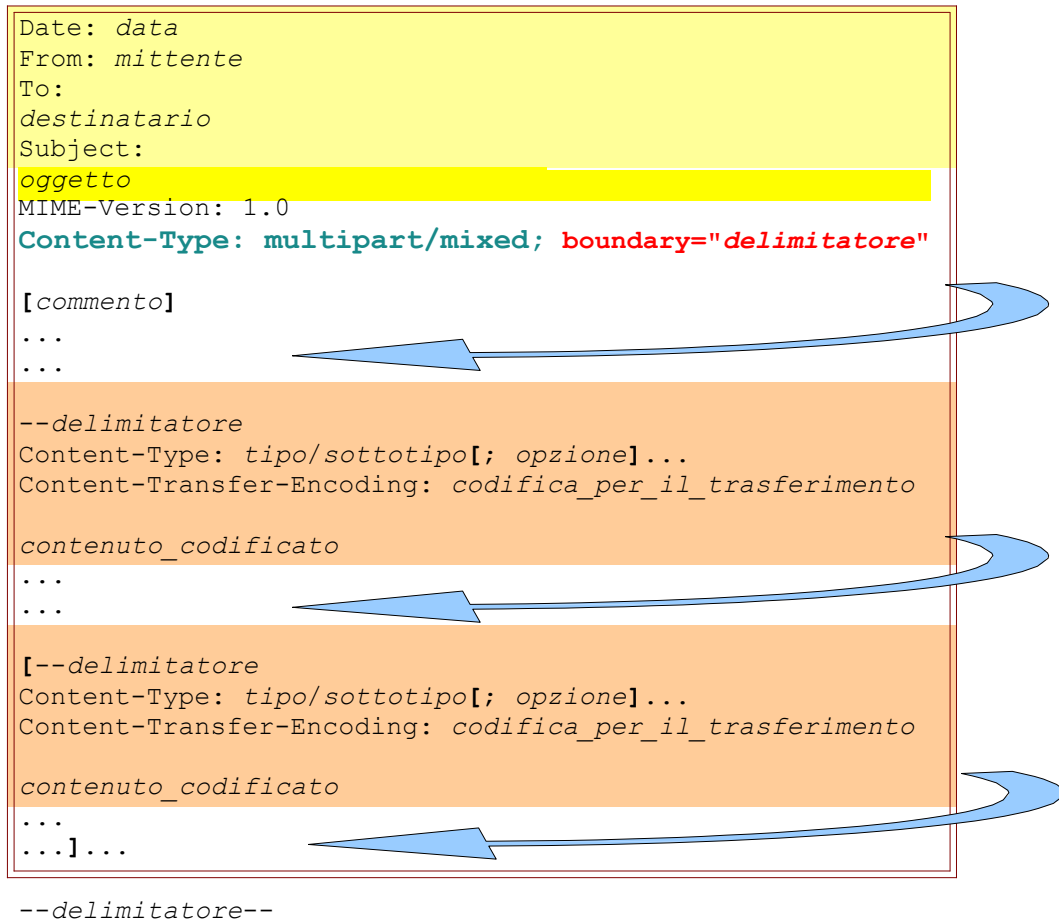


Figura 5: Struttura Logica di un messaggio MIME multiparte

Telnet e i protocolli di posta elettronica

Volendo schematizzare la struttura:



Un caso particolare di messaggi `multipart` è quello che consente di trasmettere il contenuto in forme alternative, come quando si affianca un messaggio in forma testuale a una copia più appariscente in formato HTML. In tal caso si aggiunge il sottotipo `alternative`:

```
Content-Type:multipart/alternative; boundary="xxxx"
```

Sessione Telnet per l'invio di email contenente diversi oggetti

Adesso siamo in grado di inviare una mail contenente del testo e 2 allegati un'immagine e un file zippato.

Chiaramente tutto i file devono essere prima convertiti secondo la codifica a 64bit, per questo basta utilizzare il programma in allegato alla dispensa.

