

## UNITÀ

# 8

# L'APPLICATION LAYER DEL TCP/IP



Guarda  
la presentazione  
dell'unità

### IN QUESTA UNITÀ

- 1 UNA VISIONE D'INSIEME DELLA RETE INTERNET
- 2 IL LIVELLO APPLICATION E I SUOI PROTOCOLLI
- 3 TELNET: IL PROTOCOLLO PER L'EMULAZIONE DI TERMINALE
- 4 FTP: IL PROTOCOLLO PER IL TRASFERIMENTO DI FILE
- 5 HTTP: IL PROTOCOLLO PER LE APPLICAZIONI WEB
- 6 SMTP, POP E IMAP: I PROTOCOLLI PER LA POSTA ELETTRONICA
- 7 I PROTOCOLLI PER LE APPLICAZIONI MULTIMEDIALI
- 8 VoIP: LA TECNOLOGIA PER LA VOCE
- 9 **LABORATORIO** PACKET TRACER: SERVER SMTP E POP3
- 10 **LABORATORIO** PACKET TRACER: SERVER FTP
- LABORATORIO ONLINE TELNET E LA POSTA ELETTRONICA
- LABORATORIO ONLINE WIRESHARK: ANALISI DI HTTP, SMTP, POP3

#### conoscenze

Organizzare il software di comunicazione in livelli.  
Conoscere le principali applicazioni utilizzate nelle reti TCP/IP e i relativi protocolli.  
Conoscere i principali protocolli per le applicazioni multimediali.

#### abilità

Saper usare i numeri di porta opportuni per le comunicazioni Client-Server tra applicativi.  
Configurare il software di rete sugli host.  
Riconoscere le vulnerabilità dei protocolli di livello Application.

#### competenze

Conoscere il funzionamento dei principali protocolli di livello Application.  
Saper scegliere il tipo di protocollo in base all'applicazione che si vuol utilizzare.  
Configurare, installare e gestire sistemi di elaborazione dati e reti.



### FLIPPED CLASSROOM

#### A casa

- Leggi la Lezione 2 di questa Unità;
- esegui una ricerca sulle applicazioni attualmente più utilizzate su Internet che usano il paradigma Peer-to-Peer per svolgere attività di: condivisione di file, comunicazione (Instant Messaging), distribuzione di contenuti (Content Delivery Network);
- trasferisci la tua analisi in una tabella o mappa concettuale in cui elenchi

le applicazioni che hai trovato, descrivendone caratteristiche, vantaggi e svantaggi.

#### In classe

- Confrontate i risultati descritti nelle tabelle o mappe realizzate;
- discutete i motivi che spiegano le eventuali differenze, al fine di comprendere meglio il funzionamento dell'instradamento nelle reti.

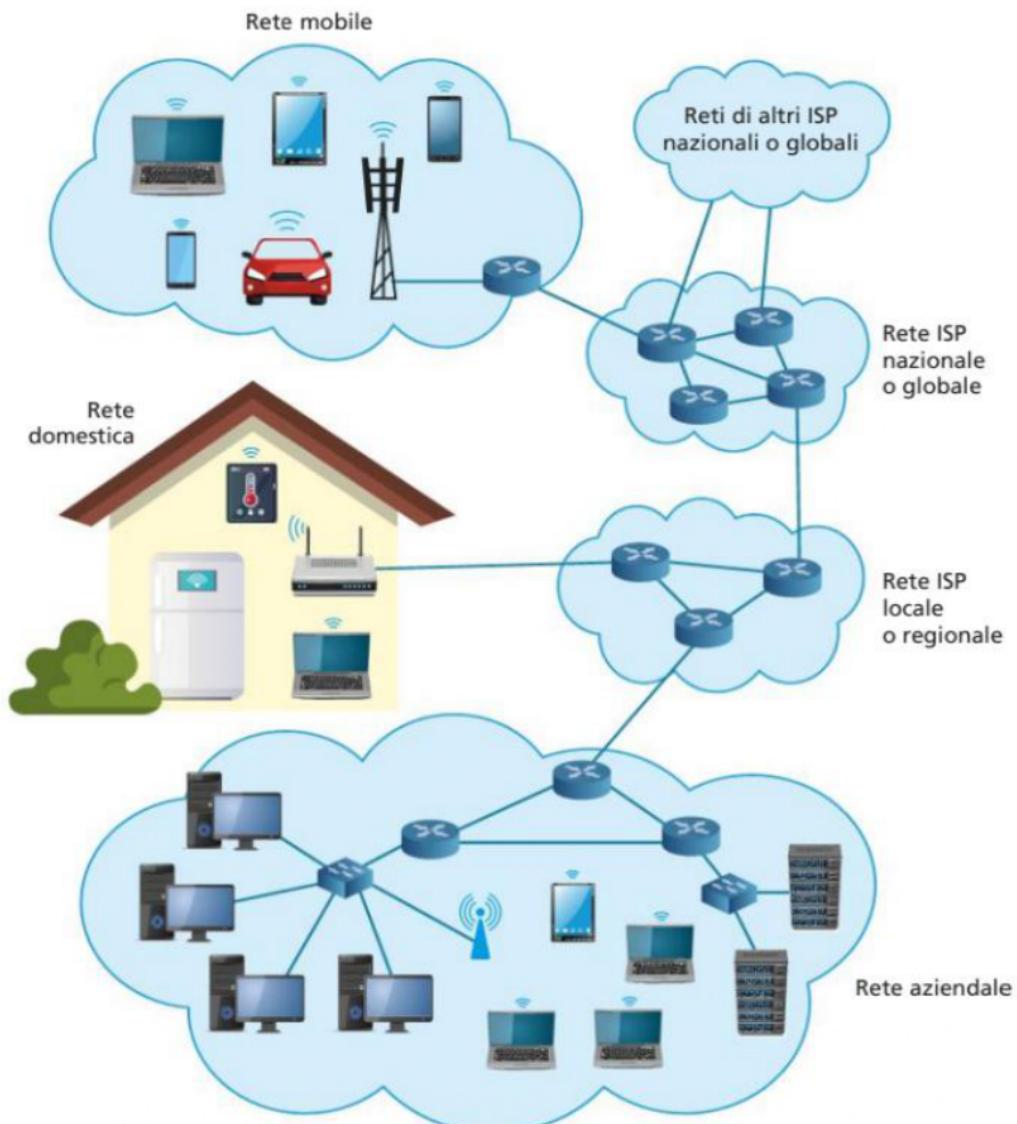
# 1 UNA VISIONE D'INSIEME DELLA RETE INTERNET

## 1.1 L'interconnessione delle reti

Nelle precedenti Lezioni e nel volume per il terzo anno, abbiamo descritto le varie componenti di Internet, la rete che interconnette miliardi di dispositivi in tutto il mondo. Più volte abbiamo ripreso e approfondito le varie parti, partendo dall'hardware, gli apparati di rete e le modalità di collegamento, per risalire lo stack TCP/IP fino ad arrivare alle applicazioni software, che offrono i servizi di rete agli utenti finali e sono descritte in questa Unità. Il viaggio nel mondo di Internet però non finisce qui, nel volume per il quinto anno riprenderemo nuovamente alcuni elementi per approfondire ulteriori caratteristiche e funzionalità, soprattutto dal punto di vista della sicurezza e della gestione. Inoltre, si descriveranno le modalità di accesso a Internet da rete mobile, con i nuovi protocolli e standard.

La **FIGURA 1** raffigura le varie componenti di Internet, evidenziando la suddivisione che abbiamo utilizzato più volte tra end system e intermediate system.

**FIGURA 1** Le reti interconnesse con Internet



In passato gli end system erano soprattutto computer, ancora al giorno d'oggi definiamo Internet come una "computer network", ma l'evoluzione della tecnologia e il numero sempre più elevato di connessioni ha ampliato la tipologia di end system, includendo dispositivi di varia natura. Nelle reti domestiche, oltre a PC, laptop, tablet e smartphone, si connettono a Internet TV, console per il gioco, ma anche elettrodomestici e termostati che possono così essere gestiti da remoto. Analogamente la rete aziendale connette tra loro dispositivi wired e wireless come PC, smartphone, tablet e stampanti multifunzione, dai quali, sulla base delle politiche aziendali, si accede a Internet.

Nella Figura 1 si mostra la connessione delle reti periferiche, dove sono collocati gli end system, con le reti degli Internet Service Provider, formate dagli intermediate system, che inoltrano i pacchetti dati verso la destinazione, come visto nell'Unità 5.

## 1.2 I protocolli per la comunicazione su Internet

Nell'Unità 1 abbiamo descritto i modelli usati per organizzare la comunicazione in rete, spiegando come il modello ISO/OSI sia ormai da considerare un modello di riferimento, mentre TCP/IP è l'architettura a livelli implementata su Internet. IETF (Internet Engineering Task Force) è l'ente internazionale di standardizzazione che si occupa delle specifiche dei protocolli di Internet. I documenti pubblicati da IETF sono chiamati RFC (Request for Comments), spesso in questo volume abbiamo riportato l'abstract degli RFC per i protocolli più importanti. Nell'Unità 2 si è presentato un altro importante ente di standardizzazione, IEEE (Institute of Electrical and Electronics Engineers), che gestisce il progetto 802 pubblicando standard per reti PAN, LAN e MAN.

Nelle successive Unità 3, 4, 5 e 6 sono stati descritti i protocolli di comunicazione e gli standard utilizzati nei livelli Network e Transport.

Con l'Unità 7 siamo ancora saliti nello stack TCP/IP, prendendo in esame due servizi e protocolli di livello Application: il DHCP e il DNS. Si collocano al livello più alto in quanto sono applicazioni Client-Server, che offrono servizi fondamentali per la comunicazione su Internet, strettamente legati ai protocolli del livello inferiore.

In questa Unità si prendono in esame i protocolli che permettono agli utenti di un'applicazione di comunicare. Per esempio il protocollo HTTP è utilizzato per la comunicazione tra le applicazioni web client (i browser sul dispositivo dell'utente) e le applicazioni web server (sui server del provider), realizzando così il servizio noto come WWW (World Wide Web). Le applicazioni che usano Internet sono applicazioni distribuite, che coinvolgono molti sistemi che scambiano dati tra loro. Per questo motivo il software sviluppato usufruisce dei servizi offerti dal livello Transport tramite le socket, le interfacce di rete viste nell'Unità 6.

A livello più basso Internet è una rete formata da hardware e software che permette di interconnettere due dispositivi che necessitano di comunicare. A livello più alto, Internet è un'infrastruttura che fornisce servizi alle applicazioni distribuite su diversi sistemi.

### FISSA LE CONOSCENZE

- Descrivi le tipologie di end system che si connettono alla rete Internet.
- Con intermediate system quali tipologie di apparati si identificano?
- Spiega che cosa significa che Internet è un'infrastruttura che fornisce servizi alle applicazioni distribuite.

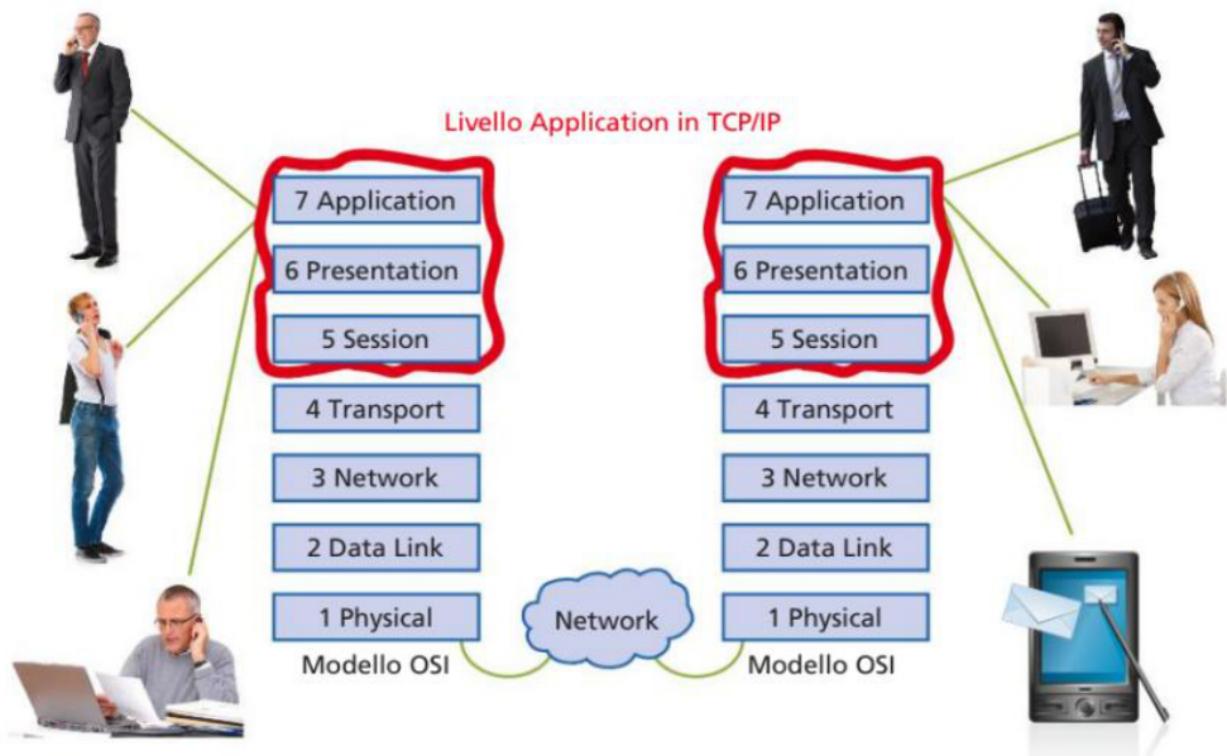
## 2 IL LIVELLO APPLICATION E I SUOI PROTOCOLLI

### 2.1 I protocolli del livello Application

Il livello Application dell'architettura TCP/IP ingloba le funzionalità svolte dai 3 livelli più alti del modello OSI: Session, Presentation e Application. Questa scelta ha permesso ai programmati di avere un buon grado di flessibilità nello sviluppo delle applicazioni.

Con l'analisi delle funzionalità tipiche del livello Application si è giunti all'origine dei dati che attraversano la rete: a questo livello si implementa l'interfaccia tra l'utente e la rete, la comunicazione viene convertita in dati che possono essere trasferiti attraverso una rete (FIGURA 2).

**FIGURA 2** Il livello Application di TCP/IP corrisponde agli ultimi 3 livelli del modello OSI



Un protocollo di livello Application definisce:

- i **tipi di messaggi** scambiati, per esempio: richiesta e risposta;
- la **sintassi** di ciascun tipo di messaggio, per esempio quanti sono i campi presenti e quanto spazio occupano;
- la **semantica** dei vari campi, qual è il contenuto informativo che trasportano;
- le **regole** che sottendono al dialogo, quando e come l'applicazione invia un messaggio o risponde a uno ricevuto.

Nelle Lezioni seguenti vedremo queste definizioni applicate ai protocolli relativi alle applicazioni più diffuse.

I protocolli del livello Application supportano la comunicazione tra i processi client e server. Nelle Lezioni seguenti esaminiamo i più diffusi: dagli storici protocolli Telnet

ed FTP all'HTTP per il web e SMTP per la posta elettronica. Un importante protocollo di livello Application comunemente usato per la gestione delle reti IP è il protocollo SNMP (Simple Network Management Protocol) che sarà analizzato nel volume per il quinto anno dove si affronta la tematica della gestione della rete.

