

# Cosa succede quando si clicca un un link

Trasferire informazioni su Internet

## **Autori**

[Silvio Peroni](mailto:silvio.peroni@unibo.it) – [silvio.peroni@unibo.it](mailto:silvio.peroni@unibo.it)

Dipartimento di Filologia Classica e Italianistica, Università di Bologna, Bologna, Italia

[Aldo Gangemi](mailto:aldo.gangemi@unibo.it) – [aldo.gangemi@unibo.it](mailto:aldo.gangemi@unibo.it)

Dipartimento di Filologia Classica e Italianistica, Università di Bologna, Bologna, Italia

## **Avviso sul copyright**

Questo lavoro è rilasciato con licenza [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/). Tu sei libero di condividere (riprodurre, distribuire, comunicare al pubblico, esporre in pubblico, rappresentare, eseguire e recitare questo materiale con qualsiasi mezzo e formato) e modificare (remixare, trasformare il materiale e basarti su di esso per le tue opere per qualsiasi fine, anche commerciale) questo lavoro alle seguenti condizioni: attribuzione, ovvero devi riconoscere una menzione di paternità adeguata, fornire un link alla licenza e indicare se sono state effettuate delle modifiche. Puoi fare ciò in qualsiasi maniera ragionevole possibile, ma non con modalità tali da suggerire che il licenziante avalli te o il tuo utilizzo del materiale. Il licenziante non può revocare questi diritti fintanto che tu rispetti i termini della licenza.

## Sommario

In questo capitolo verrà introdotto il secondo dei protocolli di comunicazione principali di Internet, TCP, e verrà presentata la struttura dei pacchetti TCP per il recapito di informazioni con connessione da un mittente a un destinatario.

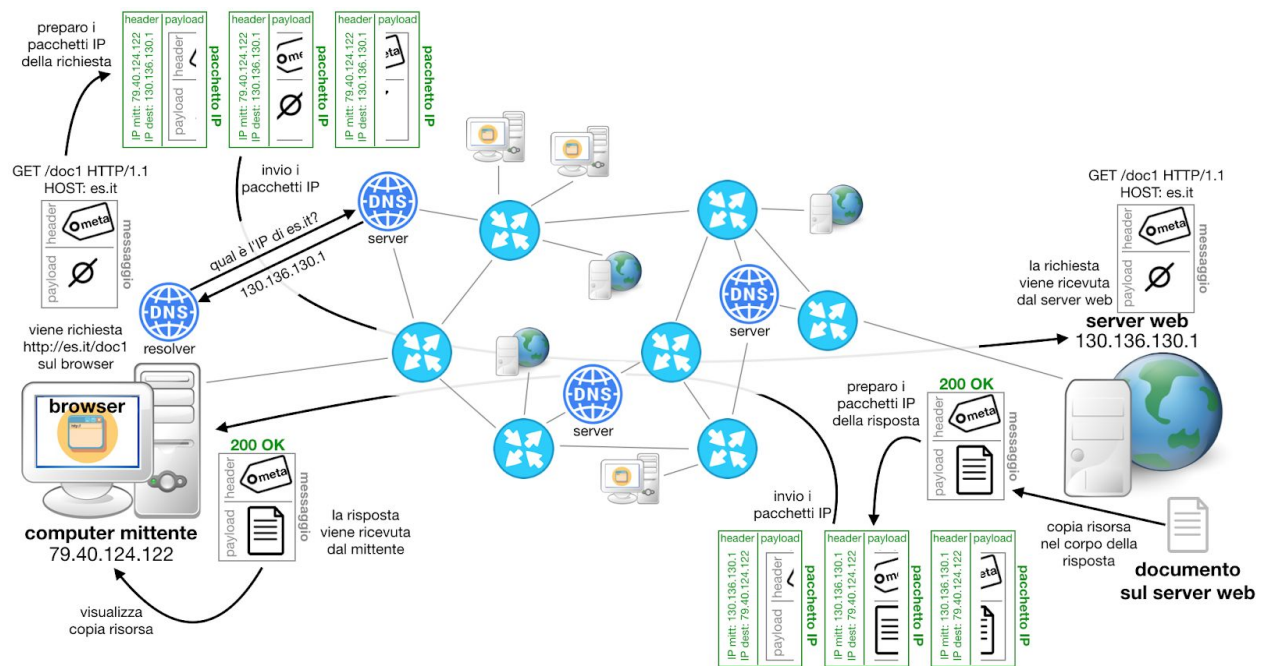
## Introduzione

Nei precedenti tre capitoli, in questo, e nei successivi affronteremo tutte le tecnologie alla base di Internet e del World Wide Web. Tuttavia, invece di affrontare questi temi in modo tradizionale, ovvero partendo dalla descrizione architettonica di Internet per poi presentare il Web, abbiamo voluto introdurre questi argomenti seguendo la traccia fornita da Tim-Berners Lee (l'inventore del Web) in un noto [documento esplicativo](#) a disposizione sul suo sito Web.

