



UNIVERSITÀ  
DI PARMA

DIPARTIMENTO DI SCIENZE MATEMATICHE, FISICHE ED INFORMATICHE  
Corso di Laurea in Informatica

# Il Livello Applicativo – Parte B

La posta elettronica

RETI DI CALCOLATORI - a.a. 2023/2024

Roberto Alfieri

# Livello Applicativo: sommario

## PARTE A

- ▶ Applicativi UDP: TFTP e DNS

## PARTE B

- ▶ I servizi di posta elettronica: SMTP, POP e IMAP.

## PARTE C

- ▶ Il World Wide Web

## PARTE D

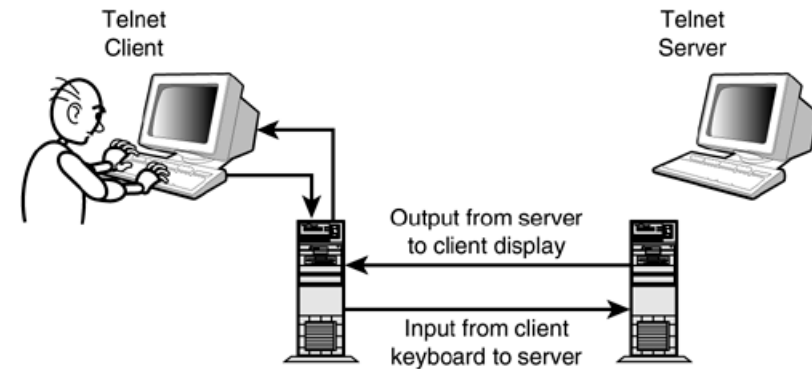
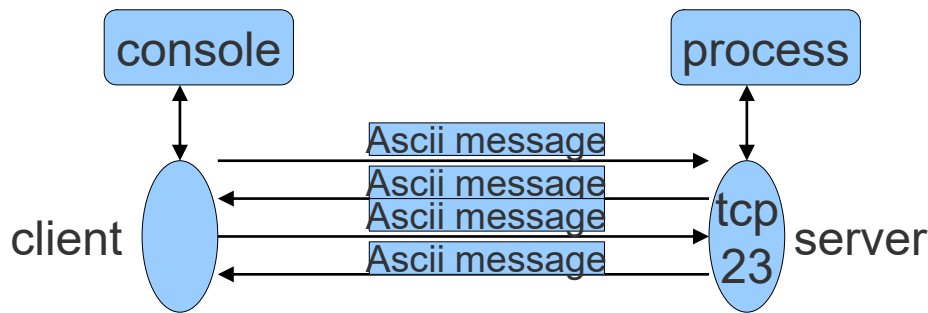
- ▶ Multimedia

# Il protocollo Telnet: emulazione di terminale remoto

Il telnet è un protocollo storico di TCP/IP (RFC 854) creato per l'accesso remoto alla console testuale di un host, basandosi su una connessione TCP tra client e server.

La porta assegnata da IANA è la 23/TCP. Operazioni:

- **Client.** Il client legge lo stream dallo standard input (tastiera) e lo gira al server. Lo stream proveniente dal server viene inviato allo standard output (video).
- **Server.** Il server legge le linee provenienti dal client, le interpreta come comandi di console e invia al client l'output del comando. Il dialogo inizia con la richiesta delle credenziali (username/password).



Telnet è un protocollo testuale (richieste e risposte sono spedite in ASCII, compresa la password) ha quindi un livello di sicurezza molto basso ed è stato sostituito da protocolli più sicuri (SSH).

Viene comunque ancora utilizzato per casi speciali (console di apparati di rete) o come semplice strumento di analisi di altri protocolli testuali (SMTP, POP3, IMAP, HTTP, ..)

# La Posta Elettronica

Definita nel 1982 con gli RFC821, RFC822 ed estesa con RFC2821 e RFC2822  
Ogni utente possiede una MailBox in cui gli altri utenti possono liberamente inserire messaggi.