Nell'Unità 5 del volume del terzo anno, avevamo descritto i due modelli Client-Server e Peer-to-Peer applicati alle reti, li ritroviamo nei protocolli del livello Application:

- **Client-Server (C/S):** è l'architettura software tra le più diffuse; fin dalle origini di Internet, infatti, la utilizzano applicazioni come il WWW, la posta elettronica e il file transfer; ogni servizio applicativo ha una componente client e una server:
  - il server è sempre attivo in attesa di ricevere le richieste dai molti client, ha un indirizzo IP assegnato staticamente e una porta TCP o UDP, di tipo Well Known nel caso delle applicazioni più diffuse;
  - un client si connette solo nel momento in cui deve comunicare con il server, sovente ha un indirizzo IP assegnato dinamicamente; da notare che i client non comunicano direttamente tra loro;
- **Peer-to-Peer (P2P):** è una comunicazione tra pari, gli utenti scambiano informazioni tra loro in modo cooperativo, mediante specifici protocolli; i peer non sono sempre connessi come i server e cambiano spesso l'indirizzo IP, quindi la gestione risulta più complessa. Le applicazioni Peer-to-Peer possono essere di file sharing, per esempio BitTorrent, di videoconferenza o telefonia su Internet come Skype e altre ancora.

In generale, per usufruire di un servizio applicativo è necessario averne l'autorizzazione, quindi gli utenti devono disporre di un account che viene loro concesso dall'amministratore del server remoto e che useranno ognqualvolta vorranno inviare delle richieste al server.

## 2.2 Applicazioni Peer-to-Peer

Nelle reti denominate Peer-to-Peer non c'è distinzione tra computer server e computer client. Infatti in questo modello ogni computer è considerato alla pari degli altri e sono i singoli utenti a decidere quali risorse del proprio computer condividere.

Quando si passa da una rete P2P alle applicazioni P2P, lo scenario cambia, infatti un'applicazione P2P permette al computer di agire sia come client sia come server all'interno di una stessa sessione di comunicazione. Ciò non è realizzabile a livello di rete P2P dove è consentito che un computer svolga sia il ruolo di client sia di server, ma su due distinte sessioni di comunicazione.

Un'applicazione Peer-to-Peer non deve utilizzare una rete Peer-to-Peer necessariamente, può anche funzionare con reti Client-Server.

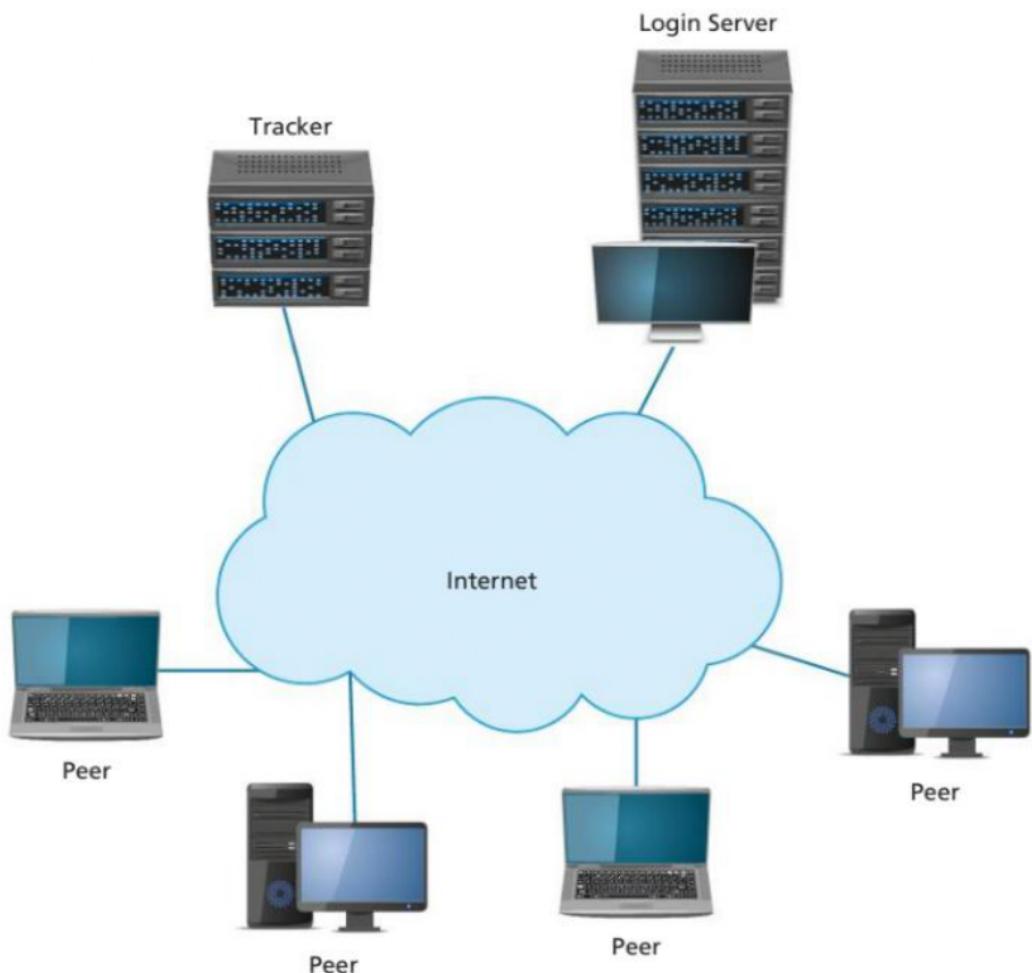
La **FIGURA 3** mostra come è strutturata una generica applicazione P2P, sono presenti:

- un portale web, denominato **Login Server**, a cui si connette un peer per verificare la disponibilità dei servizi desiderati;
- un server, denominato **Tracker**, che fornisce al nuovo utente l'elenco dei peer disponibili;
- i computer **peer**, che sono macchine anonime che si collegano al sistema quando ne hanno necessità.

### #prendinota

Una peculiarità delle architetture P2P è l'**auto-scalabilità**, per esempio: in un'applicazione di file sharing, alcuni peer scaricano un file generando un certo traffico in rete. Nel momento in cui mettono a loro volta a disposizione il file per altri peer, aumentano automaticamente la capacità del sistema.

**FIGURA 3** Architettura di una generica applicazione P2P



Un'applicazione P2P è caratterizzata da 3 aspetti fondamentali:

- 1. ricerca:** quando un nuovo peer utilizza l'applicazione P2P per prima cosa ricerca quali servizi, dati e peer sono disponibili;
- 2. locazione:** il nuovo peer necessita di alcune informazioni utili a trovare il tracker dell'applicazione, per esempio il suo indirizzo IP, e i peer che hanno i dati che desidera. Inoltre, anche il nuovo peer deve fornire al tracker informazioni utili per la sua localizzazione e sui dati che possiede e può rendere disponibili agli altri peer;
- 3. trasferimento dei dati:** le applicazioni P2P utilizzano approcci diversi per realizzare le operazioni di upload e download dei dati desiderati dal peer. I più frequenti sono il metodo **push**, in cui è il peer che carica i dati a stabilire i peer destinatari degli stessi, e il metodo **pull** in cui è il peer che vuole scaricare i dati a inviare la richiesta a un insieme di potenziali peer da cui effettuare il download.

#### FISSA LE CONOSCENZE

- Qual è la differenza tra il livello Application del modello OSI e quello dell'architettura TCP/IP?
- Quali modelli di comunicazione si possono implementare a livello Application?
- Descrivi le caratteristiche delle applicazioni Peer-to-Peer.

## 3 TELNET: IL PROTOCOLLO PER L'EMULAZIONE DI TERMINALE

### 3.1 La sessione Telnet

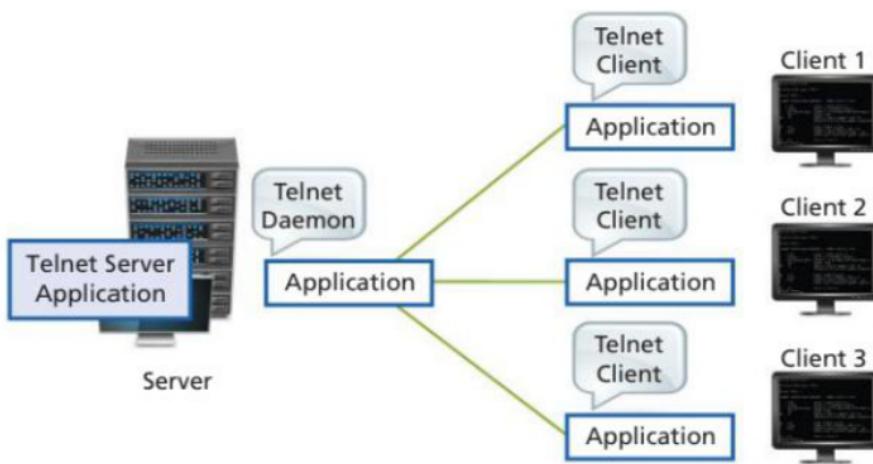
Telnet è un protocollo Client-Server: la componente client di Telnet è un'applicazione di emulazione di terminale (testuale) che permette agli utenti di un sistema di accedere ad applicazioni che si trovano su host remoti, come se fossero direttamente connessi a tali sistemi. L'host di destinazione deve avere la componente server di Telnet. Il procedimento può essere ripetuto in quanto dall'host remoto ci si può connettere a un altro host e così via.

Nella **FIGURA 4** si mostra un tipico scenario Client-Server in cui viene utilizzato il protocollo Telnet: più client Telnet richiedono la connessione a un server Telnet. Sul server è in esecuzione un servizio chiamato Telnet Daemon. Le richieste dei client devono essere gestite sul computer server contemporaneamente e in modo separato. Per far ciò il protocollo Telnet si affida alle funzionalità offerte dai livelli sottostanti.

Vediamo un semplice esempio di utilizzo di questa applicazione: un client Telnet viene avviato su un computer Windows per connettersi a un computer remoto Linux. Sul computer Windows si apre una finestra che consente all'utente di lavorare direttamente sul computer Linux. L'utente deve possedere un account che gli consenta l'accesso al computer remoto, quindi prima di poter inviare comandi ed eseguire applicazioni, l'utente deve essere autenticato.

In generale, non si usa Telnet per connettere un computer Windows con un altro computer Windows, perché Windows ha una propria modalità per connettere i suoi computer attraverso una rete: la funzionalità Connessione desktop remoto. Inoltre, dalla versione Windows Vista non è più disponibile il comando Telnet dal Prompt dei comandi, in modalità predefinita, però è possibile ripristinarlo dal Pannello di controllo.

**FIGURA 4** Tipico scenario Client-Server dell'applicazione Telnet



#### → PROBLEMA

Abilitare Telnet su Windows 10.

#### → SVOLGIMENTO

Su Windows 10 il client Telnet è disabilitato, ma può essere abilitato dal Pannello di controllo: in Programmi selezionare la voce Attiva o disattiva funzionalità di Windows e spuntare la casella relativa a Telnet.

Per attivare Telnet è poi necessario eseguire l'applicazione Prompt dei comandi come amministratore (tasto destro) e digitare:

```
C:\>dism /online /Enable-Feature /FeatureName:TelnetClient
```

**esercizio**

## 3.2 Lo standard del protocollo Telnet

Le specifiche di Telnet furono definite agli inizi degli anni Ottanta e sono contenute negli RFC 854 e RFC 855.

### IN ENGLISH PLEASE

Network Working Group

J. Postel

**Request for Comments: 854**

J. Reynolds

ISI

Obsoletes: NIC 18639

May 1983

### TELNET PROTOCOL SPECIFICATION

#### 1. INTRODUCTION

The purpose of the TELNET Protocol is to provide a fairly general, bi-directional, eight-bit byte oriented communications facility. Its primary goal is to allow a standard method of interfacing terminal devices and terminal-oriented processes to each other. It is envisioned that the protocol may also be used for terminal-terminal communication (*linking*) and process-process communication (distributed computation).

### #prendinota

Il protocollo Telnet è uno dei primi protocolli applicativi creato per la suite TCP/IP. Come altri servizi applicativi nati insieme a Internet, Telnet è ormai poco usato su reti pubbliche, nella sua forma originale, a causa della scarsa sicurezza che offre.

### #prendinota

Telnet è ancora utilizzato per il test dei servizi di rete presenti sui web server e di posta elettronica, in quanto permette di inviare, in modo semplice, i comandi e di esaminarne le risposte. Telnet è anche utilizzato per collegarsi come console ad apparati di rete, per esempio per accedere a un router remoto.

Telnet utilizza **TCP** come protocollo di trasporto e la porta **23**. I dati e i comandi sono trasmessi in formato ASCII a 8 bit.

Alcuni dei principali comandi di Telnet sono:

- **open host [port]** apre una sessione Telnet su host usando port;
- **close** chiude la sessione Telnet;
- **display** mostra i parametri relativi alla sessione;
- **send code** invia caratteri speciali al server;
- **status** visualizza lo stato attuale della sessione.

I comandi che invia il client al server devono essere preceduti da un **carattere di escape** per far sì che il server interpreti le informazioni ricevute come comandi e non come dati.

La versione originale di Telnet offre un livello minimo di sicurezza per controllare l'accesso al computer remoto, realizzato con username e password, che, però, viene a cadere dal momento che i dati viaggiano in chiaro sulla rete.

L'impiego di SSH (Secure SHell) rende il protocollo più sicuro grazie all'uso della crittografia.

Molte sono le implementazioni del protocollo Telnet che si possono scaricare gratuitamente da Internet. Un esempio, valido sia in ambiente Unix che Windows, è il software **PuTTY** che offre un client Telnet con crittografia, utilizza infatti SSH-2. Il sito web è: <http://www.chiark.greenend.org.uk/~sgtatham/putty>.

### FISSA LE CONOSCENZE

- Qual è lo scopo originario del protocollo applicativo Telnet?
- In quali altri casi può essere utilmente impiegato Telnet?
- Telnet è l'unica modalità per potersi connettere a un computer remoto?
- Descrivi alcuni comandi tipici di Telnet.

## 4 FTP: IL PROTOCOLLO PER IL TRASFERIMENTO DI FILE

### 4.1 Gli standard del protocollo FTP

Il **File Transfer Protocol (FTP)** è un protocollo per il trasferimento di file tra un computer client e un server. FTP è stato standardizzato negli anni Ottanta e le sue specifiche sono descritte nell'RFC 959.

#### IN ENGLISH PLEASE

Network Working Group

J. Postel

**Request for Comments: 959**

J. Reynolds

Obsoletes RFC: 765 (IEN 149)

ISI

October 1985

#### FILE TRANSFER PROTOCOL (FTP)

##### 1. INTRODUCTION

The objectives of FTP are 1) to promote sharing of files (computer programs and/or data), 2) to encourage indirect or implicit (via programs) use of remote computers, 3) to shield a user from variations in file storage systems among hosts, and 4) to transfer data reliably and efficiently. FTP, though usable directly by a user at a terminal, is designed mainly for use by programs.

A distanza di pochi anni dalla specifica di FTP venne definita una versione più leggera denominata **Trivial File Transfer Protocol (TFTP)**, specificata nell'RFC 1350. La novità fu la sostituzione del protocollo di trasporto TCP, utilizzato in FTP, con **UDP**, **port 69**. TFTP è ancora usato per trasferire file all'interno di una rete locale, per via della maggior sicurezza che offre la LAN (TFTP non prevede né autenticazione né cifratura) e della bassissima percentuale di pacchetti errati o persi.

#### IN ENGLISH PLEASE

Network Working Group

K. Sollins

**Request For Comments: 1350**

MIT

STD: 33

July 1992

Obsoletes: RFC 783

#### THE TFTP PROTOCOL (REVISION 2)

[...]

##### 1. Purpose

TFTP is a simple protocol to transfer files, and therefore was named the Trivial File Transfer Protocol or TFTP. It has been implemented on top of the Internet User Datagram protocol (UDP or Datagram) [2] so it may be used to move files between machines on different networks implementing UDP. (This should not exclude the possibility of implementing TFTP on top of other datagram protocols.) It is designed to be small and easy to implement. Therefore, it lacks most of the features of a regular FTP. The only thing it can do is read and write files (or email) from/to a remote server. It cannot list directories, and currently has no provisions for user authentication.

#### #prendinota

Un esempio di applicazione TFTP, gratuita, è **Solarwinds TFTP Server**, utilizzata soprattutto per lavorare sugli apparati di rete per operazioni di upload, backup o di configurazione.

