

# 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.

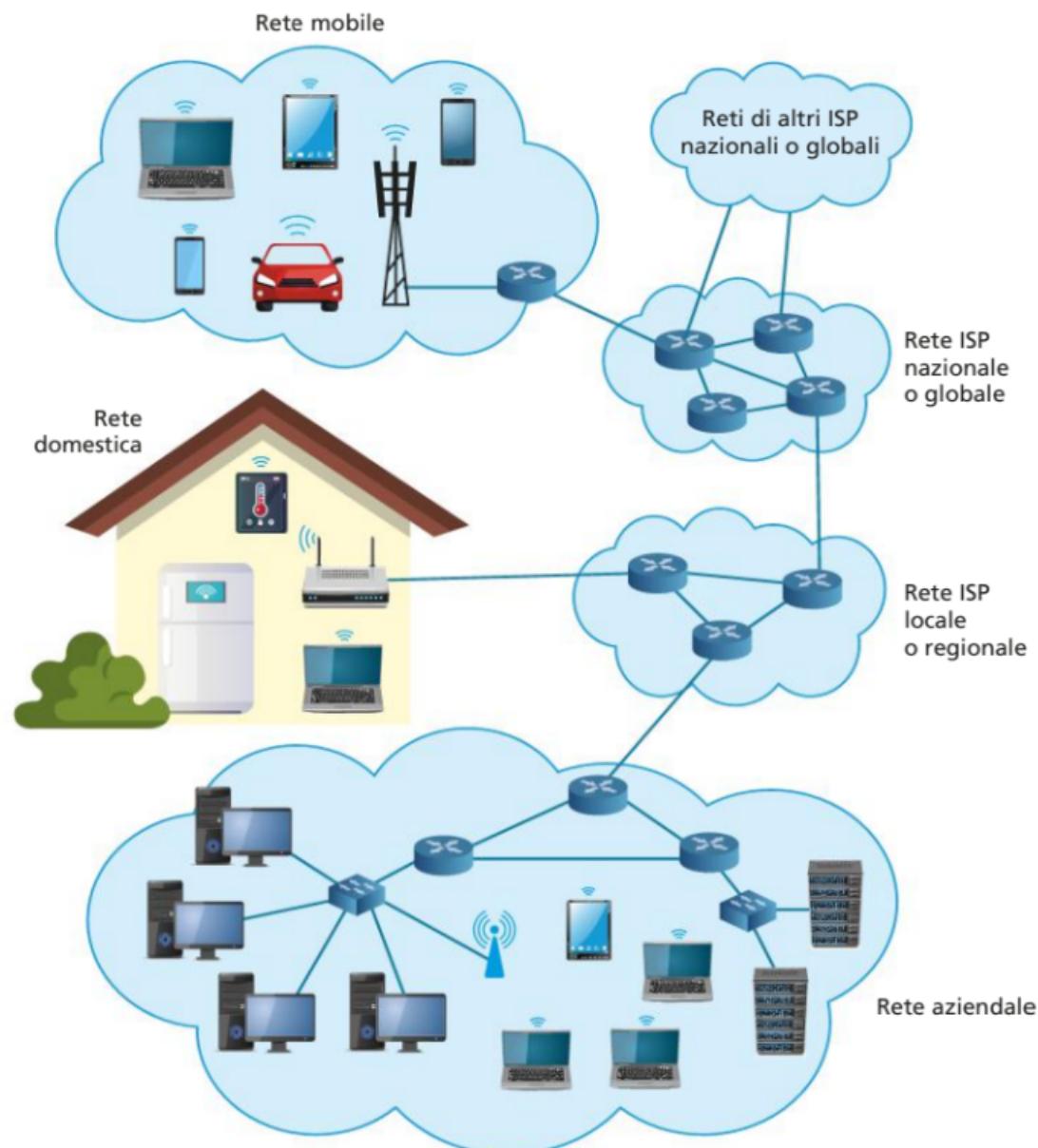
# 1 UNA VISIONE D'INSIEME DELLA RETE INTERNET