Nel 2015 lo spam rappresentava il 55% del flusso mondiale di mail  
(<https://securelist.com/kaspersky-security-bulletin-spam-and-phishing-in-2015/73591/> )

Un indirizzo di posta è costituito da due identificativi: nome del server che gestisce la mailbox e identificativo dell'utente: `<utente>@<server>`

Dove `<server>` è l'indirizzo IP o un suo identificativo nel DNS.

Il record MX del DNS consente di creare caselle di posta in un dominio, a cui possiamo associare uno o più server di posta per la sua gestione.

Ad esempio il seguente record DNS (ottenuto con `dig -t MX unipr.it`):  
`unipr.it. 86400 IN MX 0 unipr-it.mail.protection.outlook.com.`

Consente di creare indirizzi di posta del tipo `<utente>@unipr.it`  
Le cui caselle verranno gestite dal server `unipr-it.mail.protection.outlook.com.`

# La Posta Elettronica : l'architettura

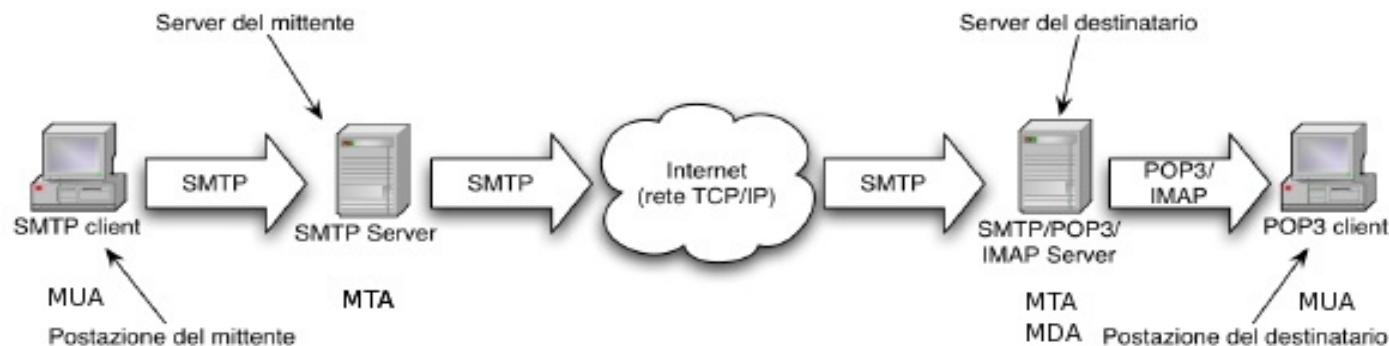
Ogni utente ha bisogno di uno MUA - “Message User Agent”, un programma che gli consente di inviare messaggi o leggere i messaggi dalla propria Mailbox.

Il MUA consegna il messaggio ad un MTA - “Message Transfer Agent” che ha il compito di trasportare il messaggio a destinazione. Il messaggio può attraversare diversi MTA prima di arrivare nella MailBox del destinatario.

Il primo MTA è detto anche Submission server perché è utilizzato dai MUA per la sottomissione delle mail e dovrebbe supportare autenticazione (SMTP AUTH).

Sull'ultimo MTA è presente anche l'MDA - “Message Delivery Agent” che si occupa della consegna del messaggio nella MailBox dell'utente.

Se il MUA del destinatario e MDA sono su host diversi occorre un protocollo per la lettura dei messaggi (POP3 o IMAP)



# Il formato dei messaggi

Il formato dei messaggi è stato codificato nel 1982 attraverso l'RFC 822, superato nel 2001 dall'RFC 2822 e poi nel 2008 dall'RFC 5322.