Di seguito l'email spedita utilizzando una sessione in telnet dove è stata stabilita la solita connessione TCP sulla porta 25 del server `smtp.mail.yahoo.com`:

```
220 smtp108.plus.mail.re1.yahoo.com ESMTP
ehlo
250-smtp108.plus.mail.re1.yahoo.com
250-AUTH LOGIN PLAIN XYMCOOKIE
250-PIPELINING
250 8BITMIME
auth login
334 VXNlcm5hbWU6
password codificata
235 ok, go ahead (#2.0.0)
mail from: <indirizzoAperdere@ymail.com>
250 ok
rcpt to: <lascuolainrete@gmail.com>
250 ok data
354 go ahead
Received: by smtp.mail.yahoo.com id AB34567 with SMTP; Sat 1 Jun 2009 12:38:54 +0200
Message-ID: 4
```

Telnet e i protocolli di posta elettronica

To: Tizio
From: Caio
Date: Sat 1 Jun 2009 12:38:54 +0200
Subject: Menù della festa
MIME-Version: 1.0
content-Type: multipart/mixed; boundary=" _NEXPART_ "
-- _NEXPART_

Content-Type: text/plain; charset="ISO-8859-1" Content-Transfer-Encoding: quoted-printable

Ecco il menù previsto per la tua festa!! Bye, Francesca.

-- _NEXPART_
Content-Type: image/jpeg; name="menu.jpeg" Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="menu.jpeg"

9j/4AAQSkZJRgABAQEASABIAAD/4QPYRXhpZgAASUkqAAgAAAAMAAABAwABAAAAAAsAAAEBAwABAAAAQAin/wIBAwAEAAAAAngAAAAMBaw

PARTI MANCANTI.....

/5GEP66o828bf8ijf/wDXi39Kd+y//wAecn/X2/8A6CKKK9xf8i+fqfofEXxx
9F+Z3Px1/wCST33/AF1X/wBBrlpv+SdeBv8AsHXv/olaKK1yj/dl6nx9P+CclP8A8jHaf9fM
f86KKK9er8RC2P/Z

-- _NEXPART_
Content-Type: application/zip; name="menu.zip" Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=" menu.zip"

UEsDBBQAAAAIAFuxPToyYWQbwAAAJ4AAAAIAAAAbWVudS50eHRVzcENw0AIRNH7StsDLiUF
pIixTSIkDAhI8b1lfczla/TmyfalOR7WEqh2OoUuJM/Rns2qbnK3CpwwMSpo4S7G6XNUyNsV
BKEXZ8qyynPnXhR0UXYixdYyYAY6vPvfUVyxTuaYg2n/uBEiuKV9235QSwECFAAUAAAACABb
sT06MmFmkG8AAACeAAAACAAAAAABACAAAAAABWVudS50eHRQSwUGAAAAAAEEAAQA2
AAAAAQAAAAAA

-- _NEXPART_ --

.
250 ok 1233265491 qp 67013

L'email ricevuta

Menù della festa Posta in arrivo X

☆ da **Caio** Cerca tutti i messaggi con l'etichetta Posta in arrivo nascondi dettagli 11.38
a Tizio
data 1 giugno 2009 11.38
oggetto Menù della festa

Ecco il menù previsto per la tua festa!!
Bye, Francesca.

2 allegati — [Scarica tutti gli allegati](#)



menu.jpeg
12K [Visualizza](#) [Scarica](#)



menu.zip
1K [Scarica](#)

Protocolli di Accesso alla Mail

Come può, un ricevente, leggere un messaggio? Ho volutamente sorvolato questa questione, perchè necessita di una nota di riguardo. L'SMTP è un servizio asincrono, quindi non è necessario che i 2 user agent di posta elettronica siano connessi simultaneamente per ricevere il servizio, questo è possibile poiché le mail indirizzate ad un particolare utente sono recapitate direttamente nella mailbox contenuta nel "suo" server di posta, che invece è sempre acceso e sempre connesso ad Internet, (il meccanismo è simile alla corrispondenza inviata vs. qualcuno che come indirizzo ha una casella di FERMO POSTA). Da ricordare inoltre, che l'SMTP, è un **push protocol**, per cui l'operazione che consente di prelevare la posta, ha necessità di utilizzare un **pull protocol**. I protocolli in questione sono il **POP3** (Post Office Protocol - v3), **IMAP** (Internet Mail Access Protocol) e l'**HTTP** (webmail).

POP3

Nell'RFC 1939 è definito il protocollo POP, un protocollo molto semplice che ha, per questo, delle funzionalità abbastanza limitate. Il POP3 inizia quando un client (user agent) apre una connessione TCP con il server di posta sulla porta 110, a questo punto procede a tre fasi distinte: **autorizzazione, transazione ed aggiornamento**.

