

Università di Roma Tor Vergata
Corso di Laurea triennale in Informatica
Sistemi operativi e reti
A.A. 2016-17

Pietro Frasca

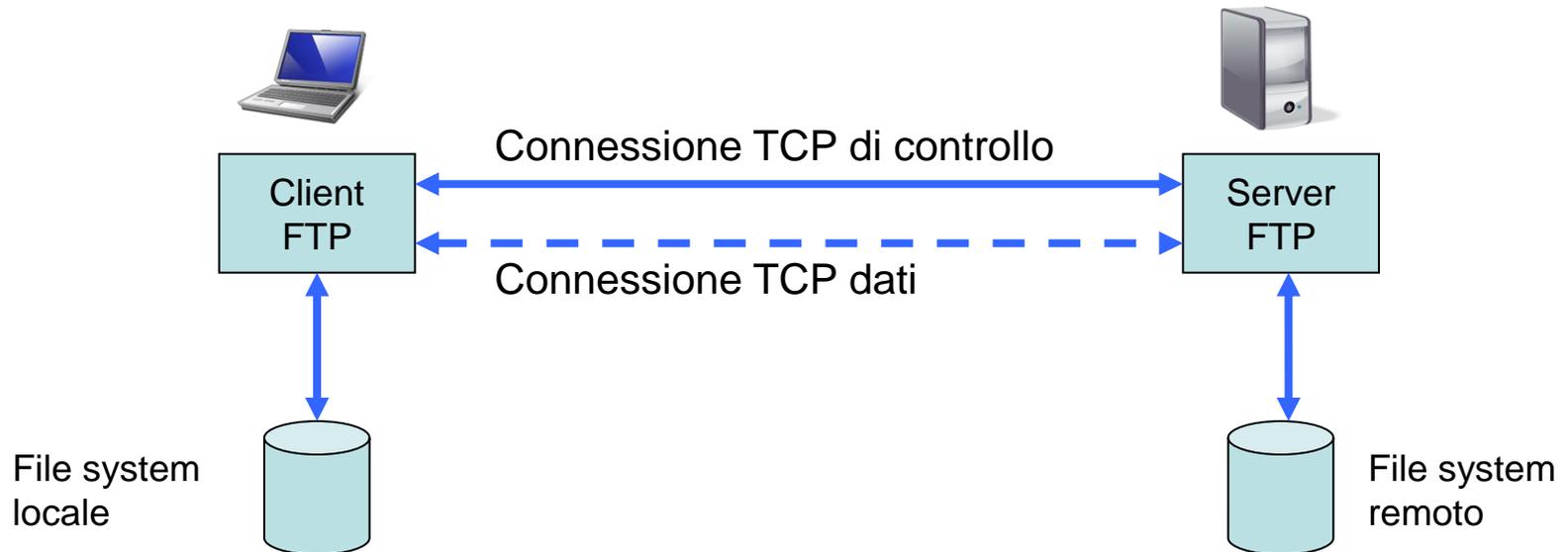
Parte II: Reti di calcolatori

Lezione 7 (31)