Attualmente ci sono molti modi per trasferire file attraverso una rete dati (come Internet) che utilizzano tecnologie non specificatamente pensate a questo scopo, per esempio email, instant messaging, chat e web server. Tutte queste applicazioni offrono il vantaggio di un'interfaccia familiare a chi le usa quotidianamente, ma mancano della robustezza che offre un'applicazione di file transfer creata a questo scopo.

## 4.2 La connessione tra client e server FTP

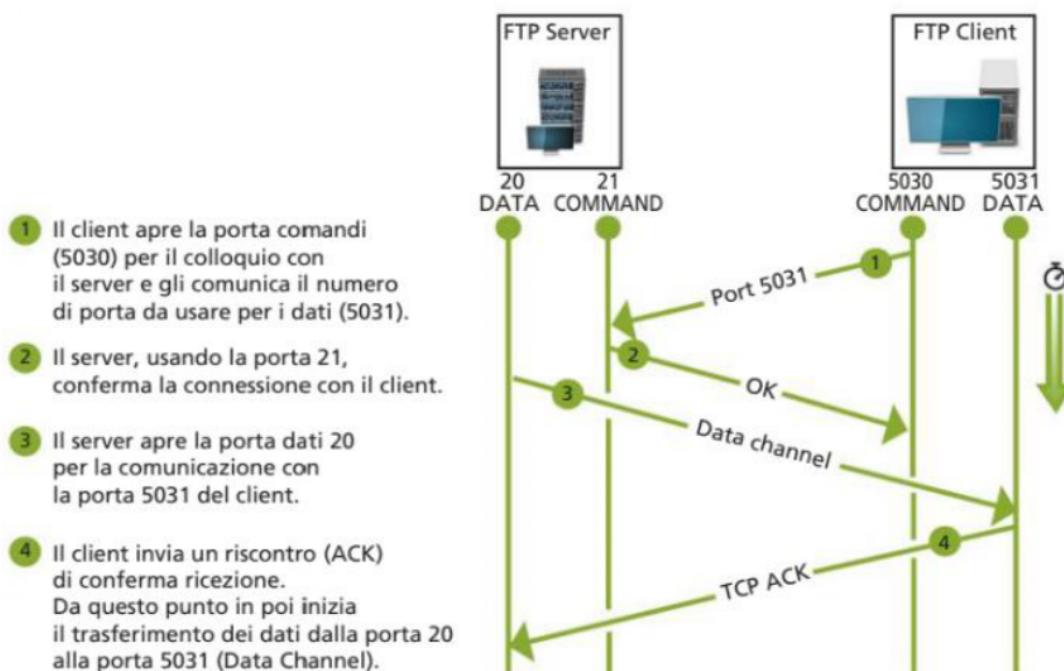
A differenza di altri protocolli, FTP utilizza **due canali** per la comunicazione tra client e server:

- un canale viene utilizzato per l'invio di **comandi**, e relative risposte, tra client e server; questo canale viene sempre aperto in direzione client → server e utilizza la **porta 21**, chiamata porta di controllo;
- l'altro canale è utilizzato per l'invio dei **dati**, viene quindi aperto in direzione server → client e utilizza la **porta 20**, chiamata porta dati.

La connessione tra client e server può avvenire secondo due modalità: FTP active mode e FTP passive mode.

### ■ FTP ACTIVE MODE

La **FIGURA 5** mostra lo scambio di messaggi tra client e server FTP nella modalità attiva: il client si connette da una porta qualsiasi **N** (con  $N > 1.023$ ) alla porta di controllo del server, la porta 21, e si mette in ascolto sulla porta dati  $N + 1$  inviando al server il comando **Port N+1**. Il server si connette alla porta dati specificata dal client ( $N + 1$ ) utilizzando la propria porta dati, la porta 20.

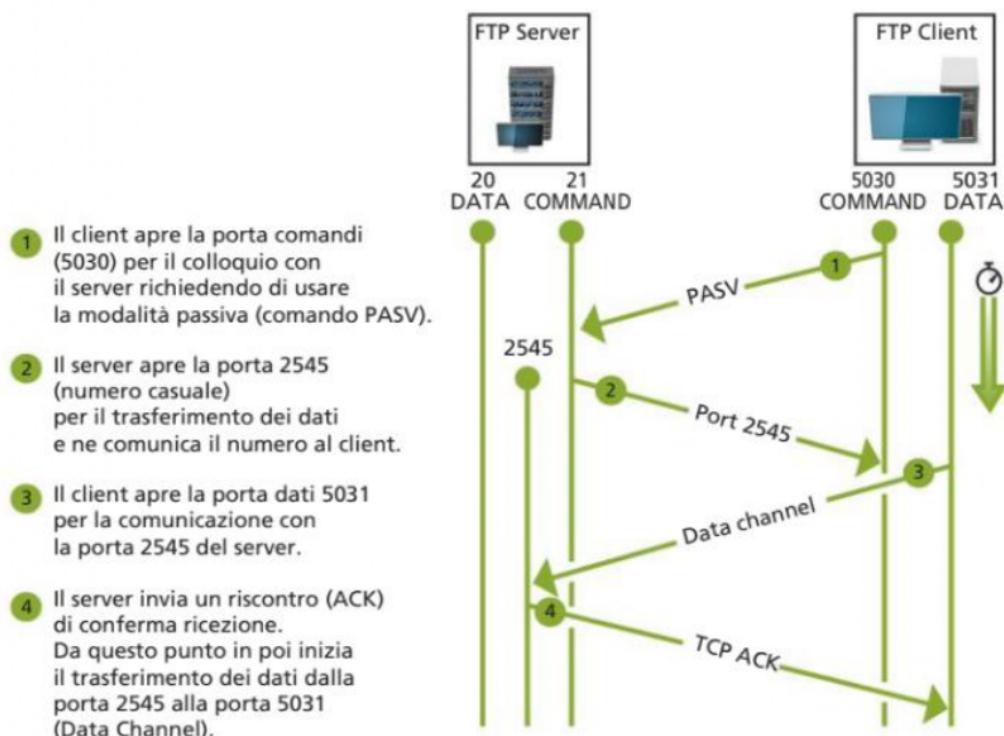


Questa modalità comporta problemi di sicurezza nel lato client: infatti non è il client FTP che si connette alla porta dati del server, esso segnala solo al server su quale porta è in ascolto. Sarà il server che aprirà un canale verso questa porta per l'invio dei dati. Quindi se nel lato client è presente un firewall questi rileva un tentativo di intrusione dall'esterno e lo blocca.

Per questo motivo è attualmente sconsigliato l'uso di FTP active mode, per poter garantire la sicurezza della rete locale (intranet).

### ■ FTP PASSIVE MODE

La **FIGURA 6** mostra lo scambio di messaggi tra client e server FTP nella modalità passiva: il client FTP inizia entrambe le connessioni con il server, sia comandi che dati, risolvendo in questo modo il problema del filtraggio della connessione da parte del firewall lato client. Il client apre localmente due porte **N** e **N+1** (con  $N > 1.023$ ) e con la porta **N**, la porta comandi, contatta il server sulla porta 21. A questo punto invia un comando **PASV** che permette al server di aprire una porta casuale **P** (con  $P > 1.023$ ) e inviare il comando **Port P** al client sulla sua porta comandi **N**. Il client inizia allora la connessione dalla sua porta dati **N + 1** alla porta **P** sul server per ricevere i dati.



**FIGURA 6** Modalità passiva di FTP

## 4.3 Le modalità di accesso al server FTP

FTP ha due modalità predefinite di accesso: **utente** e **anonima**.

La modalità utente prevede un accesso al server FTP con username e password, mentre nella modalità anonima si ha accesso come utente **anonymous**. Quest'ultima è molto utilizzata per scambio di dati pubblici, come modulistica, codice, ecc.

La modalità di accesso anonima presenta due limitazioni:

- è necessario limitare l'accesso alle sole informazioni che si vogliono diffondere;
- non deve essere permesso l'uso di FTP server per la distribuzione di materiale di terzi (per esempio si può rendere la cartella di upload accessibile solo in scrittura e non in lettura).

Se possibile, per la sicurezza del sistema, è meglio evitare di usare l'accesso anonimo. Nel caso in cui sia proprio necessario, esso deve essere configurato correttamente

e amministrato con attenzione, soprattutto se si vuole rendere accessibili in upload, quindi scrivibili, delle directory (o cartelle) nelle aree FTP anonymous.

### ■ CLIENT E SERVER FTP

Sono disponibili vari software FTP, a pagamento o distribuiti con licenza open source. Solitamente i software client FTP sono gratuiti e permettono di collegarsi a un server FTP per caricare un file trasferendolo dal proprio computer o per scaricare un file dal server. L'operazione di upload di un file è tipicamente svolta quando si utilizza un servizio di hosting di un sito web e si necessita di caricare le pagine web sul server per pubblicarle. Uno dei software client FTP più diffusi è **Filezilla**, gratuito e disponibile per sistemi Windows, Mac e Linux. Scaricando questo software sul proprio computer è possibile in modo semplice e sicuro caricare i file delle nostre pagine web sul web server del provider. La sicurezza è data dall'utilizzo della crittografia nel trasferimento dei dati, tecnica non prevista nella specifica originale di FTP, come descritto nel paragrafo successivo. Filezilla offre anche il software per il server FTP, ma solo per sistemi Windows. Il sito del progetto da cui scaricare il software e la documentazione è: <https://filezilla-project.org>.

## 4.4 Le vulnerabilità di FTP

I maggiori problemi di sicurezza di FTP sono riconducibili al fatto che le specifiche non prevedono la cifratura delle informazioni scambiate tra client e server:

- **password in chiaro:** le password viaggiano in chiaro attraverso la rete e sono facilmente intercettabili con strumenti come gli sniffer che consentono di analizzare il traffico tra client e server;
- **dati in chiaro:** anche i dati vengono trasferiti senza essere crittografati, anch'essi sono dunque intercettabili.

La soluzione a questi problemi è stata una nuova specifica di FTP denominata **FTP over TLS (FTPS, RFC 4217)** che aggiunge un livello tra Transport (TCP) e Application (FTP), per la gestione della crittografia, utilizzando il protocollo **Transport Layer Security (TLS)**. TLS è una versione più recente del protocollo Secure Sockets Layer (SSL).

Altri problemi di sicurezza sono legati a:

- **sessione in due processi:** la necessità di avere due processi per ogni connessione rende più semplice effettuare manovre malevoli;
- **permessi utente:** i permessi di accesso FTP vanno incrociati con i permessi utente sul server in modo da limitare lo spazio su disco e le operazioni sui file.

Con il diffondersi del World Wide Web, molti utenti preferiscono usare il browser come FTP client. La maggior parte dei browser supporta solo la modalità passiva quando si accede con **ftp://URL**.



### Esercizio commentato

Trasferimento di una cartella con FTP

### FISSA LE CONOSCENZE

- Qual è lo scopo dei protocolli applicativi FTP e TFTP?
- Quali modalità di colloquio tra un client FTP e un server FTP possono essere implementate?
- Quali sono le porte Well Known utilizzate per FTP?
- Quali modalità di accesso al server sono previste in FTP?
- Quali sono le maggiori vulnerabilità del protocollo FTP?

## 5 HTTP: IL PROTOCOLLO PER LE APPLICAZIONI WEB

### 5.1 HTTP e WWW

**HTTP (HyperText Transfer Protocol)** è il protocollo di livello Application usato nell'applicazione Client-Server **WWW (World Wide Web)**, la parte di Internet più usata e cresciuta più velocemente. Il protocollo HTTP regola lo scambio di messaggi tra il web server e il web client (si parla anche di HTTP server e HTTP client o anche di WWW server e WWW client). Nell'uso comune il client corrisponde al browser e il server al sito web.

La **FIGURA 7** mostra un semplice esempio di comunicazione nel WWW che utilizza il protocollo HTTP:

- il **browser** è il programma client dell'applicazione, esempi sono Edge, Firefox, Chrome, Opera. Il browser svolge due funzioni fondamentali: inoltre la richiesta di una pagina web al server (HTTP request) e presenta i dati ricevuti (HTTP response) all'utente. Le pagine web sono create con un linguaggio chiamato **#HTML** (HyperText Markup Language);
- il **web server** contiene le pagine web del sito e risponde alle richieste che riceve dai web client, i più diffusi web server sono Apache, open source e multi-piattaforma, e Internet Information Services (IIS) per i sistemi Windows.

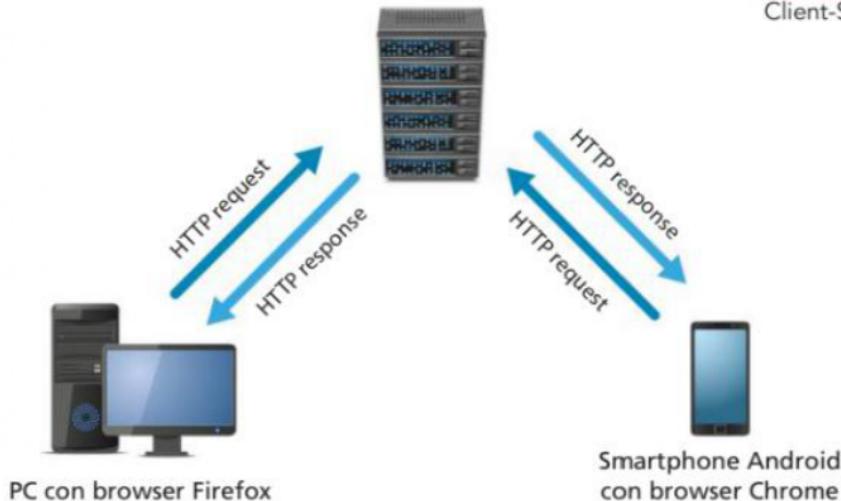
#### #techwords

##### HTML

È il linguaggio nato per realizzare i siti web, utilizzato anche per la creazione di contenuti e di applicazioni mobile. Non è un linguaggio di programmazione ma di markup (contrassegno), che permette di indicare come disporre i vari elementi all'interno di una pagina web.

Server con web server Apache

FIGURA 7 Comunicazione Client-Server con HTTP



Il WWW e i suoi protocolli sono nel tempo diventati la piattaforma di comunicazione per applicazioni quali la posta elettronica, per esempio Gmail, per distribuzione di video, per esempio YouTube, e per la maggior parte delle applicazioni mobile che usano Internet.

### GLI STANDARD HTTP/1.0, HTTP/1.1, HTTP/2 E HTTP/3

La prima versione del protocollo HTTP (HTTP/1.0) è stata standardizzata in **RFC 1945** in cui sono definite le modalità di scambio dei messaggi tra client e server e la struttura

## #prendinota

**W3C** è la più importante organizzazione internazionale per il WWW, formata da aziende informatiche, operatori telefonici, organizzazioni no-profit, università e centri di ricerca. Oltre alla definizione degli standard, sviluppa software open source per il WWW.



ra dei messaggi HTTP. Lo standard fu il risultato del lavoro congiunto di IETF e di **W3C (World Wide Web Consortium)**.

## IN ENGLISH PLEASE

Network Working Group

**Request for Comments: 1945**

Category: Informational

T. Berners-Lee

MIT/LCS

R. Fielding

UC Irvine

H. Frystyk

MIT/LCS

May 1996

**Hypertext Transfer Protocol -- HTTP/1.0**

## Abstract

The Hypertext Transfer Protocol (HTTP) is an application-level protocol with the lightness and speed necessary for distributed, collaborative, hypermedia information systems. It is a generic, stateless, object-oriented protocol which can be used for many tasks, such as name servers and distributed object management systems, through extension of its request methods (commands). A feature of HTTP is the typing of data representation, allowing systems to be built independently of the data being transferred.

HTTP has been in use by the World-Wide Web global information initiative since 1990. This specification reflects common usage of the protocol referred to as "HTTP/1.0".