## 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.

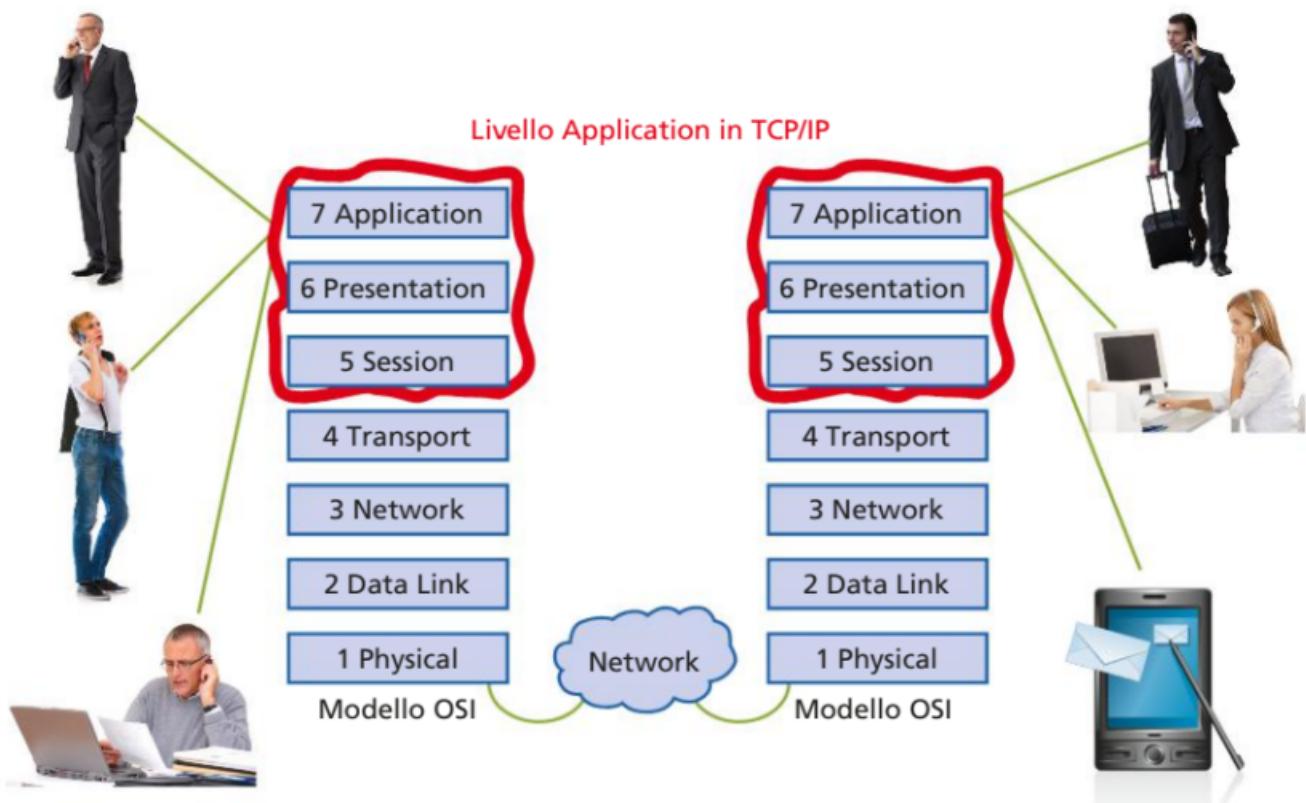
## 2 IL LIVELLO APPLICATION E I SUOI PROTOCOLLI

### 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 ogniqualvolta 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.