Durante la prima parte l'utente fornisce il nome utente e la password per autenticarsi;

nella seconda fase l'utente legge, o meglio, scarica l'email;

nella terza fase, quella di aggiornamento, occorre quando l'utente digita QUIT e chiude la sessione POP3, a questo punto il server di posta elimina i messaggi che sono marcati per essere eliminati.

Una piccola nota riguardo al fatto che un utente può lasciare i messaggi sul server, per leggerli su più computer, per questo è possibile scegliere tra l'opzione di cancellare i messaggi o di lasciare i messaggi sul server.

IMAP

Come il POP3, anche l'IMAP è un protocollo che permette di leggere le email, ma a differenza di esso, permette un'organizzazione in cartelle per l'archiviazione, permettendo così di sincronizzare la posta scaricata con quella sul server, avendola a disposizione anche quando non siamo al nostro computer. Infatti se il nostro server supporta l'IMAP, è possibile loggarsi e gestire (creare, rinominare, cancellare) le cartelle ed i messaggi all'interno delle cartelle. Inoltre l'IMAP permette di cercare un messaggio nelle proprie cartelle e, cosa più importante, di gestire gli oggetti dell'email, è infatti possibile (per risparmiare la banda) visualizzare solo l'header, o solo un oggetto (un allegato) di un determinato messaggio, senza doverlo "scaricare" tutto.

HTTP (WebMail)

L'HTTP viene usato soprattutto perchè permette di leggere la posta attraverso un comunissimo browser web; in questo modo si ovvia a molti problemi, innanzitutto è difficilissimo prendere i famosi virus per computer, secondo è possibile leggere i messaggi senza doverli scaricare completamente (pensate ad un amico che vi invia 10 foto e a voi interessa solamente il testo della mail), terzo, in questo modo è possibile sia leggere che scrivere nuovi messaggi; non a caso è possibile fare ciò grazie ai linguaggi di scripting server side (come ASP, PHP o JSP) perchè implementano i protocolli di invio (SMTP) e di ricezione (POP o IMAP), prediligendo per la ricezione il protocollo IMAP che permette un'organizzazione in cartelle per archiviare le email (Hotmail docet).

Telnet e i protocolli di posta elettronica

Dopo aver chiarito i meccanismi dei protocolli di ricezione di posta elettronica e in particolare del POP3, vediamo adesso come riuscire ad accedere alla propria mailbox via Telnet.

Connettersi ad un server POP3 usando telnet

Allora per stabilire la connessione TCP con il nostro server basta mandare in esecuzione il client Telnet prescelto inserendo il nome e il numero della porta del servizio desiderato.

Digitando telnet hostname 110

dove hostname è il nome del vostro server POP3 nel mio caso è pop.mail.yahoo.com

Dopo un periodo più o meno breve il server selezionato vi risponderà (di solito con un messaggio di benvenuto) comunicandovi che è pronto a ricevere i vostri comandi. Le risposte del server POP3 sono normalmente di due tipi: positive (“+OK”) e negative (“-ERR”).

Esempio:

```
+OK POP3 server ready
```

Fase dopo fase: il significato dei comandi previsti dal protocollo.

Questa è una tipica sessione x una comunicazione POP3

(Ogni area colorata indica una delle diverse fasi del Protocollo):

```
telnet MailServer 110
S: +OK POP3 server ready
C: user pluto
S: +OK
C: pass topolino
S: +OK user successfully logged on
C: list
S: 1 384
S: 2 739
S: .
C: retr 1
S: (blah blah ...
S: ... ..
S: ... blah blah)
C: dele 1
C: retr 2
S: (blah blah ...
S: ... ..
S: ... blah blah)
C: quit
S: +OK POP3 server signing off
```

Prima Fase - L'autorizzazione

Il server è nella cosiddetta fase di autorizzazione, si aspetta cioè che voi vi facciate riconoscere al fine di poter soddisfare le vostre richieste. Dovete quindi fornire nell'ordine il nome col quale accedete al vostro server POP3 e la password. I comandi per farlo sono rispettivamente USER e PASS

Seconda fase - La transazione (elenco comandi)

Tralasciando le operazioni realmente svolte dal server a questo punto, possiamo dire di essere nella cosiddetta fase di transazione: il server numera tutte le mail presenti nella vostra mailbox, prende nota delle dimensioni delle stesse e si mette in attesa dei vostri comandi.