I messaggi sono codificati in ASCII standard (7bit) in righe di max.1000 caratteri

Formato:	<b>Intestazione</b>	(Header)
	<b>&lt;cr&gt;&lt;lf&gt;</b>	(riga vuota)
	<b>corpo del messaggio</b>	(Body)

**Intestazione:** in ogni riga la coppia <campo>:<valore>

Principali campi:

- ▶ To: indirizzi dei destinatari
- ▶ Cc: indirizzi destinatari secondari
- ▶ Bcc: destinatari secondari con indirizzi nascosti
- ▶ From: La persona che ha creato il messaggio (obbligatorio)
- ▶ Received: Riga aggiunta da ogni MTA attraversato
- ▶ Date: data e ora di invio del messaggio
- ▶ Subject: Breve riepilogo del messaggio
- ▶ Message-ID: Identificatore unico del messaggio (generato automaticamente).
- ▶ User-Agent: Client di posta utilizzato dall'utente

# Esempio messaggio

MTA  
Attraversati



Received: from smtp.unipr.it (sud.cce.unipr.it [160.78.48.162])  
by unipr.it (8.12.6/8.12.6) with ESMTP id j4CCXorj032665  
for <destinatario@unipr.it>; Thu, 12 May 2005 14:33:50 +0200

Received: from smtp2.mathworks.com (smtp2.mathworks.com [144.212.95.218])  
by smtp.unipr.it (8.12.10/8.12.10/SuSE Linux 0.7) with ESMTP id j4CCTeOW001983  
for <destinatario@unipr.it>; Thu, 12 May 2005 14:29:45 +0200

Received: from mail-vif.mathworks.com (fred-ce0.mathworks.com [144.212.95.18])  
by smtp2.mathworks.com (8.12.11/8.12.11) with ESMTP id j4CCVPR5009740  
for <destinatario@unipr.it>; Thu, 12 May 2005 08:31:25 -0400 (EDT)

Received: from telesto.mathworks.com (telesto.mathworks.com [144.212.95.234])  
by mail-vif.mathworks.com (8.11.7/8.11.7) with ESMTP id j4CCVOH06340  
for <destinatario@unipr.it>; Thu, 12 May 2005 08:31:25 -0400 (EDT)

Message-ID: <74127410.1115901084851.JavaMail.wwwadmin@telesto.mathworks.com>  
Date: Thu, 12 May 2005 08:31:24 -0400 (EDT) #Nota: EDT US East Coast  
From: [sender@mathworks.com](mailto:sender@mathworks.com)  
User-Agent: Mozilla Thunderbird 1.0.2 (Windows/20050317)  
Reply-To: [sender@mathworks.com](mailto:sender@mathworks.com)  
To: destinatario@unipr.it  
Subject: Greetings

Greetings from Mathworks  
bye

# MIME

Il formato RFC822 del 1982 era stato pensato per messaggi in cui corpo era esclusivamente testo espressi in ASCII standard.

Questo schema non ammette lettere accentate, alfabeti non latini, messaggi multimediali ecc.

La soluzione è stata proposta da MIME RFC1341, oggi ampiamente utilizzata in Internet.

MIME (Multipurpose Internet Mail Extensions) introduce 5 nuove intestazioni:

1) **MIME-version**

2) **Content-description**

3) **Content-id**

4) **Content-transfer-encoding**. Il nome dello schema di codifica utilizzato per trasformare il messaggio in ASCII standard. Principali valori:

**Ascii 7bit**: nessuna codifica. Linee fino a 1000 caratteri

**Ascii 8bit**: viola la versione originale del protocollo, ma probabilmente funziona.

**Quoted-printable**: messaggi ASCII non standard. I caratteri superiori al 127 sono codificati con = seguito dal codice ASCII in esadecimale. (città -> citt=9A)

**base64**: Per dati binari. Ogni sequenza di 6 bit viene trasformato in un carattere ASCII grazie ad una codifica di 64 simboli (sprecati 2 bit ogni 6, i dati codificati occupano il 35% in più)