In particolare, l'obiettivo è quello di rispondere ad una semplice, seppur significativa, domanda:

- Cosa succede quando si clicca su un collegamento ipertestuale (o link) di una pagina Web?

In [Figura 1](#) sono riassunte le tecnologie che abbiamo introdotto nei tre capitoli precedenti e che ci hanno permesso di iniziare ad introdurre gli elementi principali per rispondere alla domanda appena presentata. In particolare, a seguito di un click su un link di una pagina web visualizzata sul browser di un computer, viene composto un messaggio HTTP di richiesta di una specifica pagina web identificata da un URL. Una volta recuperato l'indirizzo IP dal nome del server web che ha la pagina a disposizione usando il DNS, il messaggio HTTP viene incapsulato da uno o più pacchetti IP che vengono poi inviati dal computer mittente al router più vicino – ad esempio il router di casa. Da lì, i pacchetti IP vengono instradati nella rete finché non raggiungono il server web destinatario, che riceve il messaggio, e impacchetta una copia del documento richiesto in un nuovo messaggio HTTP di risposta. A sua volta, questo messaggio viene incapsulato in uno o più pacchetti IP inviati dal server web al router più vicino, e poi instradati fino al mittente. Una volta ricevuto, il messaggio di risposta viene passato al browser che visualizza a video il contenuto del *payload* del messaggio HTTP.



**Figura 1.** Riassunto delle tecnologie che abbiamo trattato nei capitoli precedenti – dove i cerchi azzurri con le frecce bianche sono router.

In questo capitolo introdurremo l'ultimo dei protocolli di base di Internet, il TCP, e riassumeremo nuovamente tutti i passaggi che avvengono dal momento in cui si clicca su un link di una pagina web al momento in cui la pagina richiesta viene visualizzata sul browser.

## Byte e bit: unità di informazione

Nello scorso capitolo abbiamo accennato al fatto che un particolare messaggio debba essere spezzato in uno o più pacchetti IP prima che questi vengano instradati in rete. Questo è dovuto a due diversi fattori. Il primo è il limite dato dalla massima quantità di dati che ogni pacchetto IP può trasportare (che dipende dalla versione considerata del protocollo, IPv4 o IPv6). Tuttavia, in aggiunta a questo limite, c'è anche il limite imposto dalla rete a cui si instradano i pacchetti, ovvero il suo [Maximum Transmission Unit \(MTU\)](#). Quindi, anche se tutto il messaggio HTTP può, in linea di principio, essere incapsulato dentro un pacchetto IP, è possibile che debba comunque essere spezzato in più pacchetti IP a causa del limite MTU imposto dalla rete. Questa frammentazione può avvenire sia nello strato di rete in alcuni casi e solo se si utilizza il protocollo IPv4 per l'instradamento, o nello strato trasporto, che sarà oggetto di questo capitolo.

Questi limiti, sia quello del protocollo IP sia quello dettato dall'MTU, sono espressi in [byte](#). Un byte è una delle più diffuse [unità di informazione](#). Storicamente corrisponde al numero di [bit](#) necessari per codificare un carattere sul computer [[Bemer, 1959](#)], e per questo è di solito l'unità minima di informazione occupabile su un computer elettronico. In particolare, un byte corrisponde a otto bit, ove con *bit* (una contrattura di *binary digit*) si intende l'unità minima di informazione che si può scambiare in una comunicazione. Un bit può assumere soltanto uno di due valori: 0 o 1.