In pochi anni, grazie alla diffusione del browser grafico Mosaic, il WWW crebbe enormemente e divennero evidenti alcuni limiti della prima versione di HTTP:

- la mancanza di meccanismi di sicurezza, perché non erano previste l'autenticazione e la crittografia dei dati;
- non era possibile ospitare più siti web sullo stesso server;
- per ogni richiesta era necessario creare una connessione separata con il server; per esempio se nella pagina web erano presenti delle immagini, il client doveva inviare ulteriori richieste al server per scaricarle.

Il protocollo fu ampliato e venne specificata una nuova versione **HTTP/1.1** pubblicata in **RFC 2616** nel 1999. Queste specifiche sono state completamente riviste nel 2014 e descritte nei nuovi **RFC 7230, 7231, 7232, 7233, 7234 e 7235**.

Nel 2015 esce una nuova versione denominata **HTTP/2** e descritta in RFC 7540 e in RFC 8740 che esprime la semantica del web in modo più efficiente e utilizza TLS (Transport Layer Security).

In via di standardizzazione è la nuova versione **HTTP/3** che utilizza UDP in sostituzione di TCP.

È già supportata da browser come Chrome e Firefox dal 2019.

Le nuove versioni HTTP/2 e HTTP/3 non rendono obsolete le precedenti versioni di HTTP.

## 5.2 Le modalità di lavoro di HTTP

Gli **#hyperlink** (collegamenti ipertestuali) permettono una facile navigazione: quando si fa clic su un hyperlink si dirige il browser su una nuova pagina.

Ogni pagina web ha un suo indirizzo simbolico detto **URL** (Uniform Resource Locator), per esempio *http://www.azienda.com/news/* (**TABELLA 1**), dove la parte iniziale **http://** indica al browser il protocollo da usare e la seconda parte **www** indica il servizio o la macchina in rete. Per conoscere l'indirizzo IP corrispondente al nome del computer si utilizza il DNS.

HTTP è avviato da TCP/IP ogni qualvolta l'URL contiene nel primo campo la parola **http**.

**TABELLA 1** HTTP identifica le risorse del WWW mediante un indirizzo simbolico: URL

http://	www.	azienda.com	/news/
indica al browser quale protocollo deve essere usato	identifica il nome di una specifica macchina (il web server)	rappresenta l'entità di dominio (domain entity) del sito web	identifica la cartella dove si trova la pagina web sul server. Se non viene specificato nulla, il browser carica la pagina web di default presente sul server

Quando si vuole leggere una pagina, i livelli superiori del client iniziano una sessione col web server. Il client fa la richiesta della pagina desiderata, il server risponde inviando la risorsa richiesta: il testo, l'audio, il video, i file grafici contenuti in quella pagina. Il client riassembra il tutto e chiude la sessione.

L'HTTP è un protocollo **stateless** (senza memoria) che permette sia la ricerca che il recupero dell'informazione in maniera veloce, seguendo gli hyperlink. La scelta di un protocollo stateless, cioè di un protocollo che non conserva memoria della connessione fatta, è stata necessaria affinché fosse possibile saltare velocemente da un web server a un altro attraverso i link ipertestuali.

HTTP a ogni richiesta di un web client effettua una nuova connessione al web server che viene chiusa al termine del trasferimento dell'oggetto richiesto (pagina HTML, immagine, ecc.).

Il server resta in attesa di una richiesta di connessione sulla sua socket, la porta assegnata di default per HTTP è la 80, salvo che nell'URL sia specificata una porta diversa, per esempio *http://www.azienda.com/news/:8080*.

La caratteristica stateless di HTTP limita l'interazione con l'utente, per esempio se effettuiamo il login su una pagina, nel momento in cui ci spostiamo su un'altra dobbiamo nuovamente inserire le nostre credenziali. La soluzione a questo problema è stata l'introduzione dei **#cookie**, piccoli blocchi di dati memorizzati nel browser che permettono di:

- implementare metodi di autenticazione, usati per esempio per i login;
- memorizzare dati utili alla sessione di navigazione, come le preferenze sull'aspetto grafico o linguistico del sito;
- tracciare la navigazione dell'utente, per esempio per fini statistici o pubblicitari.

### #techwords

#### Hyperlink

Gli ipertesti (hypertext), grazie agli hyperlink, abbandonano la secolare abitudine alla lettura lineare, sequenziale, stabilita dall'autore di un testo, per passare a una lettura che vede il lettore come protagonista, non più come frutto passivo. Ciò che consente questo processo sono gli hyperlink o "parole calde": ad alcuni termini di un ipertesto viene associata la possibilità di collegarsi, col semplice clic del mouse, ad altre parti dell'ipertesto. Il percorso di lettura che ne consegue, quindi, è deciso dal lettore. Si può pensare la lettura di un ipertesto come una forma di lettura ramificata, che, di link in link, porta il lettore direttamente al cuore di ciò che gli interessa leggere.

### #techwords

#### Cookie (biscotto)

Sono file di testo di piccola dimensione inviati da un web server a un web client e poi rimandati indietro dal client al server, senza subire modifiche, ogni volta che il client accede allo stesso server. Poiché possono essere usati per monitorare la navigazione su Internet, i cookie sono oggetto di discussioni concernenti il diritto alla privacy.

## 5.3 I metodi e i messaggi di HTTP

L'acquisizione di una risorsa da parte del client può essere schematizzata in 4 fasi:

- **connessione**: il client crea una connessione TCP/IP con il server usando il suo nome di dominio (o l'indirizzo IP) ed eventualmente il numero della porta di trasmissione; come detto, se non viene fornito il numero di porta, il protocollo assume per default che il numero sia 80;
- **richiesta**: il client invia la richiesta di una risorsa (pagina HTML, immagine, ecc.) mediante una riga di caratteri ASCII che termina con una coppia di caratteri CR-LF (Carriage Return, Line Feed);
- **risposta**: la risposta inviata dal server è un messaggio in linguaggio HTML nel quale è contenuta la risorsa richiesta o una segnalazione d'errore;
- **disconnessione**: il server subito dopo aver spedito la risorsa richiesta si disconnette. Anche il client può interrompere la connessione in ogni momento; in questo caso il server non registrerà nessuna condizione d'errore.

Il protocollo HTTP mette a disposizione del client una serie di metodi. Un **metodo** HTTP può considerarsi un comando, proprio del protocollo HTTP, che il client invia come richiesta al server.

La versione HTTP/1.0 ha 3 metodi obbligatori: GET, HEAD, POST. Alcune implementazioni di HTTP/1.0 ne aggiungono altri due: PUT e DELETE. In HTTP/1.1 sono stati aggiunti altri 3 metodi: OPTIONS, TRACE e CONNECT.

Nel dettaglio:

- GET: richiede una risorsa (pagina HTML, immagine, ecc.) al server; quando un utente fa clic su un hyperlink il client invia una GET al server;
- HEAD: richiede solo l'header senza la risorsa, di fatto viene usato soprattutto per la diagnostica;
- POST: invia informazioni al server, cioè all'URL specificato;
- PUT: richiede l'upload di un file sul server, creandolo o riscrivendolo (se autorizzato);
- DELETE: richiede la cancellazione di un file sul server (se autorizzato);
- OPTIONS: richiede l'elenco dei metodi permessi dal server;
- TRACE: traccia una richiesta, visualizzando come viene trattata dal server;
- CONNECT: richiede una connessione mediante proxy, utilizzata, per esempio, per la creazione di un tunnel.

Vi sono due tipi di messaggi HTTP: messaggi richiesta (request) da parte del client e messaggi risposta (response) da parte del server.

### ■ IL MESSAGGIO REQUEST

È composto dalle seguenti 3 parti:

1. riga di richiesta (request line);
2. sezione header (informazioni aggiuntive);
3. body (contenuto della richiesta).

La riga di richiesta è composta da metodo, URI e versione del protocollo. **URI** sta per Uniform Resource Identifier e indica l'oggetto della richiesta.

Per esempio per ottenere una pagina web la richiesta è: **GET /info.html HTTP/1.1**.

Gli header di richiesta più comuni sono:

- **Host:** nome del server a cui si riferisce l'URI;
- **User-Agent:** identificazione del tipo di client: browser, produttore, versione, ecc.

## ■ IL MESSAGGIO RESPONSE

È composto dalle seguenti 3 parti:

- 1. riga di stato:** contiene un codice di risposta a 3 cifre in cui la prima cifra specifica il tipo di stato:

- **1xx:** Informational (messaggi informativi);
- **2xx:** Success (la richiesta è stata soddisfatta);
- **3xx:** Redirection (non c'è risposta diretta, ma la richiesta è ritenuta corretta e viene detto come ottenere la risposta);
- **4xx:** Client error (la richiesta non può essere soddisfatta perché sbagliata);
- **5xx:** Server error (la richiesta non può essere soddisfatta per un problema interno del server).

- 2. header:** contengono informazioni aggiuntive. Quelli più comuni sono:

- **Server:** indica il tipo e la versione del server. Può essere visto come l'equivalente dell'header di richiesta User-Agent;
- **Content-Type:** indica il tipo di contenuto restituito. Essi sono detti tipi MIME (Multimedia Internet Message Extensions, presenti anche nella posta elettronica, come descritto nella Lezione successiva). Esempi di tipi MIME sono:
  - text/html (documento HTML);
  - text/plain (documento di testo non formattato);
  - text/xml (documento XML);
  - image/jpeg (immagine in formato JPEG).

- 3. body:** è la parte in cui si trova il contenuto della risposta. I codici di risposta più comuni sono:

- **200 OK:** il server ha fornito correttamente il contenuto nella sezione body;
- **400 Bad Request:** la richiesta non è comprensibile al server;
- **403 Forbidden:** il client non è autorizzato a ricevere i dati richiesti;
- **404 Not Found:** la risorsa richiesta non è stata trovata e non se ne conosce l'ubicazione;
- **500 Internal Server Error:** il server non è in grado di rispondere alla richiesta per un suo problema interno;
- **505 HTTP Version Not Supported:** la versione di HTTP non è supportata.

Un server HTTP ha il compito (che può risultare computazionalmente dispendioso) di rispondere a tutte le richieste che giungono dalla rete. Si pensi che WWW server di siti professionali raggiungono facilmente le 300.000 richieste al giorno.

La versione HTTP/1.1 ha permesso di aumentare l'efficienza consentendo di utilizzare la stessa connessione TCP/IP per effettuare operazioni multiple.

## 5.4 I proxy HTTP

Un web server e un web client possono utilizzare un **#proxy HTTP** (detto anche *proxy server*) per gestire lo scambio di messaggi. La presenza di un proxy server fa sì che le richieste HTTP dei client vengano automaticamente indirizzate al proxy.

### #techwords

#### Proxy

È un programma che si interpone tra un client e un server facendo da tramite o interfaccia. Il client si collega al proxy, invece che al server, e gli invia delle richieste. Il proxy, a sua volta, si collega al server e inoltra la richiesta del client, poi, ricevuta la risposta, la inoltra al client.

I proxy nella maggior parte dei casi lavorano a livello Application.

Un proxy HTTP può essere usato per diversi motivi:

- **connettività:** un proxy server può essere configurato per permettere a una rete privata di accedere a Internet con un unico computer, cioè un computer fa da proxy tra gli altri computer e Internet;
- **privacy:** un proxy server può garantire un maggiore livello di privacy mascherando il vero indirizzo IP del client in modo che il server remoto non venga a conoscenza di chi ha effettuato la richiesta;
- **caching:** un proxy server può immagazzinare per un certo tempo i risultati delle richieste di un client e se un altro client effettua le stesse richieste, può rispondere senza dover consultare il server originale. Collocando il proxy in una posizione "vicina" (prossima) ai client, questo permette un miglioramento delle prestazioni e una riduzione del consumo di ampiezza di banda;
- **monitoraggio:** un proxy server può permettere di tenere traccia di tutte le operazioni effettuate (per esempio, tutte le pagine web visitate), consentendo statistiche e osservazioni dell'utilizzo della rete che possono anche violare la privacy degli utenti;
- **amministrazione:** un proxy server può applicare regole definite dall'amministratore di sistema per determinare quali richieste inoltrare e quali rifiutare, può limitare l'ampiezza di banda utilizzata dai client oppure filtrare le pagine web in transito, per esempio bloccando quelle il cui contenuto è ritenuto offensivo in base a determinate regole.

I server esterni a cui si collega il client quando si utilizza un proxy vedranno generalmente l'indirizzo IP del proxy (e non quello del client). Se l'uso di un proxy garantisce una relativa privacy del client (il server esterno, o chi analizzi il traffico diretto a esso, non potrà infatti conoscere l'indirizzo IP del client), può impedire la connessione a quei siti che utilizzino l'indirizzo IP del client per scopi di autenticazione o di riconoscimento delle sessioni (come per esempio nei collegamenti agli sportelli bancari online).

Il protocollo HTTP prevede però che un proxy server possa inserire nelle richieste che inoltra al server degli header standardizzati, che permettono di riconoscere che la richiesta è stata inoltrata da un proxy e possono contenere anche l'indirizzo IP del client, che in questo modo può essere noto a un server remoto opportunamente configurato. Quando viene usata questa funzionalità, il web server remoto si fida dell'indirizzo del client inviatogli dal proxy server non potendo in alcun modo verificare questa informazione. L'amministratore di un proxy server può decidere se inviare o meno questi header determinando quindi il livello di anonimato del proxy.

I proxy HTTP, a seconda dell'anonimato che riescono a fornire, possono essere suddivisi in:

- **NOA** (NO Anonymous Proxy Server) proxy non anonimi (o trasparenti): modificano alcuni header trasmessi dal browser e ne aggiungono altri, mostrano anche l'indirizzo IP reale del richiedente. Sono facili da riconoscere da parte del web server;
- **ANM** (Anonymous Proxy Server) proxy anonimi: non trasmettono l'IP del richiedente, ma modificano o aggiungono alcuni header. Sono pertanto facilmente riconoscibili;

- **HIA** (High Anonymous Proxy) proxy altamente anonimi (o élite): non trasmettono l'IP del richiedente e non modificano gli header della richiesta. Sono difficili da riconoscere attraverso i normali controlli;
- **proxy distorcenti**: trasmettono un IP casuale, diverso da quello del richiedente e modificano o aggiungono alcuni header. Solitamente vengono scambiati per proxy anonimi, ma offrono una protezione maggiore, in quanto il web server remoto vede le richieste di un utente provenienti da indirizzi IP diversi.

Per vedere se il proxy server consente una navigazione anonima, ossia se non rivela l'IP del client a nessun altro server della rete, è bene effettuare un **whois** (Lezione 7 dell'Unità 5). Il server del sito per il whois deve restituire l'IP del proxy server; se invece rende visibile un IP diverso, presumibilmente si tratta di quello del client e il test è fallito.

## 5.5 La sicurezza con HTTPS

Per garantire la sicurezza nelle transazioni commerciali o in generale nel trasferimento di dati sensibili, si usa il protocollo **HTTPS** (HyperText Transfer Protocol over Secure Sockets Layer).

Le differenze tra HTTPS e HTTP sono sostanzialmente due:

- l'utilizzo della porta 443 al posto della 80;
- l'applicazione del protocollo TLS/SSL.

In pratica tra il protocollo TCP e il protocollo HTTP si interpone un livello di crittografia/autenticazione come il Secure Sockets Layer (SSL) o il Transport Layer Security (TLS), in modo da impedire intercettazioni dei contenuti.

Infatti, viene implementata una tecnica di crittografia asimmetrica che utilizza chiavi private e pubbliche a lungo termine, per generare chiavi di sessione a breve termine. Queste chiavi sono utilizzate successivamente per cifrare il flusso dei dati scambiati tra client e server.

Un sito web non può avere dei contenuti accessibili con HTTPS e altri con HTTP, per esempio la pagina di login su HTTPS e le altre pagine su HTTP: ciò implicherebbe una vulnerabilità a possibili attacchi.

Inoltre, se il sito è su HTTPS, anche i cookie devono essere trasmessi in modo sicuro. È quindi necessario impostare un parametro, chiamato **Secure attribute**, che segnala al browser di inviare il cookie solo su HTTPS e mai su HTTP.