Martedì 28-03-2017

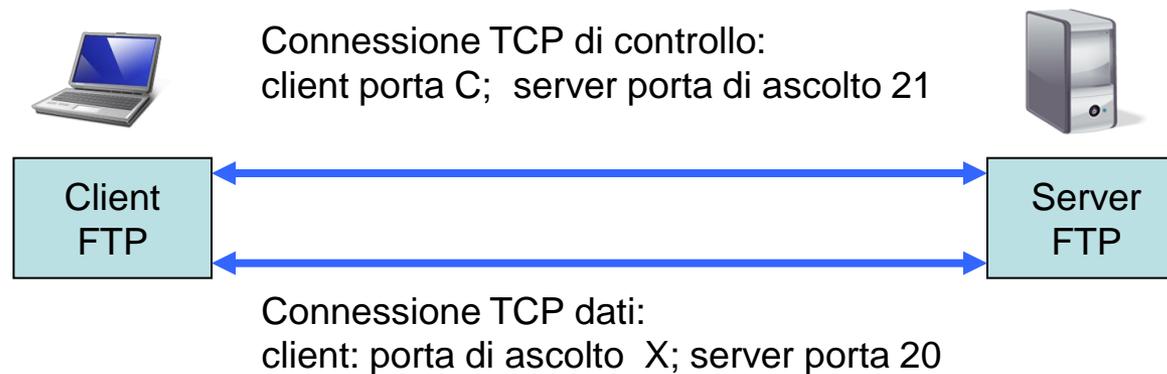
Il protocollo FTP

- Il protocollo FTP (**F**ile **T**ransfer **P**rotocol, *protocollo di trasferimento file*), consente di trasferire file da un host (locale) ad un altro host (remoto) e viceversa.
- FTP si è evoluto negli anni. La prima specifica (RFC 114), sviluppata presso il **MIT** (Massachusetts Institute of Technology), è del 1971. L'attuale specifica (RFC 959) è del 1985.

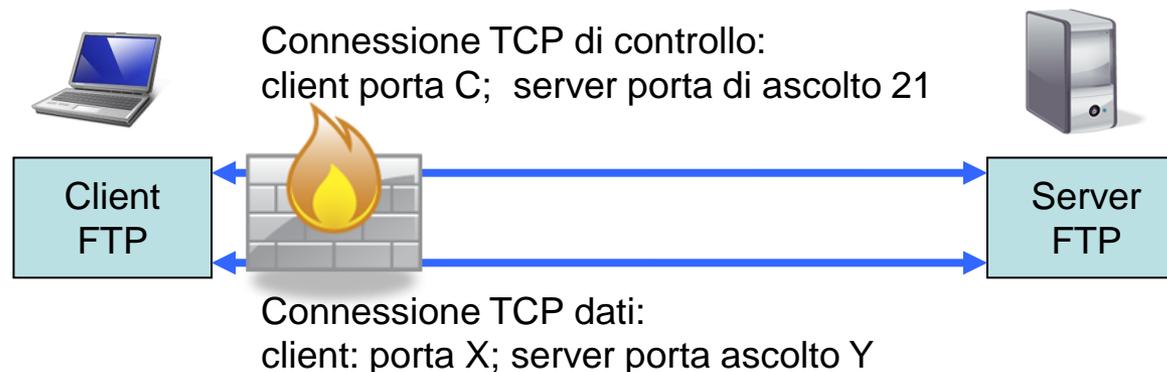


- Anche l'FTP, come l'HTTP, utilizza il TCP. L'FTP usa due connessioni TCP in parallelo: una **connessione di controllo** *per inviare i comandi* e una **connessione dati** per il trasferimento dei file.
- Il client prima instaura una connessione TCP di controllo con il server sulla **porta 21**. Il numero di porta di connessione del client è scelto automaticamente dal TCP che assegna il primo numero non usato superiore a 1024.
- Il client invia l'identificazione dell'utente e la password.
- Quando il client invia un comando per un trasferimento di file (sia da client verso server che da server verso client) al server, viene creata una connessione dati TCP che sarà usata per il trasferimento di file. L'FTP invia un file sulla connessione dati e poi la chiude. Se, durante la stessa sessione, l'utente vuole trasferire un altro file, l'FTP apre un'altra connessione dati. Quindi, con FTP, la connessione di controllo rimane aperta per tutta la durata della sessione, ma viene stabilita una nuova connessione dati per ciascun file trasferito all'interno della sessione cioè, **la connessione dati non è persistente**.

- FTP può essere eseguito in **modalità attiva o passiva**, che determina il modo in cui viene stabilita la connessione dati.
- In **modalità attiva**, il client attraverso la connessione di controllo, invia al server mediante il comando PORT il numero di porta, ad esempio **X** su cui esso ascolterà e il server avvia la connessione dati.



- In situazioni in cui il client è protetto da un firewall e non può accettare le connessioni TCP in ingresso, può essere utilizzata la **modalità passiva**. In questa modalità, il client invia il comando PASV al server e riceve da questo, mediante il comando PORT, un numero di **porta Y** di ascolto che il client utilizza per aprire la connessione dati al server.