Esempio di codifica di una immagine: `openssl base64 -e -in immagine.png -out immagine.b64`

Esempio di decodifica: `openssl base64 -d -in immagine.b64 -out immagine.png`

5) **Content-type**. Natura del corpo del messaggio

Espresso nella forma `type/subtype` (esempio `Content-Type: text/plain`)

Utile per attivare automaticamente il Viewer corretto (esempio `video/mpeg`)

Tipi e sottotipi sono definiti da IANA - <http://www.iana.org/assignments/media-types/index.html>



# Principali Content-types

Type	Subtype	Description
Text	Plain	Unformatted text
	Enriched	Text including simple formatting commands
Image	Gif	Still picture in GIF format
	Jpeg	Still picture in JPEG format
Audio	Basic	Audible sound
Video	Mpeg	Movie in MPEG format
Application	Octet-stream	An uninterpreted byte sequence
	Postscript	A printable document in PostScript
Message	Rfc822	A MIME RFC 822 message
	Partial	Message has been split for transmission
	External-body	Message itself must be fetched over the net
Multipart	Mixed	Independent parts in the specified order
	Alternative	Same message in different formats
	Parallel	Parts must be viewed simultaneously
	Digest	Each part is a complete RFC 822 message

# Esempio messaggio con MIME

Received: from ...

Message-ID: ...

Date: ...

From: ...

To: ...

Subject: ..

MIME-Version: 1.0

Content-Type: multipart/mixed; boundary="-----ThIs-RaNdOm-StRiNg-/=\_468328521:"

-----ThIs-RaNdOm-StRiNg-/=\_468328521:

Content-Transfer-Encoding: 7bit

Content-Type: text/plain

Ecco l'allegato

ciao

-----ThIs-RaNdOm-StRiNg-/=\_468328521:

Content-Transfer-Encoding: base64

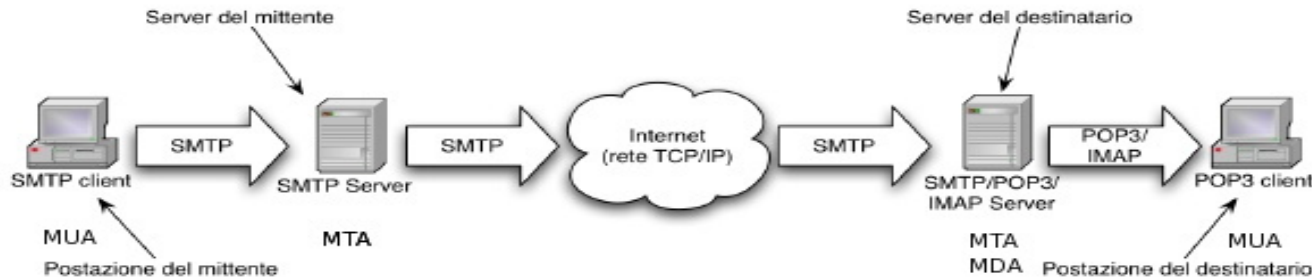
Content-Type: application/msword

BgAAAAAAAAAGYGAAAAAAAAAZgYAAAAAAAAABIAwAAPAEAAAI BAAAAAAAAASAMAAAAAAAAACAQAAAAAAAAAEgD  
AAAAAAAAA6gkAAAAAAAAAAAAAAAAAGYGAAA  
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAASAMAAAAAAAAADqCQAAAAAAAAAGYGAAABIAwAAZgYAAAAAAAAAAAA  
AAAAAK4JAAAAAAAAAgEAAAAAAAAACAQAAA

-----ThIs-RaNdOm-StRiNg-/=\_468328521:--

# SMTP

SMTP è un protocollo applicativo (25/TCP) che si occupa del trasferimento di un messaggio da MUA a MTA o da MTA a MTA. L'MTA può essere destinatario finale (MTA+MDA) o di trasferimento (Mail Relay, Mail scanner, ..). Il protocollo è codificato **ASCII Standard** ovvero prevede uno scambio di dati testuali.



