

# SMTP

Obiettivo: invio affidabile ed efficienti di messaggi a un host che potrebbe non essere online, tramite server intermediari che disaccoppiano mittente e destinatario.

Caratteristiche:

- protocollo push;
- richiede un servizio di trasporto affidabile che mantiene l'ordine dei byte (TCP 25);
- identifica gli host tramite nomi di dominio, quindi ha bisogno di un DNS;
- basato su deleghe di responsabilità: il client trasferisce il messaggio (facendo più tentativi su un certo intervallo di tempo — *spooling*) o comunica l'insuccesso.

## Server mail

Mantiene:

- delle *mailbox* per gli utenti di cui è responsabile, contenenti la loro posta ricevuta;
- una coda (*spool*) di messaggi in uscita, da inoltrare ad altri host.

Il server può processare gli indirizzi dei destinatari, per esempio espandendo un alias (uno-molti o molti-uno).

## Protocollo

**fasi** handshake, traserimento messaggio, chiusura connessione;

**comandi e dati** testo ASCII;

**risposte** codice di stato e descrizione.

## Handshake

- il client apre la connessione TCP;
- il server invia lo stato 220 `READY FOR MAIL` e un identificatore;
- il client risponde con il comando `HELO` (o `EHLO` per indicare il supporto di estensioni) e un identificatore;

## Trasferimento

Comandi:

- `MAIL FROM:<reverse-path>`
- `RCPT TO:<forward-path>`
- `DATA crlf <message> crlf crlf`

Il server risponde ad ogni comando con un codice di stato (e.g. 250 `OK`).

## Messaggio

Composto da intestazioni (To, From, Subject, ... potenzialmente diversi da quelli specificati con i comandi SMTP) e corpo.

I messaggi sono testo ASCII, ma tramite l'estensione MIME estende si possono usare altri formati:

- intestazioni `MIME-Version`, `Content-Transfer-Encoding` (e.g. `base64`, `quoted-printable` per messaggi con pochi caratteri non ASCII), `Content-Type`;
- contenuto `multipart/mixed`; `boundary=<...>` permette di usare codifiche diverse per parti diverse del messaggio; dopo ogni occorrenza del `boundary` si indicano le intestazioni di codifica.