- Il trasferimento dei dati può avvenire in quattro modalità, in base al formato dei dati:
 - **Modalità ASCII:** utilizzato per il testo. I dati se necessario vengono convertiti al formato ASCII a 8-bit prima della trasmissione e (ancora una volta, se necessario) nella rappresentazione di caratteri dell'host ricevente. Di conseguenza, questa modalità non è adatta per i file che contengono dati diversi da testo normale.
 - **Modalità immagine** (detta anche modalità **binaria**): l'host mittente invia il file byte per byte e il destinatario salva il flusso di byte appena lo riceve.
 - **Modalità EBCDIC:** utilizzata per il testo normale tra gli host che utilizzano il set di caratteri EBCDIC. Questa modalità è altrimenti come la modalità ASCII.
 - **Modalità locale:** permette a due computer con configurazioni identiche di inviare dati in un formato proprietario senza la necessità di convertirlo in ASCII

- Durante una sessione, l'FTP deve mantenere lo stato della sessione dell'utente. In particolare, deve tenere traccia delle directory, locale e remota, mentre l'utente naviga all'interno del file system locale e remoto. Dovendo tenere traccia di queste informazioni di stato e mantenere aperta sempre la connessione di controllo, i SO limitano il numero totale di sessioni che l'FTP può aprire simultaneamente.

Comandi FTP

- Il client e il server si scambiano messaggi di richiesta e risposta, attraverso la connessione di controllo (porta 21), in formato ASCII a sette bit. Quindi, come per i messaggi HTTP, i comandi FTP sono leggibili.
- Ogni messaggio di richiesta (comando) ha un formato costituito da un massimo di quattro caratteri maiuscoli ASCII.
- Alcuni dei comandi più usati sono mostrati nella tabella seguente:

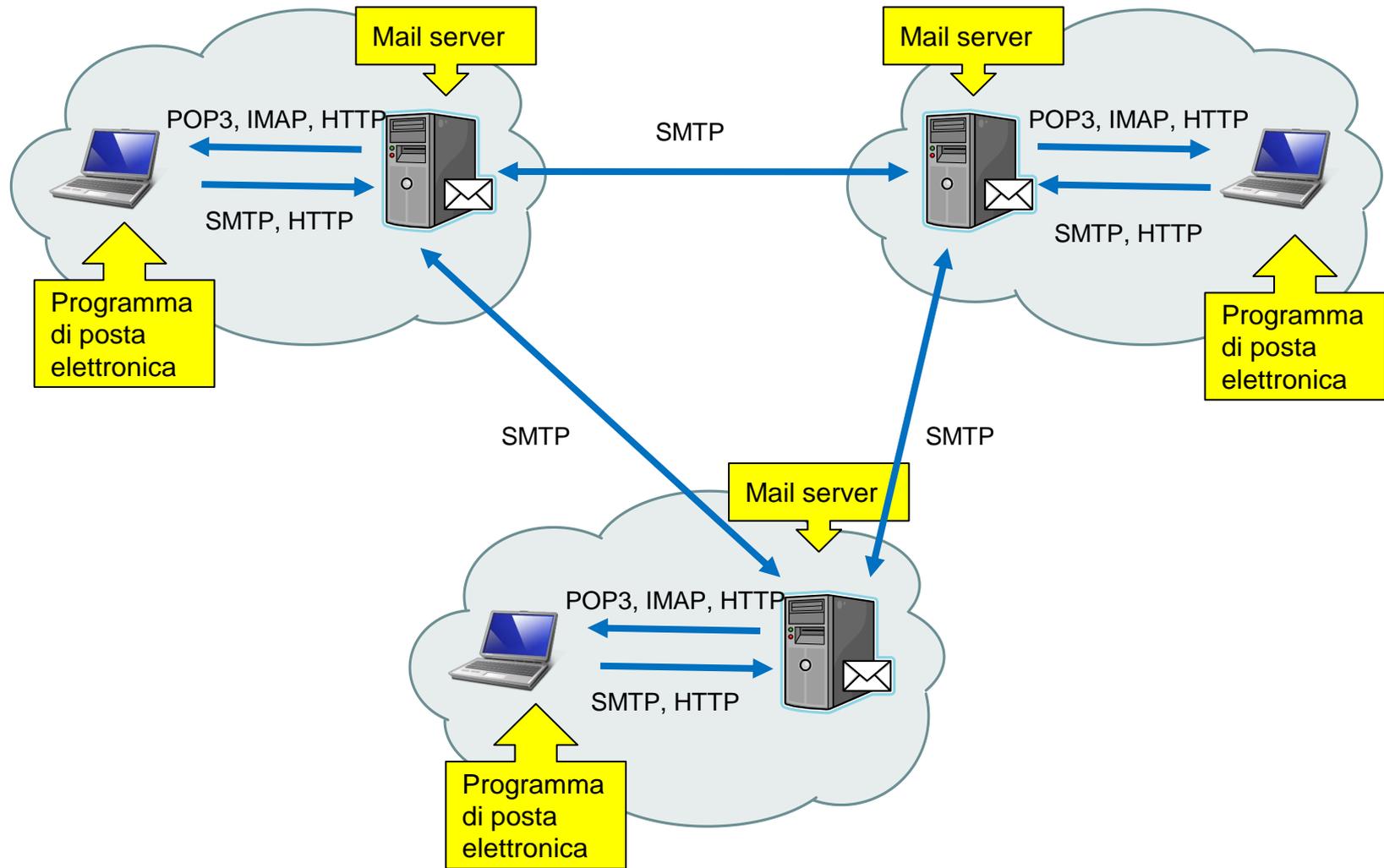
Comando	descrizione
CWD path	Cambia directory corrente
DELE path	Cancella il file specificato in path.
HELP	Ritorna la lista dei comandi accettati dal server. Con argomento fornisce spiegazioni riguardo il comando specificato.
LIST [path]	Visualizza un elenco di tutti i file contenuti nella directory remota. L'elenco dei file viene spedito su una nuova e non persistente connessione dati TCP.
PASS password	Invia la password al server
PASV	Inizializza una connessione passiva.
PWD	Visualizza la directory corrente remota.
QUIT	Chiude la sessione.
RETR path	trasferisce (<i>get</i>) un file dalla directory remota corrente alla directory corrente locale.
STOR path	trasferire (<i>put</i>) un file nella directory corrente dell'host remoto.
USER username	Invia l'identificativo dell'utente al server.