Principali **comandi** del client SMTP:

- HELO: indirizzi dei destinatari
- MAIL FROM: mittente del messaggio
- RCPT TO: destinatario del messaggio
- DATA: corpo del messaggio
- QUIT: fine del messaggio
- RSET: reset
- HELP: nome del comando

Principali **risposte** del server SMTP:

- 220: Servizio pronto
- 250: Comando richiesto completato
- 251: Utente non locale, il messaggio sarà inoltrato
- 221: Chiusura canale di trasmissione
- 421: Servizio non disponibile
- 500: Errore di sintassi
- 501: Errore di sintassi nei parametri
- 554: Transazione fallita

# Formato della Mailbox

Esistono 2 formati principali:

## Mbox

I messaggi ricevuti sono accodati in un singolo file per ogni utente

Su Unix si trova in `/var/spool/mail/nomeutente`

Ogni messaggio inizia con una linea

“From sender@domain..”

## Maildir

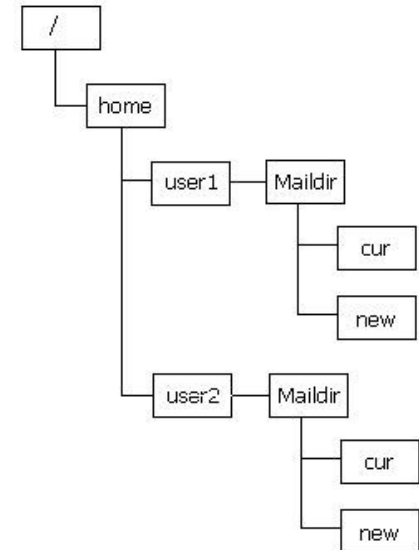
Si utilizza una directory per ogni utente

All'interno della directory viene creato un file (di testo) per ogni messaggio ricevuto.

From [sender1@domain1](mailto:sender1@domain1)  
Messaggio1

From [sender2@domain2](mailto:sender2@domain2)  
Messaggio2

...



# MUA : Mail User Agent

Un MUA è una applicazione che viene usata dall'utente per inviare e ricevere e-mail.

## **comando «mail»**

MUA di base dei sistemi Linux. Non gestisce allegati MIME, POP e IMAP.

E' utile per l'invio automatico di messaggi all'interno di processi.

L'invio è affidato all'MTA su localhost. Esempio di utilizzo:

```
mail -s test user@domain<< EOF
prova di spedizione
EOF
```

## **comando «alpine»**

MUA testuale, ma con gestione dello schermo. Software libero per Linux e MacOS.

Gestisce gli allegati MIME, può gestire la Mailbox da remoto con il protocollo IMAP

## **«Mozilla Thunderbird»**

E' un MUA open software con interfaccia grafica disponibile per Windows, Linux e MacOS. Gestisce POP3/ IMAP e un filtro bayesiano anti spam.

# Esempio di dialogo SMTP con telnet

telnet localhost 25 ...

## Client

HELO client.com  
MAIL FROM: <src@com>  
RCPT TO: <dest@com>  
DATA

From: tizio  
To: caio  
Subject: prova

testo  
testo

MESSAGGIO

.  
QUIT

## Server

220 srv.com SMTP service ready  
250 client.com OK  
250 sender src@com OK  
250 Recipient dest@com OK  
  
250 Message Accepted  
221 client.com closing connection

Un MTA è un server che accetta nel campo RCPT TO indirizzi diversi dall'host locale.

# ESMTP (Extended SMTP)

Visto che con il passare degli anni vengono richieste sempre nuove funzionalità ad SMTP, con l'RFC 1869 del 1995 è stata definita una generale struttura standard di SMTP, denominata ESMTP, in grado di gestire le estensioni presenti e future.