### FISSA LE CONOSCENZE

- Qual è il compito del protocollo HTTP?
- Che cos'è un hyperlink?
- Che cos'è un metodo HTTP?
- Quali sono gli 8 metodi dell'HTTP/1.1?
- Quali sono i 2 tipi di messaggi HTTP?
- Qual è il ruolo del proxy server in una comunicazione HTTP?

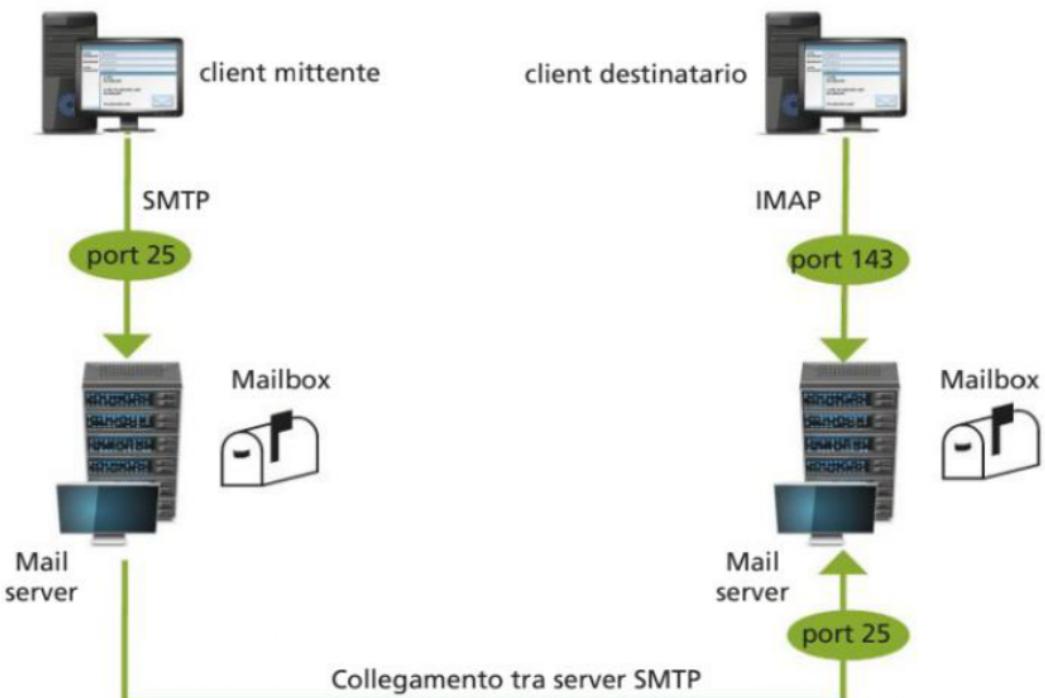
## 6 SMTP, POP E IMAP: I PROTOCOLLI PER LA POSTA ELETTRONICA

### 6.1 Invio e ricezione di email

La posta elettronica (electronic-mail o email) è una delle prime applicazioni nate con Internet e continua a essere una delle più importanti e utilizzate.

La **FIGURA 8** mostra le fasi di invio, trasmissione in rete e ricezione di una email.

**FIGURA 8** Invio e ricezione di una email con i protocolli SMTP e IMAP



Sono evidenziate le componenti principali di un sistema di posta elettronica:

- **Mail client**, è l'applicazione di email utilizzata dall'utente per inviare/ricevere email, per esempio Outlook o Thunderbird;
- **Mail server**, è l'applicazione di email che risiede sui server: riceve e inoltra i messaggi, gestisce le caselle di posta, **mailbox**, degli utenti; l'insieme dei server costituisce l'infrastruttura del sistema di posta elettronica;
- i **protocolli**, sono stati definiti più protocolli per la posta elettronica: **SMTP** (Simple Mail Transfer Protocol) per l'invio delle email e per la comunicazione tra i mail server, **POP3** (Post Office Protocol, version 3) e **IMAP4** (Internet Message Access Protocol version 4) per la ricezione delle email.

#### ■ LA POSTA ELETTRONICA SUL WEB

Un'alternativa allo scenario presentato in Figura 8 è il sistema **web-based email**, o **webmail**, nel quale l'utente utilizza il browser per inviare e ricevere le email. Il primo a offrire un servizio di webmail fu Hotmail, a metà degli anni Novanta, seguito poi da altri, come Google Gmail e Yahoo! Mail.

In questo scenario cambia il client di email: non è più un programma, ma un'interfaccia utente fornita tramite pagine web. Quando l'utente accede alle pagine web del

servizio di email gli viene chiesto di autenticarsi con login e password. Queste credenziali sono inviate al server che le valida, costruisce sul momento una pagina web con il contenuto della mailbox e la invia all'utente.

L'invio e la ricezione dei messaggi avviene quindi con il protocollo HTTP, essendo una comunicazione tra web client e web server. Il web server si occuperà poi di introdurre i messaggi nel tradizionale sistema di posta elettronica basato sul protocollo SMTP.

## ■ GLI INDIRIZZI DELLA POSTA ELETTRONICA

Ogni utente (client) è individuato da un indirizzo di posta composto dal user ID dell'utente seguito dal simbolo @ e dal dominio del gestore del servizio di posta elettronica:

*nomeutente@dominiogestoreservizio*

Per esempio: bianchi@azienda.com.

Se il client mittente e il client destinatario usufruiscono dello stesso fornitore del servizio email (quindi hanno lo stesso dominio, per esempio azienda.com) allora il server SMTP ha il compito semplificato perché con un semplice programma, chiamato delivery agent, può direttamente depositare la email nella mailbox. Se invece i fornitori sono diversi, e dunque sono diversi i domini, il server SMTP del mittente deve interrogare il **DNS** (Domain Name System) per risalire dal nome di dominio all'indirizzo IP del server SMTP del destinatario. Per associare il server SMTP a un dato nome di dominio si usa un Resource Record di tipo MX (Mail eXchange), come visto nella Lezione 6 dell'Unità 7.

## 6.2 Il protocollo SMTP

Il protocollo SMTP gestisce il trasferimento del messaggio di posta elettronica dal mittente al destinatario.

La prima versione del protocollo SMTP è del 1982 contenuta nell'RFC 821, ma era già utilizzato da molti anni dagli utenti di Internet. SMTP è stato revisionato nel 2008, **RFC 5321**, con successivi aggiornamenti riguardanti l'uso dei codici di risposta.

### IN ENGLISH PLEASE

Network Working Group

J. Klensin

**Request for Comments: 5321**

October 2008

Obsoletes: 2821

Updates: 1123

Category: Standards Track

### Simple Mail Transfer Protocol

#### Abstract

This document is a specification of the basic protocol for Internet electronic mail transport. It consolidates, updates, and clarifies several previous documents, making all or parts of most of them obsolete. It covers the SMTP extension mechanisms and best practices for the contemporary Internet, but does not provide details about particular extensions. Although SMTP was designed as a mail transport and delivery protocol, this specification also contains information that is important to its use as a "mail submission" protocol for "split-UA" (User Agent) mail reading systems and mobile environments.

### #prendinota

L'ingegnere informatico americano Ray Tomlinson nel 1971 inventò la posta elettronica elaborando un programma che permetteva a tutti coloro che frequentavano le università americane, collegate tra loro tramite la rete ARPANET, di potersi scambiare messaggi scritti. Lo stesso Tomlinson nel 1972 usò il simbolo @ (at, cioè "presso" in inglese, chiocciola in italiano) come separatore tra il nome del destinatario e il server che svolgeva le funzioni di cassetta della posta. Nel marzo del 2010, Paola Antonelli, Senior Curator del Department of Architecture and Design del MoMA di New York, ha reso noto che la chiocciola è stata inserita nella collezione, perché non è soltanto uno strumento utilizzato in informatica, ma è un mezzo di comunicazione e una forma della nostra identità.

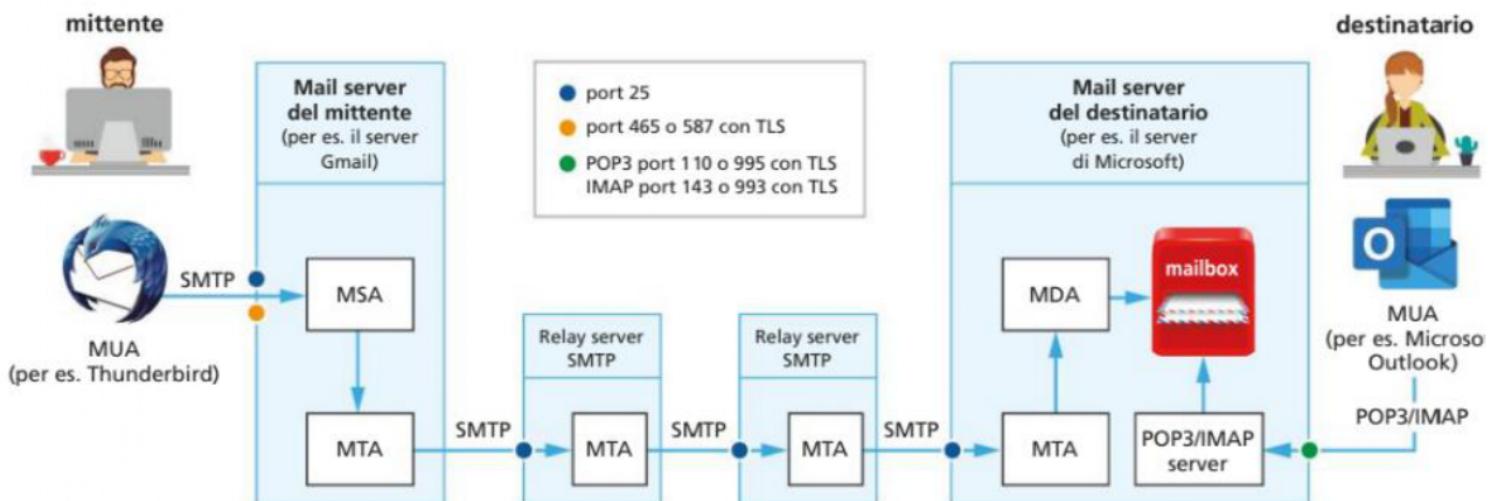
La **FIGURA 9** mostra il sistema di email su Internet. Come descritto nelle specifiche di SMTP si usano i seguenti termini per riferirsi al software utilizzato su client e server:

- **Mail User Agent (MUA)** è il software client sul computer dell'utente, il mittente è chiamato **sender** e il destinatario **recipient**; questo programma offre un'interfaccia grafica con funzionalità per la composizione, lettura e organizzazione delle email;
- **Mail Submission Agent (MSA)** è il software che risiede sul mail server al quale l'utente invia una email tramite il MUA;
- **Mail Delivery Agent (MDA)** è il software che risiede sul mail server e gestisce la mailbox dalla quale l'utente legge la email ricevuta;
- **Mail Transfer Agent (MTA)** è il software che risiede sui mail server che si occupano dell'inoltro della email all'interno del sistema di posta elettronica.

I distinti ruoli che ciascun software MTA, MSA e MDA assume nella trasmissione di un messaggio di posta elettronica possono essere svolti da uno stesso programma che risiede nel server e implementa più funzionalità.

Nei sistemi di posta webmail il MUA è rappresentato dall'interfaccia web eseguita nel browser del computer dell'utente.

**FIGURA 9** Il sistema di posta elettronica



Analizziamo le fasi di invio e ricezione di una email mostrate nella Figura 9 entrando nel dettaglio di ciò che succede sui server:

- **invio:** il mittente (sender) invia al server una email utilizzando il software MUA presente sul suo dispositivo, questa viene presa in carico dal mail server che gestisce la sua mailbox, in particolare dal software MSA, con una connessione TCP alla porta 25 del server. MSA invia il messaggio al software MTA, presente sullo stesso mail server, affinché lo inoltri al mail server che gestisce la mailbox del destinatario (recipient);
- **transito:** l'MTA del mittente apre una connessione TCP verso l'MTA del mail server di destinazione, sempre tramite la porta 25, sulla quale trasmette il messaggio di posta elettronica. Nel mail server di destinazione, il messaggio è inviato da MTA al software MDA che lo memorizza nella mailbox del destinatario. A volte la connessione tra i due mail server non è diretta, ma il messaggio passa attraverso più MTA presenti su server intermedi, chiamati **relay server**;
- **ricezione:** il client destinatario utilizza il software MUA per connettersi alla porta 110 del server POP3 o alla porta 143 per IMAP4 e accedere alla sua mailbox per leggere il messaggio ricevuto.

Qualora il server SMTP destinatario non risulti raggiungibile, il server SMTP mittente prova a rispedire la email per un certo periodo di tempo. Se l'invio continua a fallire, il server mittente invia al client mittente una segnalazione di mancato recapito (badmail o delivery failure).

## ■ LA SICUREZZA DI SMTP

Il protocollo SMTP non garantisce né l'autenticazione del mittente né l'autorizzazione all'invio, chiunque può spedire una email a chiunque.

Per questi motivi lo **#spamming** non può essere in alcun modo limitato ed è inoltre possibile inviare email facendo apparire come mittente l'indirizzo corrispondente a un altro account.

Inoltre, nelle specifiche originarie di SMTP, non è previsto l'impiego della crittografia e i messaggi sono trasmessi in chiaro nella rete.

Questo problema è stato risolto con l'utilizzo di connessioni TLS (Transport Layer Security) e la registrazione presso IANA di un'altra porta, la 465, da utilizzare in sostituzione della porta 25.

La maggior parte dei provider non utilizza più la porta 25 per la connessione tra client e server SMTP sia perché affetta dal traffico di spam e malware, sia perché la richiesta di un servizio di confidenzialità implica l'utilizzo di connessioni TLS che usano la porta **587 (RFC 6409)** o la porta **465 (RFC 8314)**. La porta 25 continua a essere usata per le connessioni tra i relay server SMTP.

La porta 587 è stata definita per sostituire la porta 25, introducendo un nuovo comando STARTTLS da utilizzare quando il client vuole stabilire una connessione sicura con TLS. Infatti, mentre con la porta 465 è implicito l'uso di TLS, con la porta 587 deve essere esplicitamente richiesto con il comando STARTTLS.

## ■ IL FORMATO DELLE EMAIL

Il formato dei messaggi di posta elettronica è definito nell'**RFC 5322**.

Una email è formata da un header (envelope) e il testo del messaggio (content o body).

L'**envelope** contiene le informazioni utili per la spedizione del messaggio. Lo standard specifica le parole chiave da usare in ogni linea, separate da una virgola, alcune sono obbligatorie, per esempio From e To, altre opzionali, per esempio Subject.

Un tipico header di una email è:

From: [bianchi@azienda.com](mailto:bianchi@azienda.com)  
To: [rossi@academy.edu](mailto:rossi@academy.edu)  
Subject: Richiesta articolo

Dopo l'envelope si inserisce una riga vuota, da lì in avanti c'è il testo del messaggio in ASCII (American Standard Code for Information Interchange).

SMTP è stato progettato per incidere il meno possibile sull'occupazione del canale. Per questo motivo utilizza il codice **ASCII** a 7 bit per codificare i caratteri del messaggio.

### #techwords

#### Spamming

È l'invio indiscriminato di messaggi di posta elettronica. La mailbox dell'utente viene inondata di email spesso pubblicitarie, ma anche contenenti link dannosi e allegati con virus o malware. Il nome SPAM deriva da **Shoulder of Pork And ham** (spalla di maiale e prosciutto), carne in scatola immessa sul mercato nel periodo della Seconda guerra mondiale, diventata sinonimo di fastidiosa e non richiesta valanga di materiale.

Per inviare dei caratteri accentati, o comunque non compresi nei primi 128 caratteri codificati dall'ASCII a 7 bit, bisogna ricorrere ad algoritmi che integrino le specifiche **#MIME** (Multipurpose Internet Mail Extensions):

- **base64** per i file in allegato;
- **quoted-printable** (abbreviazione QP) per i caratteri speciali contenuti nel corpo del messaggio.