Posta elettronica

- La posta elettronica è stata una delle prime applicazioni di Internet. Col tempo è diventata sempre più sofisticata ed è una delle applicazioni di rete più utilizzate.
- La posta elettronica consente la comunicazione in modo asincrono: un'utente invia e riceve messaggi senza doversi sincronizzare con gli orari di altri utenti.
- Le attuali applicazioni di posta elettronica consentono di inviare messaggi di testo con allegati multimediali di qualsiasi tipo di file come musica, immagini, video, etc.
- Consente l'uso di **mailing list**, con le quali è possibile inviare un messaggio a più destinatari contemporaneamente.
- Attualmente sono stati realizzati numerosissimi programmi di posta elettronica. Ci sono anche applicazioni di pubblico dominio sia con interfaccia grafica che alfanumerica, come mail (unix) e pine.

- Il funzionamento della posta elettronica si basa su quattro componenti principali:
 - **i programmi di posta elettronica (*mail user agent*)**
 - **i server di posta elettronica (*mail server*)**
 - **l'SMTP (*Simple Mail Transfer Protocol*).**
 - **Protocolli per l'accesso alla posta (come *POP3, IMAP e HTTP*)**
- Quando un mittente termina di scrivere il suo messaggio, il programma di posta invia il messaggio al suo mail server, dove viene inserito nella coda dei messaggi in uscita. Quando il destinatario vuole leggere il messaggio, il suo programma recupera il messaggio dalla sua casella di posta nel suo server di posta.
- Un mail server è in grado di gestire molte utenze di posta dette mailbox (caselle di posta). Per ciascuna mailbox il mail server può memorizzare moltissimi messaggi.

La figura mostra uno schema generale della posta elettronica.



- Un messaggio inviato dal mittente, arriva al server di posta del mittente e quindi arriva al server di posta del destinatario, dove viene posto nella casella di posta.
- Quando un destinatario vuole accedere ai suoi messaggi, il server di posta esegue la **procedura di autenticazione** richiedendo username e password.
- Il server di posta gestiscono anche eventuali problemi che si verificano nel consegnare i messaggi al server del destinatario. Se il server mittente non può inviare la posta al server destinatario, allora mantiene il messaggio in una coda (**message queue**) e proverà a rinviarlo in seguito. L'intervallo di tempo tra tentativi successivi è configurabile e generalmente è di 30 minuti. Se il messaggio non viene spedito entro qualche giorno, il server rimuove il messaggio e notifica l'eccezione al mittente con un messaggio e-mail.