Per utilizzare ESMTP occorre presentarsi con EHLO (anziché HELO). Se EHLO è accettato il server risponde con la lista delle estensioni supportate. Esempio:

```
> EHLO client.com
```

```
250-8BITMIME
```

```
(8 bit data transmission)
```

```
250-SIZE
```

```
(Message Size Declaration)
```

```
250-DSN
```

```
(Delivery Status Notification)
```

```
250-AUTH
```

```
(SMTP autenticato)
```

```
250-STARTTLS
```

```
(comunicazione cifrata con StartTLS)
```

```
.....
```

Le estensioni più interessanti sono AUTH e STARTTLS, che introducono autenticazione e cifratura dei dati in fase di sottomissione del messaggio.

# Sicurezza dell'SMTP

Poiché il protocollo SMTP non prevede autenticazione chiunque può contattare un MTA per

- spedire mail verso chiunque (**spam**)
  - Ingannare la vittima per indurlo a rivelare informazioni sensibili o svolgere azioni dannose (**phishing**)
  - furto di identità, ovvero impersonare l'identità di altri (**spoofing**)
  - Sfruttare gli allegati della mail per diffondere malware (virus, trojan, worm, ecc)
- Inoltre, poichè manca anche la riservatezza le mail sono esposte **ad intercettazione**.

Alcune semplici tecniche per mitigare:

- Filtro indirizzi IP sul server SMTP per accettare esclusivamente messaggi in cui il mittente o il destinatario sono locali.
- porte 25 bloccate dal firewall perimetrale per forzare l'utilizzo di MTA istituzionali (ad esempio dotati di antivirus e antispam).
- Verifica della registrazione DNS diretta e inversa del client
- Utilizzo del Resource Record SPF ([Sender Policy Framework](#)) sul DNS per identificare quali sono gli indirizzi IP abilitati a spedire per il nostro dominio.



## Sottomissione dei Messaggi con SMTP-AUTH

[Extended SMTP](#) è una estensione (RFC 1869) che include nuove funzionalità, tra cui l'autenticazione con SMTP-AUTH per la fase di sottomissione di e-mail attraverso l'introduzione di un MTA dedicato denominato MSA (Message Submission Agent)

Tramite l'estensione SMTP AUTH un MSA può:

- richiedere le credenziali (user/pass o certificati) del client
- cifrare le comunicazioni tra MUA e MSA
- utilizzare una porta di ascolto diversa: 587/tcp

La cifratura SMTP-AUTH riguarda solo la consegna del messaggio all'MSA. Se si vuole una cifratura end-to-end occorre utilizzare una estensione di MIME denominata Secure MIME (S/MIME).

# SMTP server

Il server SMTP ha il compito di ricevere e gestire messaggi di posta elettronica. Opportunamente configurato può svolgere la funzione di

- MSA se riceve messaggi dal MUA offrendo servizi di autenticazione e riservatezza
- MTA se invia il messaggio ricevuto verso altri server SMTP
- MDA se gestisce caselle di posta in cui deposita i messaggi ricevuti

Esistono diverse distribuzioni open source di ESMTP server; le più diffuse sono Sendmail e Postfix.

# Lettura dei Messaggi

La consegna del messaggio della mailbox utente è gestita dall'MDA (Message Delivery Agent) che ha anche il compito di consegnare il messaggio al MUA dell'utente previo opportuno meccanismo di autenticazione.

Quando è nato il protocollo SMTP gli utenti lavoravano sulla stessa macchina dove risiedevano le Mailbox, che quindi il MUA poteva accedere direttamente.

Con l'avvento dei PC il MUA si è disaccoppiato dall'MDA ed è nata la necessità di nuovo protocollo di rete per la comunicazione MUA-MDA. Due possibili protocolli: POP3 o IMAP.