**esercizio****→ PROBLEMA**

Nella propria casella di posta elettronica, selezionare un messaggio con un file pdf allegato ed esaminarne il contenuto codificato in **#MIME**. Ripetere l'analisi su messaggi con altri tipi di allegati, immagini, video, ecc.

**#techwords****MIME e MIME type**

Lo standard MIME fu proposto da Bell Communication nel 1991 per sopperire alle limitazioni di SMTP e consentire agli utenti di inviare email con caratteri non-ASCII7 e file contenenti immagini, audio, video e programmi (file binari).

MIME è utilizzato anche dai web server per comunicare al browser il tipo di dati che gli viene inviato nella risposta (MIME type).

**→ SVOLGIMENTO**

Nel servizio di posta **Gmail** per vedere il contenuto del messaggio nel formato MIME si deve aprire il messaggio e selezionare la voce Mostra originale nel menu con i tre puntini. La stessa procedura si può seguire con il servizio di posta elettronica **Yahoo! Mail** selezionando la voce Visualizza messaggio in formato Raw.

Il testo che compare è il seguente, relativo a una email con un file pdf allegato.

```

MIME-Version: 1.0
Date: Sun, 24 Gen 2021 09:49:26 +0100
Message-ID: <CAAO7sbcZDryMw4-ZWHj5PaLs6eTVJOur8SrcyjEHpzNhrv87g@mail.
gmail.com>
Subject: PROVA
From: user1@gmail.com
To: user2@yahoo.com
Content-Type: multipart/mixed; boundary="0000000000003dba8005b4a92079"

--0000000000003dba8005b4a92079
Content-Type: multipart/alternative; boundary="0000000000003dba7d05b4a92077"

--0000000000003dba7d05b4a92077
Content-Type: text/plain; charset="UTF-8"

Messaggio di prova con allegato

--0000000000003dba7d05b4a92077
Content-Type: text/html; charset="UTF-8"

<div dir="ltr">Messaggio di prova con allegato<br></div>

--0000000000003dba7d05b4a92077-
--0000000000003dba8005b4a92079
Content-Type: application/pdf; name="ocument.pdf"
Content-Disposition: attachment; filename="ocument.pdf"
Content-Transfer-Encoding: base64
X-Attachment-Id: f_khsiuwid0
Content-ID: <f_khsiuwid0>

--0000000000003dba8005b4a92079-

```

Vediamo i vari campi presenti:

- MIME-Version:1.0, indica che il messaggio è nel formato MIME;
- Content-Type: multipart/mixed, indica che sono presenti parti in formato testo e altre parti non testuali;
- Content-Type: multipart/alternative, indica che il messaggio è inviato in più formati, nel nostro esempio il contenuto del messaggio si presenta sia come normale testo sia in html;
- boundary="...", è una stringa che separa le parti del messaggio MIME, nel nostro esempio sono definite due stringhe, una delimita le due parti del contenuto, plain e html, l'altra delimita la parte che contiene l'allegato;
- Content-Type: text/plain, specifica il tipo di formato, text, e il sottotipo, plain, contenuto in quella parte del messaggio; text/plain è il formato predefinito di Content-Type;
- Content-Type: text/html, specifica il tipo di formato, text, e il sottotipo, html, contenuto in quella parte del messaggio;
- Content-Type: application/pdf, specifica il tipo di formato, application, e il sottotipo, pdf, del file allegato, il cui nome è specificato nel campo name. Altri formati sono previsti per i vari tipi di allegati, per esempio: image/jpeg, audio/mp3, video/mp4;
- Content-Disposition: attachment, indica di presentare il file come un allegato con il nome specificato nel campo filename; l'alternativa ad attachment è inline, che indica di visualizzare automatico il contenuto del file quando la email viene aperta (si usa soprattutto per le immagini);
- Content-Transfer-Encoding: base64, indica che il contenuto del messaggio è stato codificato secondo lo schema base64 e quindi il client deve operare la necessaria decodifica per consentire all'utente di leggere il messaggio nella sua codifica originale, per esempio UTF-8.

## I PRINCIPALI COMANDI DI SMTP

I comandi SMTP hanno il seguente formato:

keywords : parametri

Non tutti i comandi prevedono dei parametri.

La **TABELLA 2** elenca i comandi principali definiti per la comunicazione dal client verso il server SMTP.

**TABELLA 2** I principali comandi SMTP

Comando	Esempio	Descrizione
EHLO (sostituisce HELO)	EHLO 193.56.47.125	Identifica il computer mittente attraverso l'indirizzo IP o il nome del dominio.
EMAIL FROM:	MAIL FROM: mittente@dominio1.com	Specifica il mittente del messaggio.
RCPT TO:	RCPT TO: destinatario@dominio2.com	Specifica il destinatario del messaggio.
DATA	DATA messaggio	Indica l'inizio del contenuto del messaggio, che sarà inviato linea per linea.
QUIT	QUIT	Chiude la connessione TCP con il server SMTP.

Il server SMTP a sua volta spedisce dei messaggi di risposta al client che gli ha inviato i comandi. Ogni risposta inizia con un codice identificativo di 3 cifre, optionalmente seguito da un testo informativo. Alcuni codici e il relativo testo sono elencati nella **TABELLA 3**.

**TABELLA 3** Codici di alcune risposte inviate dal server SMTP al client

Categoria	Codici	Descrizione
Positive Completion Reply	2xx	Informa che l'azione richiesta dal client è stata portata a termine con successo.
Positive Intermediate Reply	3xx	Informa il client che il comando è stato accettato, ma l'azione richiesta è in sospeso.
Transient Negative Completion Reply	4xx	Informa il client che il comando non è stato accettato per una situazione di errore temporanea, il client può provare a inviare di nuovo il comando.
Permanent Negative Completion Reply	5xx	Informa il client che il comando non è stato accettato e che il server non è in grado di eseguire l'azione richiesta.

## 6.3 Il protocollo POP

### #prendinota

SMTP è un protocollo **push** (spingi): il mail server del mittente invia il file al mail server del destinatario. Il destinatario ottiene il messaggio con un'operazione **pull** (estrai), quindi sono stati definiti i protocolli di accesso POP3 e IMAP4, o HTTP per la webmail.

Il protocollo POP è il primo a essere stato definito per l'accesso al server di posta per scaricare i messaggi dalla mailbox. Ha subito varie modifiche dalla sua prima versione, l'attuale versione è **POP3** definita nell'**RFC 1939**.

Con POP3 i messaggi di posta elettronica, per essere letti, vengono scaricati in locale sul computer e cancellati dal server. Questo risulta particolarmente utile qualora il client abbia convenienza a leggere le email offline, ma se si usa una webmail non sarà più possibile leggere le email dopo averle scaricate.

È comunque sempre possibile configurare il client per lasciare una copia del messaggio nella mailbox del server.

Le porte definite per POP3 sono la **110** e la **995** per le connessioni TLS.

Questo protocollo gestisce l'autenticazione attraverso **username** e **password**. Quest'ultima, come le email, non è cifrata. Per poter codificare la password e beneficiare di un'autenticazione sicura è possibile selezionare un servizio opzionale che solo pochi server implementano.

Il protocollo POP3 blocca la casella postale durante la consultazione al fine di evitare una consultazione simultanea da due utenti.

I principali comandi POP3 sono riportati nella **TABELLA 4**.

**TABELLA 4** I principali comandi POP3

Comando	Descrizione
USER identificativo	Questo comando permette di autenticarsi. Esso deve essere seguito dal nome dell'utente, cioè da una stringa di caratteri che identificano l'utente sul server. Il comando USER deve precedere il comando PASS.
PASS password	Il comando PASS permette di indicare la password dell'utente il cui nome è specificato nel comando USER precedente.
STAT	Informazione sui messaggi contenuti sul server.
RETR	Numero di messaggi da recuperare.
DELE	Numero di messaggi da cancellare.
LIST [msg]	Numero di messaggi da visualizzare.
NOOP	Permette di mantenere le connessioni aperte in caso di inattività.
TOP <messageID> <n>	Comando che visualizza n linee di messaggio, dove n è dato in argomento. In caso di risposta positiva da parte del server, questo rinvia le intestazioni del messaggio, poi una linea vuota e infine le n prime linee del messaggio, indipendentemente dalla sessione.
QUIT	Chiede l'uscita del server POP3. Esso implica la cancellazione di tutti i messaggi segnati come eliminati e rinvia lo stato di questa azione.

## 6.4 Il protocollo IMAP4

IMAP4 è un protocollo di accesso per leggere i messaggi ricevuti nella mailbox, come POP3. Questo protocollo è particolarmente indicato per i client in grado di mantenere una connessione continua a un server (online), infatti permette di sincronizzare il client con il server.

IMAP4 è stato definito nell'**RFC 3501** per superare le limitazioni di POP3, infatti offre molte più possibilità:

- accesso alla posta sia online sia offline: il client rimane connesso e risponde alle richieste che l'utente fa attraverso l'interfaccia; questo permette di risparmiare tempo se ci sono messaggi di grandi dimensioni;
- più utenti possono utilizzare la stessa casella di posta: permette connessioni simultanee alla stessa mailbox, fornendo meccanismi per controllare i cambiamenti apportati da ogni utente;
- accesso a singole parti di un messaggio: la maggior parte delle email sono trasmesse nel formato MIME, che permette una struttura ad albero del messaggio, dove ogni ramo è un contenuto diverso (intestazioni, allegati o parti di esso, messaggio in un dato formato, ecc.). Il protocollo IMAP4 permette di scaricare una singola parte MIME o addirittura sezioni delle parti, per avere un'anteprima del messaggio o per scaricare una email senza i file allegati;
- informazioni sui messaggi presenti nella mailbox: ogni singolo client può tenere traccia di ogni messaggio, per esempio per sapere se è già stato letto o se ha avuto una risposta;
- organizzazione in cartelle (folder) delle email ricevute: si possono creare, modificare o cancellare cartelle sul server.

Con IMAP4 i messaggi, sia della cartella Posta in arrivo sia delle altre cartelle, rimarranno sempre sul server e sul computer client ne sarà scaricata soltanto una copia. Si potrà quindi accedere alla propria casella da più dispositivi e ritrovare tutte le email, purché tutti gli accessi avvengano via IMAP4 o webmail e non con POP3.

Le porte definite per IMAP4 sono la **143** e la **993** per le connessioni TLS.



### LABORATORIO ONLINE

#### TELNET E LA POSTA ELETTRONICA

L'attività di laboratorio consiste nell'inviare una email utilizzando una sessione Telnet tra il nostro computer client e il server SMTP, tramite la porta 25, e poi, tramite la porta 110, collegarsi al server POP3 per ricevere la stessa email.



### LABORATORIO ONLINE

#### WIRESHARK: ANALISI DI HTTP, SMTP, POP3

L'attività di laboratorio consiste nell'utilizzo di Wireshark per esaminare i dati che vengono scambiati tra un web client e un web server con HTTP e i dati scambiati tra client e server di posta elettronica con SMTP e POP3.

### FISSA LE CONOSCENZE

- Descrivi le fasi di invio e ricezione di una email.
- Da quali parti è composto un indirizzo di posta elettronica?
- Quale codice utilizza il protocollo SMTP e perché?
- Il protocollo SMTP non offre garanzie di sicurezza. Perché?
- In che modo il protocollo POP3 gestisce l'autenticazione dell'utente?
- Perché è stato introdotto il protocollo IMAP4?

## **7** I PROTOCOLLI PER LE APPLICAZIONI MULTIMEDIALI

### 7.1 La classificazione delle applicazioni multimediali

Le applicazioni multimediali in rete si occupano del trasferimento di dati di tipo **audio** e **video** attraverso la rete.

La possibilità di trasmettere segnali audio e video in rete consente di avere forme più avanzate di comunicazione, ma richiede un elevato impiego di risorse, decisamente maggiore rispetto ad altri tipi di trasmissione. Infatti, prima di essere trasmessi in rete i segnali audio/video devono essere digitalizzati e compressi.

Tipicamente, le applicazioni multimediali sono classificate in 3 categorie: le applicazioni memorizzate su un server di streaming, le applicazioni live e le applicazioni interattive.

#### ■ STORED STREAMING APPLICATION

I dati multimediali sono memorizzati su un server e trasmessi al client su richiesta (**on demand**). Si dice che sono trasmessi in streaming, perché il client è in grado di visualizzarli subito, prima che il trasferimento sia completato. Questa caratteristica comporta stringenti vincoli temporali, per la consegna dei dati ancora da inviare, affinché la fruizione da parte del client sia adeguata (un ritardo di 5-10 secondi è ancora accettabile). La trasmissione avviene in modalità **unicast** e si adatta al dispositivo e alla disponibilità di banda dell'utente. Lo streaming sul client è visualizzato tramite un media player che cerca di rimuovere i **#jitter**, decomprimere i dati, visualizzare i controlli per un uso interattivo da parte dell'utente.

Alcuni esempi sono: visione di film, serie TV, ascolto di brani musicali.

#### ■ LIVE STREAMING APPLICATION

La trasmissione dei dati di queste applicazioni è simile alla diffusione dei programmi radio e televisivi; la differenza è che la trasmissione avviene attraverso la rete Internet. Anche in questo caso, come nel precedente, si tratta di traffico sensibile al ritardo e non è possibile ritrasmettere i pacchetti. La differenza è che le applicazioni di tipo stored streaming prevedono una trasmissione dei contenuti on demand, mentre in quelle live streaming la trasmissione è in **diretta** e può anche essere in multicast per servizi come IPTV (la trasmissione on demand è invece sempre unicast). In una trasmissione live i contenuti vengono inseriti dal fornitore del servizio, man mano che si rendono disponibili. Ciò comporta un processo di compressione dei dati più veloce e meno ottimizzato, che può tradursi in una maggiore quantità di dati da trasmettere in rete.

Alcuni esempi sono: IPTV, Internet radio, videogiochi online, concerti online.

#### ■ INTERACTIVE APPLICATION

Si tratta di applicazioni di tipo **interattivo**, con esigenze di trasmissione in tempo reale: bassissimo jitter e nessuna ritrasmissione. Tali requisiti sono in genere soddisfatti attraverso il sovrardimensionamento della rete o la definizione di classi di priorità nell'assegnazione della banda (rimane però il problema se il carico della rete aumenta considerevolmente). Alcuni esempi sono: telefonia via Internet, audio/video conferenza.

#### #techwords

**Jitter** indica una variabilità nel tempo di arrivo dei pacchetti. Se il ritardo è costante il destinatario riceverà il messaggio in modo comprensibile. Se invece i pacchetti subiscono ritardi variabili, l'effetto sarà di una comunicazione a scatti.

Questo concetto è stato introdotto nell'Unità 8 del volume 2 a proposito della QoS (Quality of Service) applicata alle comunicazioni di tipo interattivo.

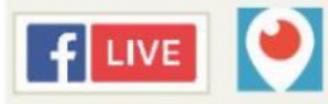
#### #prendinota

##### Streaming Application

Esempi di fornitori di contenuti in streaming on demand sono Amazon Prime video, Hulu, Netflix, Spotify e Youtube.



Facebook Live e Periscope sono esempi di fornitori di live streaming.



## 7.2 Real Time Streaming Protocol

Per soddisfare le esigenze delle applicazioni di streaming audio e video è stato standardizzato il protocollo **RTSP** (Real Time Streaming Protocol), **RFC 7826**, che si colloca al livello Application dello stack TCP/IP e segue il paradigma Client-Server tipico di Internet.

RTSP offre all'utente quei comandi, tipici dei player, che HTTP non è in grado di fornire (play, pause, ecc.), inoltre tiene traccia dello stato del client in ogni sessione (esempi di stato del client sono: riproduzione, fermo immagine, ecc.). A questo scopo, il protocollo RTSP numera le sessioni e questi valori sono usati come identificatori nelle richieste e risposte RTSP, per aiutare il server a mantenere lo stato delle sessioni aperte con i vari client.