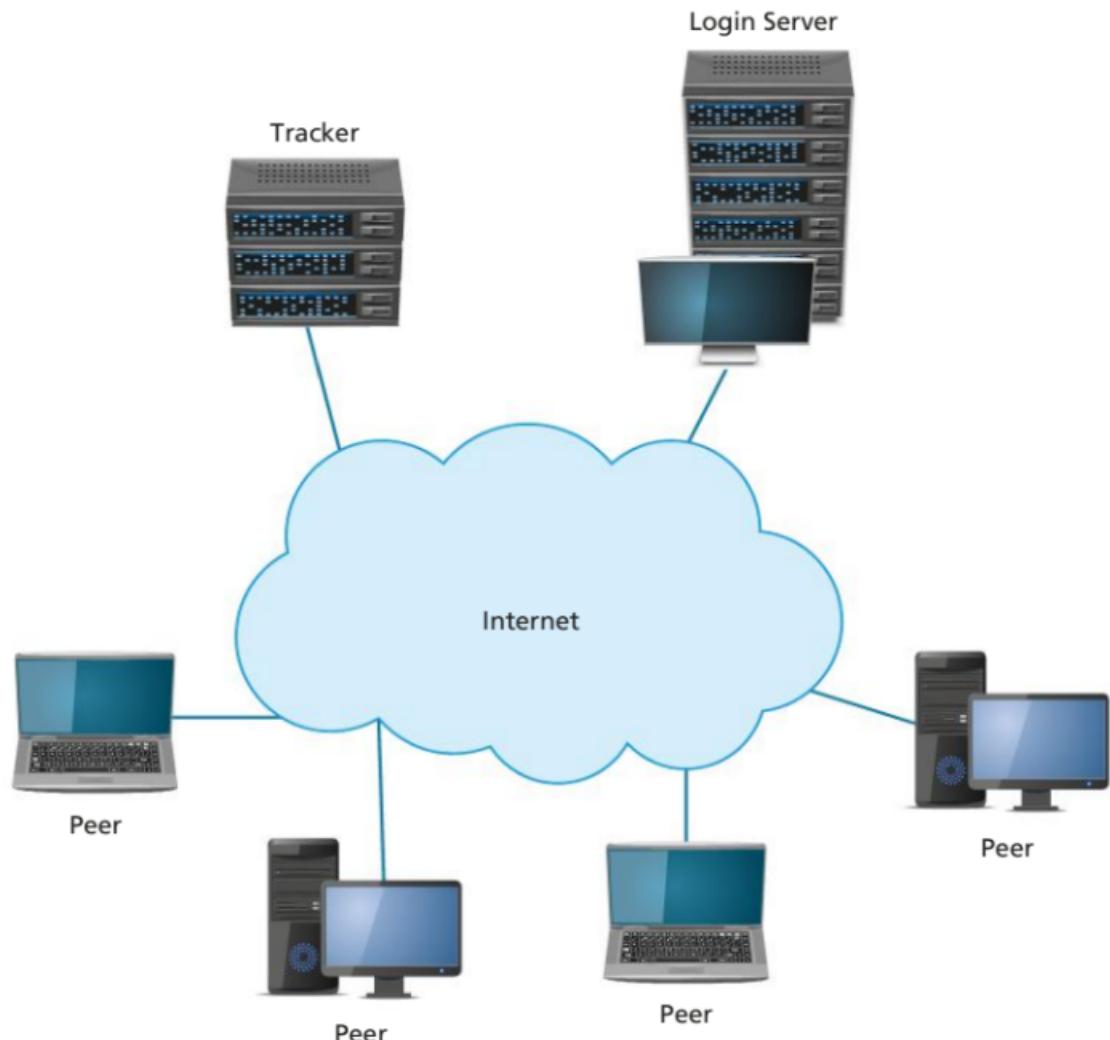
La **FIGURA 3** mostra come è strutturata una generica applicazione P2P, sono presenti:

- 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

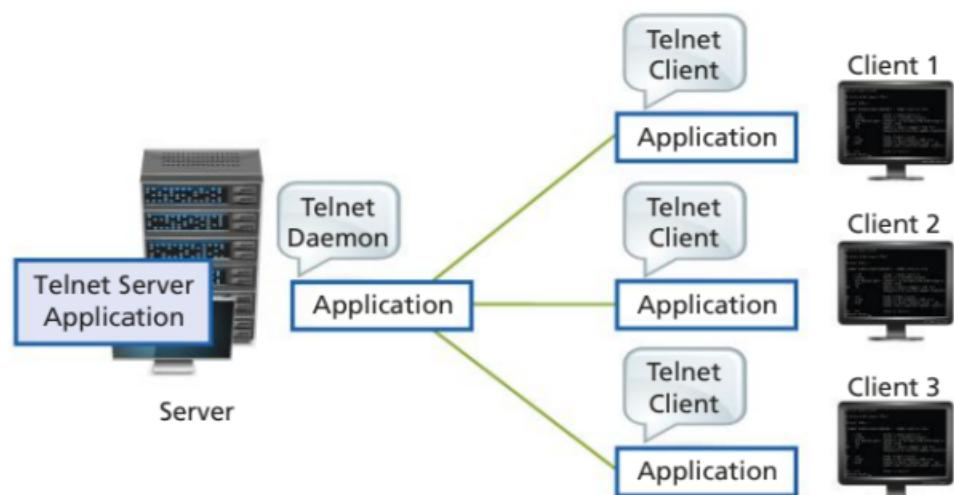
Telnet è un protocollo Client-Server: la componente client di Telnet è un'applicazione di emulazione di terminale (testuale) che permette agli utenti di un sistema di accedere ad applicazioni che si trovano su host remoti, come se fossero direttamente connessi a tali sistemi. L'host di destinazione deve 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



esercizio

#### → 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
```

## 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

Obsoletes: NIC 18639