Il concetto di bit è stato indirettamente usato da diversi studiosi del passato, per esempio da Charles Babbage con la sua macchina analitica (descritta nei capitoli precedenti) che proponeva l'uso di schede perforate, dove la presenza o assenza di un buco in una certa posizione poteva essere interpretata rispettivamente come l'assegnamento di un 1 o di uno 0. Tuttavia, la prima persona a usare la parola *bit* fu [Claude Shannon](#) (mostrato in [Figura 2](#)) nel 1948 in un articolo che ha segnato la nascita della [teoria dell'informazione](#) [[Shannon, 1948](#)]. In breve, la teoria dell'informazione riguarda uno specifico campo di ricerca che studia come quantificare, memorizzare, e scambiare informazione, che ha tuttora svariate applicazioni pratiche, oltre che aver caratterizzato e veicolato l'invenzione di diverse tecnologie del passato, incluso lo sviluppo di Internet. Ad esempio, evincere la [capacità massima di un canale](#) per trasmettere informazione in modo affidabile, è stato uno degli studi della teoria dell'informazione che caratterizza qualsiasi comunicazione tra una sorgente e un destinatario attraverso un canale di comunicazione, e ha particolare rilevanza per stabilire, per esempio, la massima unità di informazione per pacchetto (MTU) in una particolare rete, accennata in precedenza.

L'idea di base è che ogni carattere, numero, programma, applicazione all'interno di un computer, così come un qualunque messaggio da scambiare in Internet, è di fatto codificato come una sequenza di bit che, in qualche modo, vengono poi riconvertiti con la sequenza di caratteri corretta dalla particolare applicazione che riceve ed interpreta quell'informazione. Per esempio, nella codifica binaria tradizionale dei numeri interi, il numero 0 è rappresentato dalla

sequenza "0", il numero 1 da "1", il numero 2 da "10", il numero 3 da "11", il numero 4 da "100", e così via.



**Figura 2.** Foto di Claude Shannon.

Fotografia scattata da Konrad Jacobs, sorgente: [https://en.wikipedia.org/wiki/File:ClaudeShannon\\_MFO3807.jpg](https://en.wikipedia.org/wiki/File:ClaudeShannon_MFO3807.jpg).

<b>indirizzo IP</b>	208	.	80	.	154	.	224
<b>bit</b>	1 1 0 1 0 0 0 0		0 1 0 1 0 0 0 0		1 0 0 1 1 0 1 0		1 1 1 0 0 0 0 0
<b>valore</b>	128 64 32 16 8 4 2 1		128 64 32 16 8 4 2 1		128 64 32 16 8 4 2 1		128 64 32 16 8 4 2 1
<b>somma</b>	128+64 +16		64 +16		128 +16 +8 +2		128+64+32

**Figura 3.** Un indirizzo IP (versione IPv4) e come viene codificato in 4 byte, un byte per ogni numero, di 8 bit l'uno.

In modo indiretto, abbiamo già in qualche modo usato il concetto di byte nei capitoli precedenti. Per esempio, tutti gli indirizzi IP (versione IPv4) che abbiamo introdotto nei capitoli precedenti sono definiti mediante l'uso di 4 byte, uno per ogni numero. Nella codifica binaria usata, ogni byte, ovvero 8 bit, permette di definire un numero intero da 0 a 255, che sono esattamente i valori che ogni numero dell'indirizzo IP può assumere. Il procedimento di [codifica](#) binaria, ovvero la rappresentazione di un certo elemento (in questo caso ogni numero di un indirizzo IP) in formato binario, è mostrato in [Figura 3](#) e funziona come segue. Ognuno degli otto bit di un numero facente parte dell'indirizzo IP ha assegnato uno specifico valore. In particolare, il valore 1 è associato al bit più a destra, mentre a quello successivo verso sinistra è assegnato un

valore doppio rispetto al precedente (in questo caso 2), e così via fino ad assegnare un valore numerico ad ognuno degli 8 bit, fino a 128. Questi valori permettono di poter specificare un qualsiasi numero compreso tra 0 e 255, semplicemente sommandone alcuni fra loro. Infatti, sommando i soli valori di tutti i bit assegnati a 1 di un certo byte di un indirizzo IP si riesce a risalire al numero intero dell'indirizzo – ad esempio, il byte 11010000 fa riferimento al numero ottenuto sommando di 128, 64, e 16 (ovvero i valori assegnati ai bit impostati a 1), che è uguale a 208, il primo numero dell'indirizzo IP mostrato in [Figura 3](#).

Ovviamente, un qualunque messaggio e/o pacchetto spedito attraverso Internet è di fatto codificato usando una sequenza di bit. Una parte di questi bit corrispondono a informazioni relative all'*header* del pacchetto, mentre altre sono relative al *payload*.