Indirizzo di posta elettronica

Un indirizzo di posta elettronica (email) è formato da due parti: la prima parte indica il nome dell'utente e la seconda il nome di dominio di appartenenza dell'utente. Le due parti sono separate dal carattere @. Ad esempio verdi@uniroma2.it è un indirizzo valido di posta.

SMTP

- L'SMTP, definito nella RFC 821, è il protocollo con architettura client/server usato per trasferire i messaggi di posta dal programma del mittente (client) al server del mittente e dal server mittente al server destinatario. Esso usa il TCP e la **porta 25**.
- In un mail server sono implementati sia il lato client che il lato server. Quando un server di posta invia messaggi, si comporta come client mentre quando riceve messaggi, si comporta come server.
- Attualmente l'SMTP è da considerarsi una tecnologia superata per via di alcune sue caratteristiche obsolete. In particolare, i messaggi di posta hanno **formato ASCII a sette bit**. Questo formato andava bene nei primi anni '80 quando la velocità di trasmissione della rete era molto bassa e si inviavano prevalentemente messaggi di solo testo per e-mail senza allegati di grandi dimensioni.
- Con l'SMTP, per inviare dati binari multimediali (o anche dati ascii a 8 bit) è necessario che siano **codificati in ASCII a 7 bit prima della spedizione**. Quando i messaggi arrivano al destinatario è necessario decodificarli nel formato binario originale. L'HTTP invece non richiede che i dati multimediali siano codificati in ASCII prima del trasferimento.

Esempio di dialogo tra il client SMTP e il server SMTP

- Con telnet è possibile attuare un dialogo diretto con un server SMTP:

```
telnet reti.uniroma2.it 25
```

```
220 reti.uniroma2.it ESMTP Sendmail 8.12.6/8.11.6;  
Mon, 26 Mar 2012 16:30:54 +0100 (CET)
```

```
MAIL FROM: frasca@mat.uniroma2.it
```

```
250 2.1.0 frasca@mat.uniroma2.it... Sender ok
```

```
RCPT TO: verdi@mat.uniroma2.it
```

```
250 2.1.5 frasca@mat.uniroma2.it... Recipient ok
```

```
DATA
```

```
354 Enter mail, end with "." on a line by itself
```

```
Ciao da Pietro
```

```
.
```

```
250 2.0.0 k0QGTswQ1109805 Message accepted for delivery
```

help

214-2.0.0 This is sendmail version 8.12.6

214-2.0.0 Topics:

214-2.0.0 HELO EHLO MAIL RCPT DATA

214-2.0.0 RSET NOOP QUIT HELP VERFY

214-2.0.0 EXPN VERB ETRN DSN AUTH

214-2.0.0 STARTTLS

214-2.0.0 For more info use "HELP <topic>".

214-2.0.0 To report bugs in the implementation send email to
214-2.0.0 sendmail-bugs@sendmail.org.

214-2.0.0 For local information send email to Postmaster at
your site.

214 2.0.0 End of HELP info

QUIT

221 2.0.0 reti.uniroma2.it closing connection

Connessione all'host perduta.

- Il server dà una risposta ad ogni comando. Ciascuna risposta è formata da un codice e da alcune descrizioni (facoltative).
- L'SMTP usa **connessioni TCP persistenti**: se il server di posta mittente ha vari messaggi da inviare allo stesso server di posta destinatario, esso può inviare tutti i messaggi sulla stessa connessione TCP.
- Ogni messaggio che un client vuole inviare inizia con il comando **MAIL FROM: indirizzo_mittente**, e termina con un punto su una riga. Il comando **QUIT** consente la spedizione di tutti i messaggi scritti.

Formati dei messaggi di posta e MIME

- un messaggio e-mail è costituito da una serie di linee di intestazione (**parte intestazione**) e dal corpo del messaggio. Le linee di intestazione sono definite nella RFC 822.
- Le linee di intestazione e il corpo del messaggio sono separate da una linea vuota (cioè da CR+LF).
- Come per l'HTTP, ciascuna linea è formata da una parola chiave seguita da due punti e da un valore, ad esempio

From: pf@uniroma2.it.