ISI

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 creati 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

**Request for Comments: 959**

Obsoletes RFC: 765 (IEN 149)

J. Postel

J. Reynolds

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

**Request For Comments: 1350**

STD: 33

Obsoletes: RFC 783

K. Sollins

MIT

July 1992

#### #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.

#### 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.

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

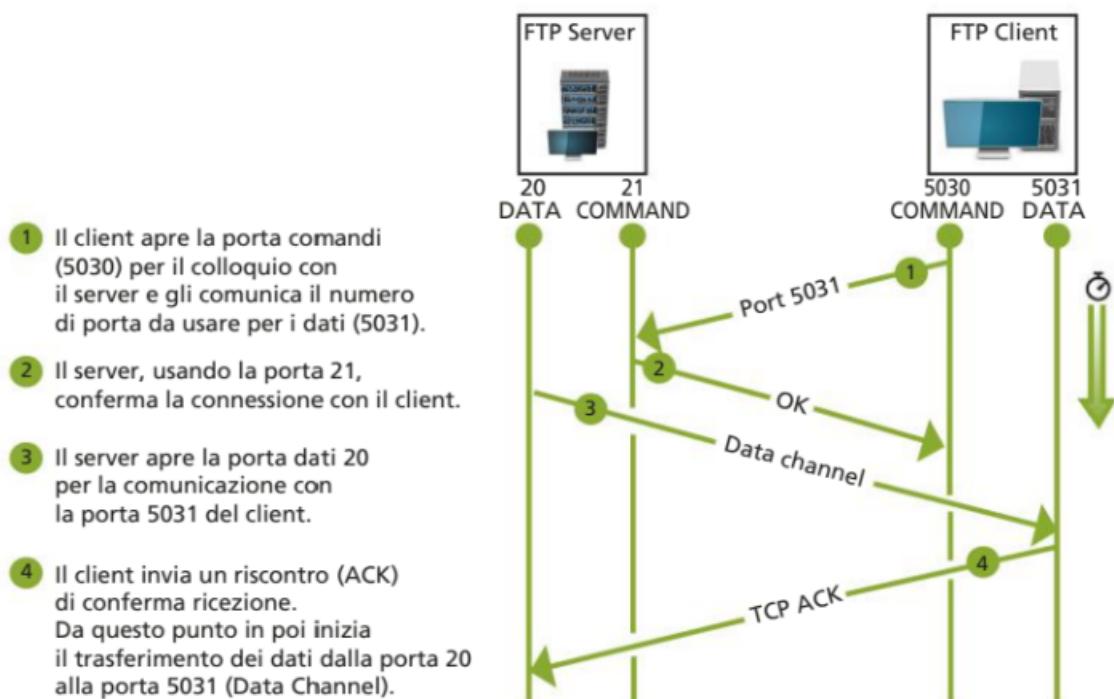
A differenza di altri protocolli, FTP utilizza **due canali** per la comunicazione tra client e server:

- 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.