## Che cos'è il Transmission Control Protocol (TCP)

Nello scorso capitolo abbiamo visto come l'instradamento basato sul protocollo IP non garantisce che i pacchetti arrivino nell'ordine corretto in cui il loro payload deve essere poi ricomposto per formare il messaggio originale. Come anticipato, sia il riordino dei pacchetti sia, in talune situazioni, lo spezzare il messaggio originale in più pacchetti conformemente con l'MTU della rete, è gestito dallo strato di trasporto. Il principale protocollo di questo strato è il [Transmission Control Protocol \(TCP\)](#) che, insieme al protocollo IP, forma la [suite di protocolli Internet](#), proposti da [Vinton Cerf](#) e [Robert Kahn](#) (mostrati in [Figura 4](#)) nel maggio del 1974 [[Cerf & Kahn, 1974](#)] per essere utilizzati in ARPANET.



**Figura 4.** Foto di Vinton Cerf (sinistra) e Robert Kahn (destra).

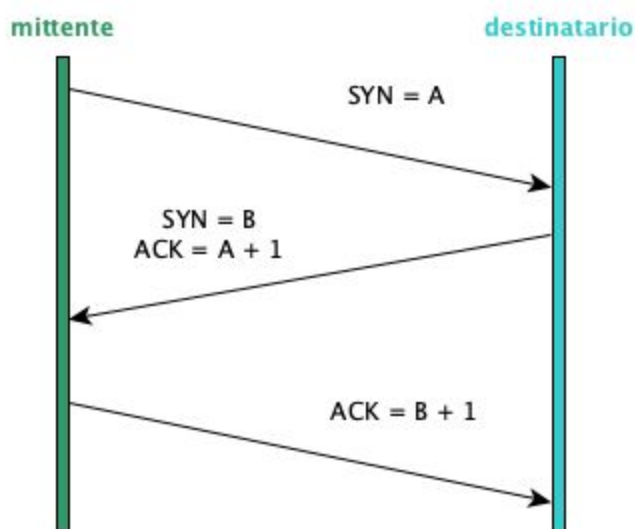
Fotografia di sinistra di Duncan Hull, sorgente: [https://en.wikipedia.org/wiki/File:Dr\\_Vint\\_Cerf\\_ForMemRS.jpg](https://en.wikipedia.org/wiki/File:Dr_Vint_Cerf_ForMemRS.jpg).

Fotografia di destra di Veni Markovski, sorgente: [https://en.wikipedia.org/wiki/File:Bob\\_Kahn.jpg](https://en.wikipedia.org/wiki/File:Bob_Kahn.jpg).



Il TCP permette la consegna affidabile, ordinata, e esente da errori di un flusso di byte tra due computer in comunicazione tra loro attraverso una rete basata sul protocollo IP, e gestisce tutti quei processi che garantiscono la ritrasmissione di un pacchetto nel caso in cui non sia stato recapitato al destinatario entro un certo tempo limite. Come tutti gli altri incapsulamenti visti in precedenza, ogni pacchetto TCP è diviso in due sezioni: l'*header*, che contiene informazioni relative alla comunicazione a livello trasporto tra i due computer mittente e destinatario, e il *payload*, ovvero le informazioni che devono essere scambiate tra i partecipanti alla comunicazione. In particolare, tra i campi importanti dell'*header*, ci sono le **porte** usate dal mittente e dal destinatario per la comunicazione – quest'ultima può essere specificata nell'URL della risorsa da ottenere – e un **numero di sequenza** che indica l'ordine tra i vari pacchetti in modo da poter ricostruire correttamente il messaggio originale.

Contrariamente al protocollo IP, il protocollo TCP permette di stabilire una [comunicazione connessa](#), in cui il mittente e il destinatario si mettono d'accordo di iniziare una comunicazione in modo esplicito prima di scambiarsi i dati, e dichiarano altrettanto esplicitamente quando questa comunicazione si può ritenere conclusa. Il processo di inizio della comunicazione, per esempio, è regolato dal meccanismo del [three-way handshake](#), mostrato in [Figura 5](#).



**Figura 5.** Un esempio grafico di *three-way handshake* per iniziare una comunicazione con il protocollo TCP, in cui il mittente e destinatario si scambiano 3 pacchetti TCP prima di iniziare a scambiare dati.

In pratica, il mittente (nell'esempio dello scenario relativo al click sul link finora discusso, il computer con il browser) che vuole iniziare una comunicazione con un destinatario (il server web dell'esempio) prima invia un pacchetto TCP di sincronizzazione (che verrà incapsulato in un pacchetto IP dallo strato di rete, come spiegato nel precedente capitolo), con il campo dell'*header* SYN impostato ad un numero intero scelto casualmente (il numero "A" in [Figura 5](#)).