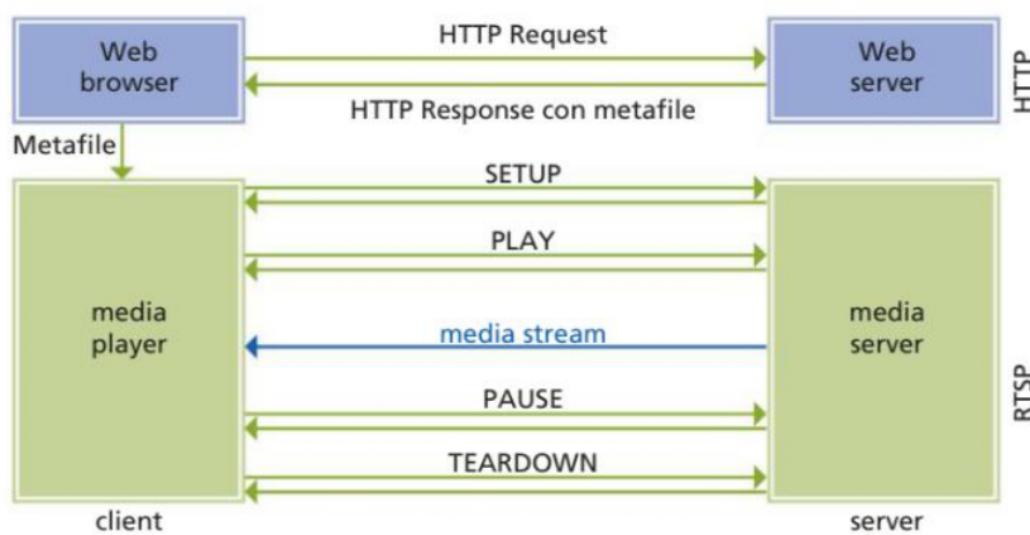
RTSP non definisce come i dati audio/video devono essere incapsulati per realizzare lo streaming sulla rete, né specifica come devono essere trasportati.

La **FIGURA 10** mostra uno scambio di messaggi RTSP tra client e server: il browser del client contatta il web server (HTTP Request), questi invia come risposta un metafile (HTTP response con metafile) contenente le informazioni necessarie per avviare il download dei dati in streaming (URL, tipo di codifica dei dati, ecc.), il browser avvia il player il quale contatta il server per instaurare una sessione RTSP (Setup). Da qui in poi avviene la riproduzione, fino alla chiusura (Teardown).

## #prendinota

Un'applicazione che tratta streaming audio/video deve poter fornire all'utente i comandi che normalmente si usano sui player, ossia: rewind, fast forward, pause, resume, repositioning, ecc.

**FIGURA 10** Le operazioni previste dal protocollo RTSP



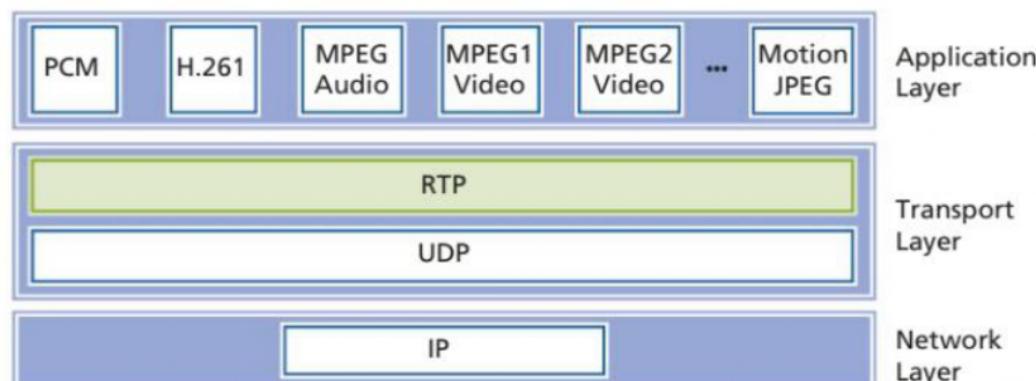
## 7.3 Real Time Transport Protocol

I protocolli di livello Transport tipici di Internet (TCP e UDP) non sono adatti per il traffico generato dalle applicazioni real time interattive. Quindi è stato standardizzato un protocollo ad hoc, per gestire questo tipo di dati: il protocollo **RTP** (Real Time Transport Protocol), **RFC 3550**, che si colloca tra il livello Transport e il livello Application. In particolare si interfaccia con il protocollo UDP dello strato Transport (**FIGURA 11**).

RTP definisce un formato standard per i pacchetti multimediali e deve essere integrato all'interno dell'applicazione:

- RTP è un processo attivo a livello di end system e specifica la struttura che devono avere i pacchetti che trasportano dati audio/video;

**FIGURA 11** La collocazione di RTP nello stack TCP/IP



- i pacchetti RTP sono incapsulati all'interno di una socket UDP;
- in ricezione, i dati applicativi devono essere estratti dai pacchetti RTP e passati al player per la riproduzione.

RTP viene usato insieme a un altro protocollo: **RTCP** (Real Time Transport Control Protocol), che svolge il compito di raccogliere statistiche al fine di ottimizzare le prestazioni. Tutti i partecipanti a una sessione (relativa, per esempio, a una partita online con più giocatori) inviano pacchetti RTCP, siano essi mittenti o destinatari.

I rapporti statistici contengono dati sul numero di pacchetti inviati, persi, jitter, ecc. e sono usati dall'applicazione per modificare la velocità di trasmissione della sorgente. Per evitare che l'invio di pacchetti RTCP da parte di tutti i partecipanti alla sessione crei congestione, è stata definita una semplice regola: la banda totale usata per i pacchetti RTCP deve essere il 5% della banda utilizzata per la sessione RTP e di questa ne viene riservato il 25% al mittente e il 75% ai destinatari.

### esempio

Supponiamo di avere la trasmissione di un video con una banda di 2 Mbps con il 5% della banda riservato ai pacchetti RTCP.

Trasformando 2 Mbps in 2.000 Kbps e applicando il 5% otteniamo che la banda riservata è pari a 100 Kbps.

Dei 100 Kbps viene usato il 75% dai destinatari (75 Kbps).

## 7.4 Le reti CDN per le applicazioni multimediali

Una rete **CDN** (Content Distribution Networks) risponde all'esigenza dei fornitori di servizi di streaming di distribuire i contenuti su più server collocati in varie aree geografiche. La CDN si occupa della gestione di questi server: memorizza una copia dei video da trasmettere e indirizza le richieste degli utenti al server che meglio potrà soddisfarle. Uno stesso video non viene memorizzato su tutti i server della CDN; se un client lo richiede a un server in cui non è presente, questi lo recupererà da un altro server della CDN e mentre lo trasmette in streaming al client, lo memorizza nel suo repository.

Una CDN può essere privata, è il caso di Netflix e di Google per i video di YouTube, oppure di terze parti, un esempio è la CDN di Akamai.

### FISSA LE CONOSCENZE

- Quali sono le problematiche del trasporto su Internet di dati generati da applicazioni multimediali?
- Come si classificano le applicazioni multimediali?
- Quale protocollo è stato standardizzato per il trasporto dei dati delle applicazioni stored streaming?
- Come avviene lo scambio dei messaggi con il protocollo RTSP?
- Spiega com'è organizzata la distribuzione dei contenuti su una rete CDN.

## 8 VoIP: LA TECNOLOGIA PER LA VOCE

### 8.1 L'applicazione Voice over IP

**VoIP (Voice over IP)**, chiamata anche **Internet Telephony**, è un'applicazione real time che utilizza i protocolli della rete IP e le relative tecniche di routing per implementare una rete telefonica distribuita e flessibile.

Le problematiche che questo approccio comporta sono quelle descritte nell'Unità 8 del volume del terzo anno a proposito della **QoS**: ottimizzare le prestazioni della rete in termini di banda, tasso d'errore e di pacchetti persi, latenza e jitter.

Per evitare il meccanismo di acknowledgment e delle eventuali ritrasmissioni del protocollo TCP, le applicazioni VoIP utilizzano **UDP** come protocollo di livello Transport, scelta comune a molte applicazioni real time. Infatti, la perdita di alcuni pacchetti durante una conversazione audio non ne compromette la comprensione da parte degli interlocutori.

Nel caso di trasporto di dati audio e video, si usa il protocollo **RTP** (Real Time Transport Protocol), descritto nella Lezione precedente.

All'inizio VoIP fu presentata come un'applicazione che consentiva di effettuare telefonate gratuitamente, ma il suo successo è dovuto anche ad altre caratteristiche che migliorano il servizio rispetto alla telefonia tradizionale:

- **realizzazione più semplice**: molte funzioni che prima erano distribuite in vari punti di accesso alla rete, ora sono centralizzate, ne consegue una più veloce installazione e riduzione delle attività di amministrazione;
- **rete di trasporto IP**: non è più necessario riservare linee dedicate per il traffico telefonico, la voce viene trasportata nelle reti IP come gli altri dati, è però necessaria una configurazione iniziale;
- **riduzione dei costi**: è significativa per tutti, ma soprattutto per le aziende dove si fanno quotidianamente molte telefonate e spesso internazionali;
- **offerta di servizi a valore aggiunto**: l'infrastruttura VoIP si presta bene a realizzare vari servizi per gli utenti, quali trasferimento di chiamata, richiamo automatico, messaggistica e video-conversazione;
- **anytime-anywhere**: l'utente può telefonare in qualunque momento e ovunque si trovi effettuando un accesso a Internet e usando un account registrato (**FIGURA 12**).

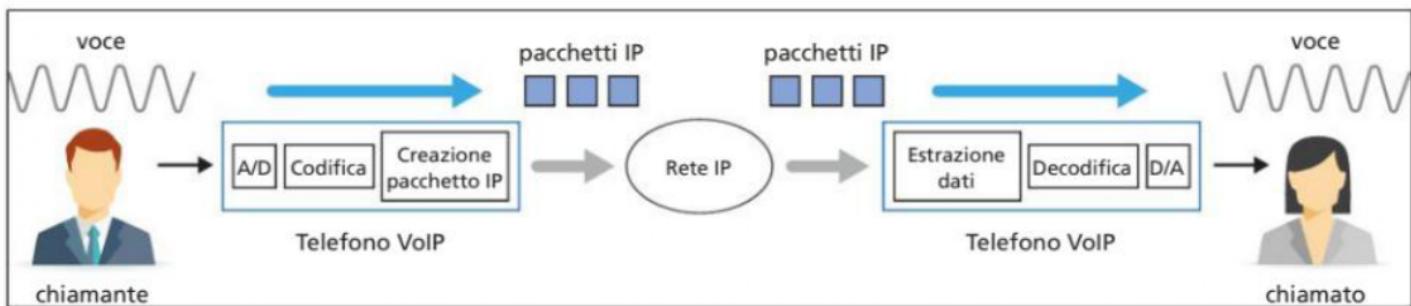


**FIGURA 12** La comunicazione tra dispositivi mobili e fissi

## 8.2 Il telefono IP

**FIGURA 13** Come funziona un telefono VoIP

La telefonia su IP richiede l'impiego di tecniche di digitalizzazione della voce, come mostrato nella **FIGURA 13**.



Il segnale audio che si ottiene dal microfono viene campionato con una velocità elevata e quantizzato per trasformarlo da segnale analogico a digitale (conversione A/D). È successivamente codificato per adattarlo al tipo di canale trasmissivo (riduzione del bit rate, compressione, regolazione della **#dinamica**).

Dopo la codifica la voce viene inserita nei pacchetti IP da trasmettere sulla rete. In genere, un pacchetto contiene 20 o 30 ms di audio. Questo processo avviene nel dispositivo chiamato **telefono VoIP** utilizzato dalla persona che inizia la chiamata. Quando il pacchetto che contiene la voce arriva a destinazione, viene consegnato al telefono VoIP della persona chiamata e qui subisce il processo inverso di estrazione dei dati dal pacchetto IP, di decodifica e conversione in segnale analogico (conversione D/A).

L'evoluzione delle tecniche di codifica della voce e, più in generale, dei segnali audio, hanno portato alla specifica di vari tipi di codici (abbreviati in **#codec**). Attualmente i codec impiegati nel VoIP comprimono l'audio di un fattore 8 o 10 rispetto alla telefonia tradizionale.

In fase di codifica, i codec applicano la tecnica di **soppressione dei silenzi**, interrompendo la trasmissione durante i periodi di inattività. Infatti, durante una chiamata la percentuale di silenzi è piuttosto elevata e queste tecniche consentono un notevole risparmio di bandwidth.

Nella fase di decodifica, i codec possono compensare eventuali **pacchetti persi** durante la trasmissione così da rendere l'audio accettabile dall'orecchio umano.

Una tecnica utilizzata è la **FEC (Forward Error Correction)** che consiste nell'inserire informazioni ridondanti nei pacchetti unitamente ai dati originali, così da poterle utilizzare in fase di ricezione per recuperare le informazioni contenute nei pacchetti persi. Questa tecnica è valida, però, se la percentuale di pacchetti persi è inferiore al 20%. Un'altra tecnica utilizzata dal ricevitore è la **ripetizione dei pacchetti**: il contenuto dei pacchetti persi si sostituisce con quello dei pacchetti che li hanno appena preceduti e sono arrivati integri a destinazione.

Spesso, però, quando il numero di pacchetti persi è basso, si sfrutta la capacità di recupero dell'orecchio umano, che è in grado di tollerare bene fino al 5% di pacchetti persi. Sulla qualità della comunicazione incide maggiormente la variabilità nel tempo di arrivo dei pacchetti (**jitter**) che può essere contenuta con l'impiego di buffer, lato ricezione, prestando però attenzione affinché non introducano un ritardo nella consegna. Per questo motivo si utilizzano dei buffer dinamici, in grado di cambiare dimensione in funzione dello stato della rete.

### #techwords

#### Dinamica

Nel linguaggio musicale la dinamica si occupa dell'**intensità dei suoni** e della loro gradazione da adottare nell'esecuzione di una composizione.

### #techwords

#### Codec

Software o hardware che si occupa della codifica digitale, e decodifica, di un segnale audio o video, utilizzando tecniche di compressione dei dati. Nei sistemi di telecomunicazioni, il codec attua anche la codifica di canale sui dati da trasmettere. Esempi di codec audio sono wav e mp3. Codec spesso utilizzati in VoIP sono G.711 e G.729.

## HARDPHONE E SOFTPHONE

Un telefono VoIP è utilizzato tipicamente nelle reti telefoniche aziendali con centralini PBX. Viene chiamato **hardphone**, per distinguerlo dalle applicazioni software installate sui dispositivi che svolgono funzioni analoghe e vengono chiamate **softphone**. Skype è un esempio di applicazione softphone.

Un hardphone VoIP dispone di un display, anche touch, e pulsanti con cui interagire con le varie funzionalità del telefono.

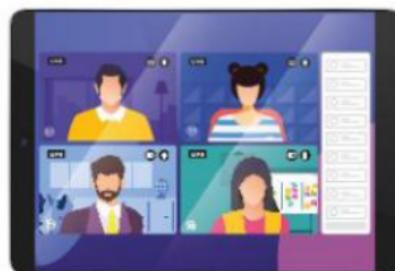
A differenza dei tradizionali telefoni, essi sono dei computer con un Sistema Operativo che permette loro di svolgere compiti avanzati come, per esempio, una videochiamata grazie alla videocamera integrata.

Esistono anche hardphone con un aspetto diverso da quello tipico del telefono, come quelli utilizzati nelle audio conferenze (**FIGURA 14**).

**FIGURA 14** Telefono VoIP per audio conferenze



**FIGURA 15**  
Esempi di softphone



Un softphone è un'applicazione software che si installa su computer, tablet o smartphone fornendo a questi dispositivi le funzionalità tipiche di un telefono VoIP (**FIGURA 15**). L'interazione con l'utente avviene tramite un'interfaccia grafica.