# POP3

**POP3** (Post Office Protocol v.3, RFC 1939) è un protocollo ASCII con autenticazione per il trasferimento dei messaggi dal MailBox allo User Agent utilizzando un servizio TCP sulla porta 110.

Dopo la connessione TCP il protocollo attraversa 3 fasi:

Autenticazione: invio delle credenziali (USER e PASS)

Transazioni: Esecuzione dei comandi (LIST, RETR, DELE, QUIT)

Aggiornamento: Dopo il QUIT il server cancella effettivamente i messaggi eliminati e interrompe la connessione

POP3 è utilizzato tipicamente da Home Users, connessi via modem o ADSL all'ISP, per trasferire (RETR) tutti i nuovi messaggi, che vengono poi eventualmente cancellati dal server (DELE). Il MailBox server funziona così da area di transito per i messaggi, che vengono gestiti sull'Hard Disk dell'utente.

Riferimenti: [http://openskill.info/release/guida\\_ai\\_protocolli\\_internet/i\\_protocolli\\_pop3\\_e\\_imap.htm](http://openskill.info/release/guida_ai_protocolli_internet/i_protocolli_pop3_e_imap.htm)

Poiché tutte le comunicazioni, incluse le credenziali di autenticazione, avvengono in chiaro, a POP3 è stato affiancato il **protocollo sicuro POP3S** in cui la comunicazione è cifrata grazie all'utilizzo del layer SSL/TLS. POP3S utilizza una porta diversa, la 995/TCP.

# Esempio di dialogo POP3 con telnet

telnet localhost 110 ...

Client	Server
	+OK POP3 ready
USER nomeutente	+OK
PASS password	+OK
STAT	+OK 2 1000      #(2 mess – 1000 bytes)
LIST	+OK
	1 500
	2 500
	.
TOP 1	# mostra intestazione messaggio 1
TOP 2	# mostra intestazione messaggio 2
RETR 2	# recupera messaggio 2
	.....
	.....
	.
DELE 2	+OK    Marked to be deleted
QUIT	+OK    Logging Out, message deleted

# IMAP

IMAP (Internet Message Access Protocol, RFC 2060), è un protocollo alternativo a POP3 per consentire all'user agent la gestione dei messaggi ricevuti, utilizzando il servizio TCP sulla porta 143.

A differenza di POP3 presume che i messaggi debbano rimanere sul server. Per questo fornisce la possibilità di gestire cartelle di posta sul server in cui archiviare i messaggi ricevuti.

IMAP è adatto per utenti che accedono alla posta utilizzando diversi MUA (casa, lavoro, portatile,...).

*Riferimenti:* [http://openskill.info/release/guida\\_ai\\_protocolli\\_internet/i\\_protocolli\\_pop3\\_e\\_imap.htm](http://openskill.info/release/guida_ai_protocolli_internet/i_protocolli_pop3_e_imap.htm)

Poiché tutte le comunicazioni, incluse le credenziali di autenticazione, avvengono in chiaro, a IMAP è stato affiancato il **protocollo sicuro IMAPS** in cui la comunicazione è cifrata grazie all'utilizzo del layer SSL/TLS. IMAPS utilizza una porta diversa, la 993/TCP.

# Esempio di dialogo IMAP con telnet

telnet localhost 143 ...

## Client

a login <username> <password>

a list "" "\*" \*

a examine inbox

a logout

## Server

OK Dovecot ready

OK Logged in

LIST "/" saved-messages

LIST "/" sent-mail-feb-2018

LIST "/" sent-mail

LIST "/" INBOX

OK List completed.

OK [PERMANENTFLAGS ()] Read-only mailbox.

6 EXISTS

0 RECENT

OK [UNSEEN 2] First unseen.

OK [UIDVALIDITY 1520015846] UIDs valid

OK [UIDNEXT 7] Predicted next UID

OK [READ-ONLY] Examine completed (0.004 secs).

BYE Logging out

OK Logout completed.

Connection closed by foreign host.