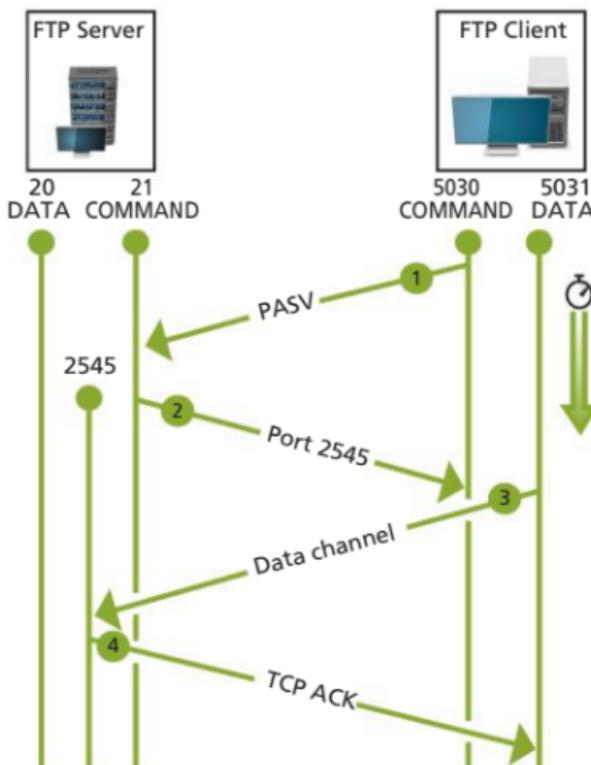
**FIGURA 5** Modalità attiva di FTP

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:

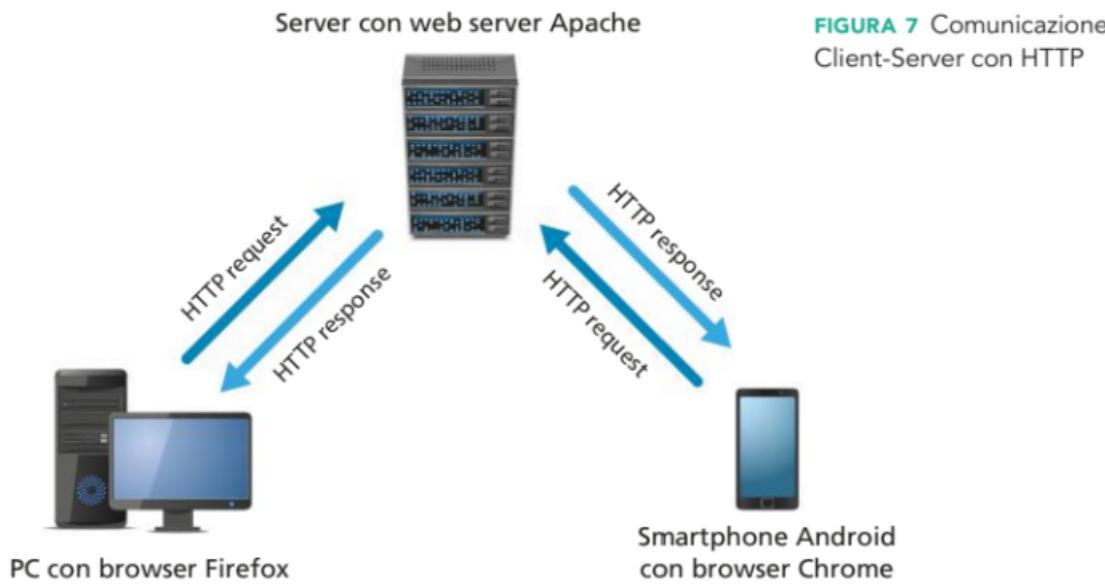
- il **browser** è il programma client dell'applicazione, esempi sono Edge, Firefox, Chrome, Opera. Il browser svolge due funzioni fondamentali: inoltre la richiesta di una pagina web al server (HTTP request) e presenta i dati ricevuti (HTTP response) all'utente. Le pagine web sono create con un linguaggio 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.