- Alcune delle parole chiave sono obbligatorie, altre facoltative.
- Le linee di intestazione **From:** e la linea **To:** sono obbligatorie.
- L'intestazione **Subject:** è una delle intestazioni facoltative, più usate (specifica l'oggetto del messaggio).

- Una tipica intestazione di un messaggio è il seguente:

From: frasca@uniroma2.it

To: student@uniroma2.it

Subject: lezione 7 di reti

- Dopo l'intestazione del messaggio, segue una linea bianca, poi il corpo del messaggio (in ASCII). Il messaggio si chiude con una linea contenente solo un punto, come visto prima.

L'estensione MIME per dati non-ASCII

- Per spedire un messaggio con contenuto diverso dal testo ASCII , è necessario inserire nel messaggio intestazioni aggiuntive.
- Queste intestazioni aggiuntive sono definite nelle RFC 2045 e RFC 2046, e nell'estensione **MIME** (***Multipurpose Internet Mail Extensions***, estensione multi-uso della posta in Internet) della RFC 822.
- Due intestazioni fondamentali per supportare i file multimediali sono l'intestazione **content-Type:** e la **Content-Transfer-Encoding:.**
- L'intestazione **Content-Type:** specifica il formato del contenuto del messaggio e permette all'applicazione ricevente di compiere le adeguate operazioni sul messaggio. Per esempio, specificando che il messaggio contiene un'immagine GIF, l'applicazione ricevente può richiamare la funzione di decompressione dell'immagine GIF e quindi visualizzarla.

- L'intestazione **Content-Transfer-Encoding**: indica l'algoritmo di codifica usato per convertire il corpo del messaggio di formato diverso dal formato ASCII a 7 bit, al formato ASCII a 7 bit.
- Quindi, quando un'applicazione di posta elettronica riceve un messaggio con le due suddette intestazioni:
 - prima usa il valore dell'intestazione **ContentTransfer-Encoding**: per convertire il corpo del messaggio dal formato ASCII a 7 bit al suo tipo originale;
 - poi usa l'intestazione **Content-Type**: per richiamare il giusto modulo per l'interpretazione del corpo del messaggio ed eventuale conseguente visualizzazione.

Esempio di messaggio con linee di intestazione MIME

From: pf@uniroma2.it
To: verdi@casa.it
Subject: mappa dell'isola del tesoro
Content-Transfer-Encoding: base64
Content-Type: image/jpeg

(dati codificati base-64 ----

----- dati codificati base-64)

- Dal messaggio MIME dell'esempio vediamo che l'immagine JPEG è stata codificata con il codificatore **base64**, che è una dei vari algoritmi di codifica standardizzate nel MIME [RFC 2045] per la conversione nel formato ASCII a 7 bit. Un'altra tecnica di codifica è la **quoted-printable**, che è usata per convertire un messaggio in formato ASCII a 8 bit in formato ASCII a 7 bit.

- L'intestazione **Content-Type**: ha il formato seguente:

Content-Type: type/subtype[; parametri]

- dove i "parametri" sono facoltativi.
- Il campo Content-Type è usato per specificare la natura dei dati nel corpo di un'entità MIME, assegnando nomi type (tipo) e subtype (sottotipo) ai media.
- Dopo i nomi tipo e sottotipo, il resto del campo di intestazione consiste di un set di parametri. In generale, il tipo è usato per dichiarare il tipo generico di dati, mentre il sottotipo specifica il formato particolare per quel tipo di dati. I parametri sono modificatori del sottotipo e come tali non influiscono significativamente sulla natura dei contenuti.
- Il MIME è stato progettato per essere ampliabile, e ci si aspetta che il set di coppie tipo/sottotipo di media e i parametri a essi associati aumentino nel tempo.