La prima evidente differenza tra hardphone e softphone è nella mobilità: il primo è un telefono da scrivania, può essere cordless, ma ha una mobilità limitata, il secondo si installa su un dispositivo mobile e segue la persona ovunque si muova.

Un hardphone offre una qualità migliore delle chiamate rispetto a un softphone, in quanto dispone di hardware e software dedicato e quindi non subisce le interferenze delle altre applicazioni che lavorano in background sul computer. Inoltre il funzionamento del softphone è soggetto allo stato, acceso o spento, del dispositivo che lo ospita. Per contro un hardphone è molto più costoso e meno personalizzabile di un softphone. Infatti, molti softphone offrono le funzionalità di base gratuitamente, richiedono di disporre di hardware come microfono, altoparlanti e videocamera che spesso sono già integrati sui laptop più moderni, oltre che, ovviamente, su tablet e smartphone.

## 8.3 I centralini telefonici su IP

Le aziende di una certa dimensione installano un centralino telefonico per lo smistamento delle chiamate negli uffici.

La rete telefonica privata di un'azienda viene chiamata **PBX (Private Branch Exchange)**.

**PBX-IP** è il sistema basato su IP che supera le limitazioni in numero di linee telefoniche e di dispositivi telefonici interni che erano presenti nei PBX tradizionali.

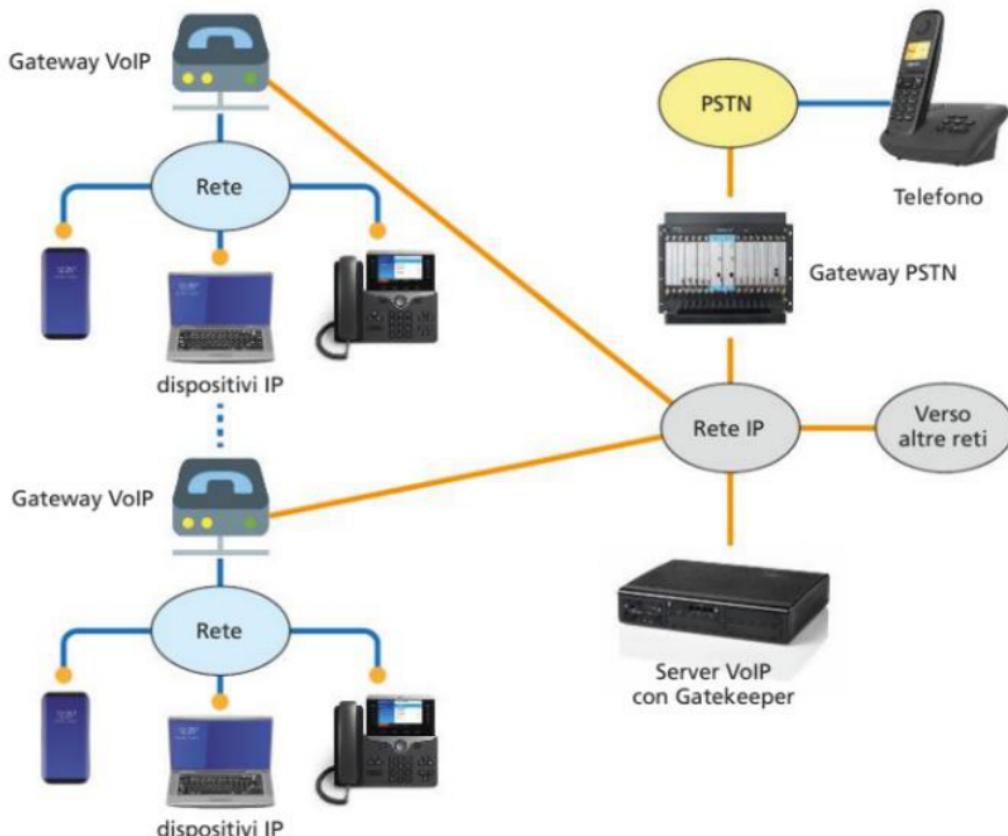
## #prendinota

**Asterisk** è uno dei più diffusi software PBX-IP, è gratuito e si installa su server Linux. Supporta i protocolli standard SIP, MGCP e H.323, lavorando come gateway tra telefoni IP e la rete PSTN. Utilizza un protocollo interno (IAX, Inter-Asterisk eXchange) per la comunicazione tra i PBX. Asterisk si ritrova come componente centrale in molti prodotti commerciali e progetti open source.



La **FIGURA 16** mostra gli elementi principali che costituiscono una rete VoIP con PBX-IP:

- **dispositivi IP:** sono quelli coinvolti nella chiamata VoIP e devono essere quindi dei dispositivi IP che dialogano con i protocolli VoIP, hardphone e softphone;
- **VoIP server:** è l'elemento centrale che stabilisce una comunicazione tra chiamante e chiamato, la gestisce e la termina; implementa i protocolli di segnalazione (spiegati più avanti) e assicura il routing corretto dei pacchetti IP che trasportano la voce;
- **Gateway:** assicura la comunicazione tra dispositivi VoIP presenti su reti con caratteristiche diverse (Gateway VoIP) e tra dispositivi VoIP e telefoni connessi alla rete tradizionale PSTN, Public Switched Telephone Network (Gateway PSTN);
- **Gatekeeper:** è un software gestionale che può essere installato sul server o su un hardware a parte e che mantiene i dati per la tariffazione insieme a quelli delle varie chiamate effettuate.



**FIGURA 16** Un esempio di sistema PBX-IP

## I CENTRALINI VIRTUALI O IN CLOUD

Il centralino virtuale è un sistema PBX installato in remoto e fornito come servizio attraverso la rete.

Le aziende che lo utilizzano possono usufruire dell'eliminazione dei costi di installazione, operativi e di gestione del centralino in quanto è il provider VoIP a farsene carico. È una soluzione altamente scalabile, infatti è possibile aggiungere o rimuovere numeri interni o linee telefoniche in base alle necessità del momento e il personale dell'azienda può lavorare anche all'esterno utilizzando lo stesso numero interno. Un centralino in cloud funziona allo stesso modo di un PBX installato in un'azienda, ma è ospitato su un server cloud.

## 8.4 Il protocollo SIP

Nei primi anni in cui ha iniziato a svilupparsi la telefonia su Internet, si utilizzavano spesso tecnologie proprietarie, ma anche successivamente, quando furono emessi i primi standard definiti dagli enti di standardizzazione, le modalità di realizzazione e i protocolli specificati erano svariati, sia proprietari sia standard sviluppati in ambito ITU-T.

Tutti, però, mantenevano la distinzione, derivata dalle reti telefoniche, tra:

- **protocollo di segnalazione:** si occupa delle fasi di instaurazione e disconnessione della chiamata e dei servizi aggiuntivi, come quelli di #directory per la gestione dei contatti;
- **protocollo di trasporto:** quando la chiamata è stabilita, gestisce il trasferimento dei pacchetti che trasportano la voce tra i telefoni VoIP.

### #techwords

#### Directory

È un elenco di nomi, nella telefonia è l'elenco telefonico (telephone directory).

Come molti altri servizi che abbiamo visto in questa Unità, anche VoIP è implementato secondo il paradigma Client-Server, quindi il protocollo di segnalazione regola la comunicazione tra client VoIP e server VoIP: il client invia al server la richiesta di chiamare un certo numero e il server lo contatterà per instaurare la comunicazione tra i due telefoni.

Una volta instaurata la comunicazione, si userà un protocollo di trasporto per la trasmissione dei pacchetti che contengono la voce.

Verso la fine degli anni Novanta si diffuse il protocollo di segnalazione **H.323** definito in ambito ITU-T. H.323 era una suite di protocolli e utilizzava molti principi della telefonia tradizionale. La sua complessità di realizzazione e la diffusione sempre maggiore del VoIP generò l'esigenza di protocolli più snelli e semplici da implementare. Nacque un gruppo di lavoro in ambito IETF per la definizione di un nuovo protocollo di segnalazione per il VoIP che portò alla specifica di **SIP (Session Initiation Protocol)**.

IETF definì anche un nuovo protocollo di trasporto, **RTP**, che si inserisce tra il protocollo di livello Transport UDP e i protocolli VoIP di livello applicativo, come visto nella precedente Lezione.

Le specifiche di SIP sono state pubblicate nell'**RFC 3261** nel 2002; pur non essendo mai stato reso obsoleto, questo RFC è stato più volte aggiornato negli anni successivi, con oltre venti nuovi RFC.

Il protocollo SIP offre meccanismi per:

- la notifica della chiamata al chiamato, equivalente allo squillo del telefono nella telefonia tradizionale;
- la negoziazione tra il chiamante e il chiamato sui dispositivi da usare e sulla codifica;
- la chiusura della chiamata, equivalente al riaggancio del telefono nella telefonia tradizionale;
- permettere al chiamante di conoscere l'indirizzo del chiamato, infatti potrebbe aver ottenuto l'indirizzo IP da un DHCP, quindi non fisso;
- la gestione della chiamata, per esempio è possibile allargare la conversazione ad altri interlocutori, cambiare codifica durante la chiamata, ecc.

Il protocollo SIP può essere usato insieme a RTP, ma non è un obbligo, infatti le specifiche prevedono che possa lavorare anche con altri protocolli e servizi.

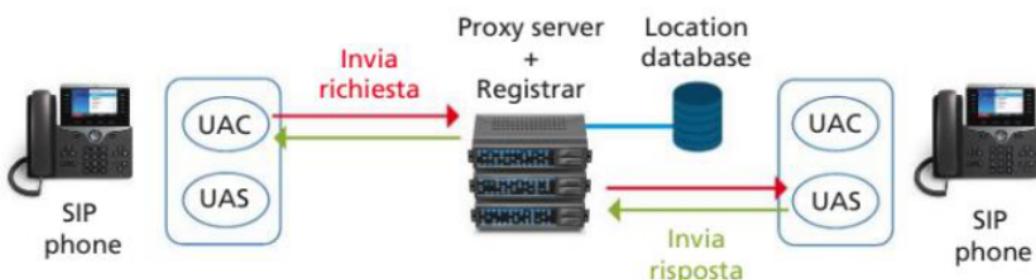
Su ogni dispositivo SIP è installata un'applicazione di tipo Client-Server che si chiama **UA (User Agent)**. Un UA può operare come client, **UAC** (User Agent Client), e inviare una richiesta al server, oppure come server, **UAS** (User Agent Server), che elabora la richiesta del client e invia la risposta.

Un UA può svolgere entrambi i ruoli, quindi essere sia un client che inizia una conversazione inviando a un server un messaggio di request, sia un server che risponde alla richiesta di un client.

Si possono avere diversi tipi di UAS:

- **Proxy server**: riceve le richieste SIP da un client e le inoltra in rete verso gli UAS, riceve le risposte e le inoltra verso gli UAC;
- **Redirect server**: invece di inoltrare la chiamata al server di destinazione, restituisce al client il suo indirizzo, così questi potrà contattare il nuovo server direttamente;
- **Registrar server**: gli utenti SIP devono registrare la loro posizione in un Registrar server, che di solito si trova all'interno di un Proxy o Redirect server e memorizza le informazioni ricevute in un Location database.

**FIGURA 17** Componenti Client-Server di SIP con Proxy server



La **FIGURA 17** mostra un esempio di comunicazione tra due telefoni SIP tramite un Proxy server che svolge anche le funzioni di Registrar server.

### ■ GLI INDIRIZZI SIP

Nel protocollo SIP gli indirizzi sono definiti nel formato URI (Uniform Resource Identifier). Esempi di indirizzi SIP:

- `sip:bianchi@azienda.com`
- `sip:bianchi@178.25.49.161`
- `sips:+00393351069482@azienda.com:5062`

dove "sips" indica che il trasporto dei dati avviene con TLS.

Gli indirizzi SIP possono essere inseriti in una pagina web come URL: quando il visitatore clicca sull'indirizzo, l'applicazione VoIP del suo dispositivo si attiva e invia la richiesta di chiamata.

### ■ I COMANDI DI SIP

Il protocollo SIP è stato progettato in origine per essere molto semplice, con un numero limitato di comandi:

- INVITE: richiesta per stabilire una connessione, *si invita* un utente a ricevere una chiamata;
- ACK: conferma della ricezione della richiesta INVITE;

- **BYE:** termina la connessione tra gli utenti;
- **CANCEL:** cancella ogni azione in sospeso, generalmente una richiesta INVITE;
- **OPTIONS:** interroga il server per avere un elenco delle sue funzionalità e stato;
- **REGISTER:** comunica a un SIP registrar uno o più indirizzi di contatto dell'utente (UA).

I numerosi RFC di aggiornamento hanno introdotto nuovi comandi: SUBSCRIBE, NOTIFY, INFO, UPDATE, ecc.

Per i messaggi di **response** (le risposte alle richieste) SIP usa dei codici simili a quelli usati da HTTP, per esempio:

- 100 (Trying);
- 200 (OK);
- 404 (Not found);
- 500 (Server internal failure).

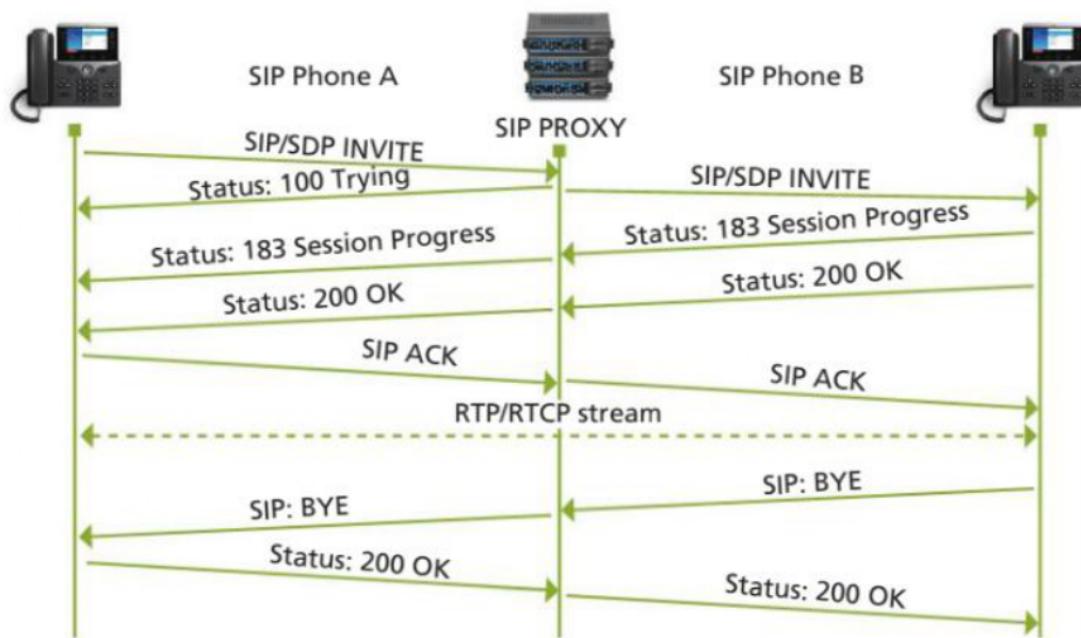
Nella **FIGURA 18** è mostrata una chiamata tra due telefoni SIP, il successivo scambio di informazioni in streaming tramite i protocolli RTP e RTCP e la chiusura della connessione.

### #prendinota

#### WhatsApp utilizza SIP?

La famosa piattaforma di messaggistica offre servizi di telefonia su IP, infatti permette di effettuare audio e video chiamate dall'applicazione client installata su dispositivi mobili o sul computer.

WhatsApp non utilizza SIP, bensì un protocollo di segnalazione proprietario.



**FIGURA 18** Esempio di colloquio tra telefoni SIP tramite Proxy server

### FISSA LE CONOSCENZE

- Descrivi le principali caratteristiche del VoIP.
- Qual è il ruolo dei codec nei servizi VoIP?
- Che cosa sono i PBX-IP?
- Quale protocollo è stato standardizzato da IETF per la telefonia su IP?
- Spiega il ruolo client e server dello User Agent presente sui dispositivi SIP.