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.



## 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 inoltrare 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, stabilità dall'autore di un testo, per passare a una lettura che vede il lettore come protagonista, non più come fruitore 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?

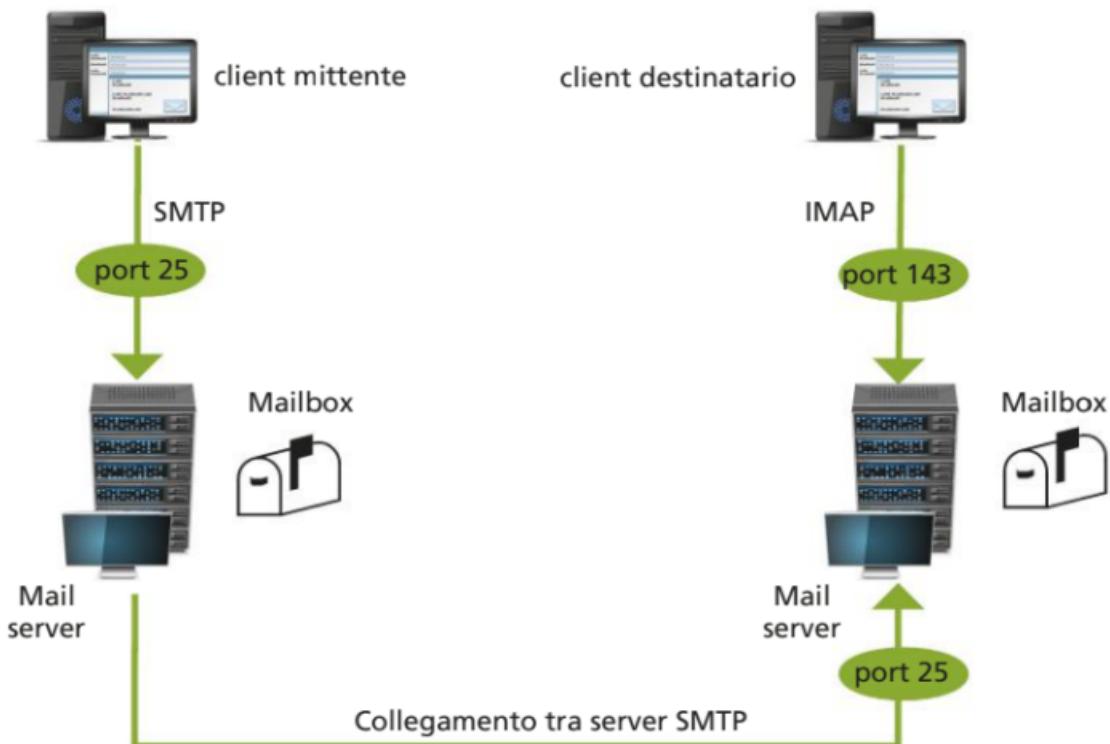
## 6 SMTP, POP E IMAP: I PROTOCOLLI PER LA POSTA ELETTRONICA

### 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

**Request for Comments: 5321**

Obsoletes: 2821

Updates: 1123

Category: Standards Track

J. Klensin

October 2008

### 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.