STAT

E' il comando che chiede informazioni sulla mailbox. Il server risponderà fornendo il numero di mail presenti e la loro dimensione totale.

Esempio:

```
.....
S: +OK
U: STAT
S: +OK 5 1572
```

LIST [n]

E' il comando che chiede l'elenco delle mail presenti. Se viene specificato un numero (n) vengono fornite informazioni su una determinata mail. Se indicate un numero non corretto o non avete messaggi vi sarà opportunamente segnalato l'errore.

Esempio A:

```
.....  
S: +OK  
U: LIST  
S: +OK 5 messages (1572 octets)  
S: 1 320  
S: 2 210  
.....  
S: 5 155  
S: .
```

Esempio B:

```
.....  
S: +OK  
U: LIST 2  
S: +OK 2 210  
S: .
```

Esempio C:

```
.....  
S: +OK  
U: LIST 7  
S: -ERR no such message
```

RETR n

E' il comando che vi permette di leggere la mail numero n. Dopo il solito +OK il server vi mostrerà l'intera mail.

Anche in questo caso se il numero della mail non è corretto vi verrà segnalato l'errore.

Esempio A:

```
.....  
S: +OK  
U: RETR 2  
S: +OK 210 octets  
S: <-----segue la mail #2----->
```

Esempio B:

```
.....  
S: +OK  
U: RETR 7  
S: -ERR no such message
```

TOP m n

Questo comando vi permette di leggere le prime n righe della mail numero m. Se n è superiore al numero di righe della mail questa viene visualizzata nella sua interezza; se invece m è il numero di una mail inesistente verrà come sempre segnalato l'errore.

Esempio:

```
.....  
S: +OK
```


Telnet e i protocolli di posta elettronica

```
U: TOP 1 7
S: +OK
S: <-----seguono le 7 righe richieste della mail #1----->
S: .
U: TOP 7 1
S: -ERR no such message
```

DELE n

E' il comando che vi permette di cancellare la mail numero n. La mail viene "marcata" per essere successivamente cancellata (quindi siete sempre in grado, prima di chiudere la sessione, di recuperarla). Dal momento in cui la marcate per essere cancellata non potete più riferirvi ad essa, a meno prima di ripristinarla (vedi il successivo comando), e quindi ogni riferimento determinerà un errore. Stessa cosa se indicate un numero non corretto.

Esempio:

```
.....
S: +OK
U: DELE 7
S: -ERR no such message
U: DELE 2
S: +OK message 2 deleted
U: DELE 2
S: -ERR message 2 already deleted (oppure no such message)
```

RSET

Come già detto è il comando che vi permette di ripristinare le mail marcate per essere cancellate che tornano disponibili. Il server risponderà con i soliti monosillabi (+OK e -ERR).

Terza fase – Aggiornamento

QUIT

Ovviamente è il comando che termina la sessione e l'unico che aggiorna la vostra mailbox. In altre parole le mail segnate per essere cancellate verranno effettivamente eliminate solo dopo l'esecuzione di questo comando. Se la transazione si conclude in altro modo (ad esempio per un blackout o per la chiusura accidentale del client) nessuna mail presente verrà modificata.

Telnet e i protocolli di posta elettronica

Una sessione di ricezione

```
+OK hello from popgate 2.43 on pop101.plus.mail.re2.yahoo.com
user indirizzoAperdere@ymail.com
+OK password required.
pass PASSWORD
-ERR [AUTH] invalid user/password user indirizzoAperdere@ymail.com
+OK password required.
pass PASSWORD
+OK
STAT
+OK 5 1572
LIST
+OK 5 messages (1572 octets)
1 320
2 210
3 240
4 310
5 155
.
RETR 2
+OK 210 octets
<-----segue la mail #2----->
.
DELE 2
+OK message 2 deleted
RETR 1
+OK 320 octets
<-----segue la mail #1----->
.
DELE 2
ERR no such message
DELE 1
+OK message 1 deleted
TOP 3 7
+OK
<-----seguono le 7 righe richieste della mail #3----->
.
TOP 2 7
-ERR no such message
QUIT
+OK popgate 2.43 on pop101.plus.mail.re2.yahoo.com signing off
```