- Attualmente sono definiti quasi 600 tipi MIME (2006). Per ogni tipo esiste un elenco di sottotipi associati. Tre di questi tipi sono:
- **text**: indica che il corpo del messaggio contiene informazioni in forma di testo. Un tipo molto diffuso di coppia tipo/sottotipo è **text/plain**. Il sottotipo plain indica testo normale senza comandi di formattazione o di stile. Un'altra coppia tipo/sottotipo è **text/html**. Il sottotipo html indica che il messaggio ha formato HTML.
- **image**: il tipo image (immagine) indica che il corpo del messaggio è un'immagine. Due diffuse coppie tipo/sottotipo sono image/gif e image/jpeg.
- **application**: è spesso usato per dati che devono essere elaborati da un'applicazione prima di poter essere visibili o utilizzabili. Per esempio, quando un utente allega un documento Word a un messaggio e-mail, si usa **application/msword** per la coppia tipo/sottotipo. Quando l'agente dell'utente destinatario osserva il contenuto del tipo application/msword, apre l'applicazione Microsoft Word e passa il corpo del messaggio MIME all'applicazione.

- Come per una pagina Web, anche un messaggio di posta può contenere più oggetti di diverso formato.
- Ricordiamo che l'HTTP invia ciascuno degli oggetti all'interno di **messaggi di risposta HTTP indipendenti** i quali possono essere trasportati su una singola connessione o su più connessioni, a seconda se si usino connessioni persistenti o non persistenti.
- La posta elettronica **inserisce tutti gli oggetti (detti "parti") nello stesso messaggio.** Per specificare che un messaggio di posta contiene più parti, si usa la linea di intestazione **Content-type: multipart/mixed.**
- Quando sono presenti più parti nello stesso messaggio è necessario indicare:
 - **l'inizio e la fine di ogni parte;**
 - **il formato di ciascuna parte;**
 - **L'algoritmo di codifica usato per ciascuna parte di formato non-ASCII ;**

La definizione dei limiti di una parte si ottiene usando **caratteri di delimitazione (boundary character)** tra ciascuna parte, mentre per specificare il formato e la codifica di ciascuna parte nel messaggio si usano, come già visto, le linee di intestazione **Content-type:** e **Content-Transfer-Encoding:.**

Esempio di messaggio multipart/mixed

From: pf@mat.uniroma2.it
To: mario@casa.it
Subject: Quadri di Van Gogh
content-Type: multipart/mixed; Boundary=StartOfNextPart

--StartOfNextPart

ti invio in allegato le foto dei quadri che ...

--StartOfNextPart

Content-Transfer-Encoding: base64

Content-Type: image/jpeg

dati *codificati* base-64

.

. *dati codificati* base-64

--StartOfNextPart

Content-Transfer-Encoding: base64

Content-Type: image/jpeg

dati *codificati* base-64

.

. *dati codificati* base-64

Il messaggio ricevuto

Il server destinatario, dopo aver ricevuto un messaggio con linee di intestazione MIME, aggiunge una linea di intestazione **Received**: all'inizio del messaggio. Questa intestazione specifica il nome del server SMTP che ha spedito il messaggio ("from"), il nome del server SMTP che ha ricevuto il messaggio ("by") e l'ora in cui il server ricevente ha avuto il messaggio. Quindi, il messaggio visto dall'utente di destinazione ha questa forma:

```
Received: from pro.mail.fr by mail.uniroma2.it;  
Mon, 26 Mar 2012 11:30:10 GMT  
From: lino@pro.mail.fr  
To: eva@uniroma2.it  
Subject: foto ricordo di Parigi.  
Content-Transfer-Encoding: base64  
Content-Type: image/jpeg  
dati codificati base-64 . . . . .  
. . . . .
```

- Un messaggio può avere **più** linee di **intestazione** **Received:**. Questo perché un messaggio può essere **inoltrato** a più di un server SMTP prima di giungere al destinatario. Per esempio, un utente può configurare la sua casella di posta in modo tale che inoltri (forward) tutti i suoi messaggi ad un altro mail server, allora i messaggi arrivati a questo utente avranno le linee:

```
Received: from mail.uniroma2.it by mail2.unimi.it  
Mon, 26 Mar 2012 11:31:21 GMT
```

```
Received: from pro.mail.fr by mail.uniroma2.it;  
Mon, 26 Mar 2012 11:30:10 GMT
```

- Queste linee di intestazione forniscono al destinatario del messaggio l'elenco dei server SMTP utilizzati con la data in cui l'inoltro è stato eseguito.