Non appena il destinatario riceve questo pacchetto TCP, risponde al mittente inviandogli un altro pacchetto con il campo dell'*header* ACK impostato al numero successivo rispetto a quello ricevuto (A+1) e con il campo SYN impostato ad un nuovo numero intero scelto casualmente (B). Una volta che il mittente riceve questo pacchetto TCP, risponde nuovamente al destinatario con un nuovo pacchetto, specificando nel campo ACK il numero successivo a quello appena ricevuto (B+1). Da quel momento, il mittente è autorizzato a mandare il messaggio, suddiviso in opportuni pacchetti, al destinatario e, conseguentemente, a ricevere la relativa risposta. Un approccio simile si usa anche per la chiusura della connessione, ma in questo caso si usa un *four-way handshake*, ovvero il mittente e il destinatario si scambiano 4 pacchetti TCP per terminare la comunicazione.

In pratica, il protocollo TCP implementa quello che normalmente avviene quando dobbiamo chiamare, per esempio, un qualche centro servizi di un operatore telefonico per chiedere delle informazioni. Di solito, prima ci mettiamo d'accordo col sistema (usualmente una voce pre-registrata) su quale sia il tipo di comunicazione da effettuare (una sorta di rappresentazione metaforica del three-way handshake), e solo allora un operatore risponde alla chiamata ed iniziamo effettivamente ad interagire con lui chiedendogli informazioni (il momento in cui inviamo i dati).

## Altri protocolli usati su Internet

In questo e nei precedenti capitoli abbiamo visto i principali usati su Internet e un particolare protocollo dello strato applicazione, ovvero HTTP, il principale meccanismo di comunicazione usato sul Web. Tuttavia, HTTP non è il solo protocollo dello strato applicazione largamente usato, ma è accompagnato da diversi altri che sono usati quotidianamente da molti utenti del Web. In questa sezione presentiamo alcuni dei più importanti, tutti totalmente basati sul protocollo TCP per lo strato di trasporto.

**SMTP.** Il [Simple Mail Transfer Protocol \(SMTP\)](#) è il principale protocollo di comunicazione dello strato applicativo per spedire email. Ogni volta che si spedisce una mail, in realtà non viene recapitata direttamente al destinatario, ma viene raccolta da un server di posta di competenza del dominio della mail del destinatario – dominio identificato dalla parte dell'indirizzo email che segue il carattere "@". Questo server è come un server web, ma invece di gestire il protocollo HTTP e di restituire pagine web, è in grado di ricevere email attraverso il protocollo SMTP, per poi inoltrarle ai relativi destinatari quando questi ne fanno richiesta mediante l'uso del protocollo POP o IMAP, introdotti di seguito.

**POP.** Il [Post Office Protocol \(POP\)](#) è uno dei due protocolli utilizzato per ricevere email da un server di posta. In pratica, ogni volta che se ne fa richiesta, ad esempio attraverso un client per email (Apple Mail, Outlook, Mozilla Thunderbird), il server di posta consegna tutti i messaggi precedentemente inviati a quel particolare destinatario e, una volta consegnati, li cancella dal server.

**IMAP.** L'[Internet Message Access Protocol \(IMAP\)](#) è l'altro protocollo utilizzato per ricevere email da un server di posta. È un protocollo più moderno rispetto a POP, ed è stato esplicitamente sviluppato per facilitare la gestione di una stessa casella di posta elettronica da parte di più dispositivi. La differenza principale rispetto a POP è che i messaggi di posta elettronica non vengono eliminati automaticamente dal server anche se sono stati già recapitati ai rispettivi destinatari a seguito di una loro richiesta, ma devono essere eliminati esplicitamente dall'utente, se lo desidera.

**FTP.** Il [File Transfer Protocol \(FTP\)](#) è un protocollo che permette il trasferimento di un qualunque file da un computer ad un server e viceversa. È uno dei protocolli più adottati di Internet per permettere lo scaricamento di una grossa mole di dati, mentre è poco adatto per scaricare documenti molti piccoli, come pagine web.

## Scenario completo di richiesta di una pagina web

La spiegazione del protocollo TCP permette di comprendere a pieno cosa succede davvero quando si clicca su un link ipertestuale di una pagina web, la nostra domanda iniziale – che è stata la scusa per poter parlare di tutte le tecnologie principali di Internet e del Web. L'esempio completo è mostrato in [Figura 6](#).

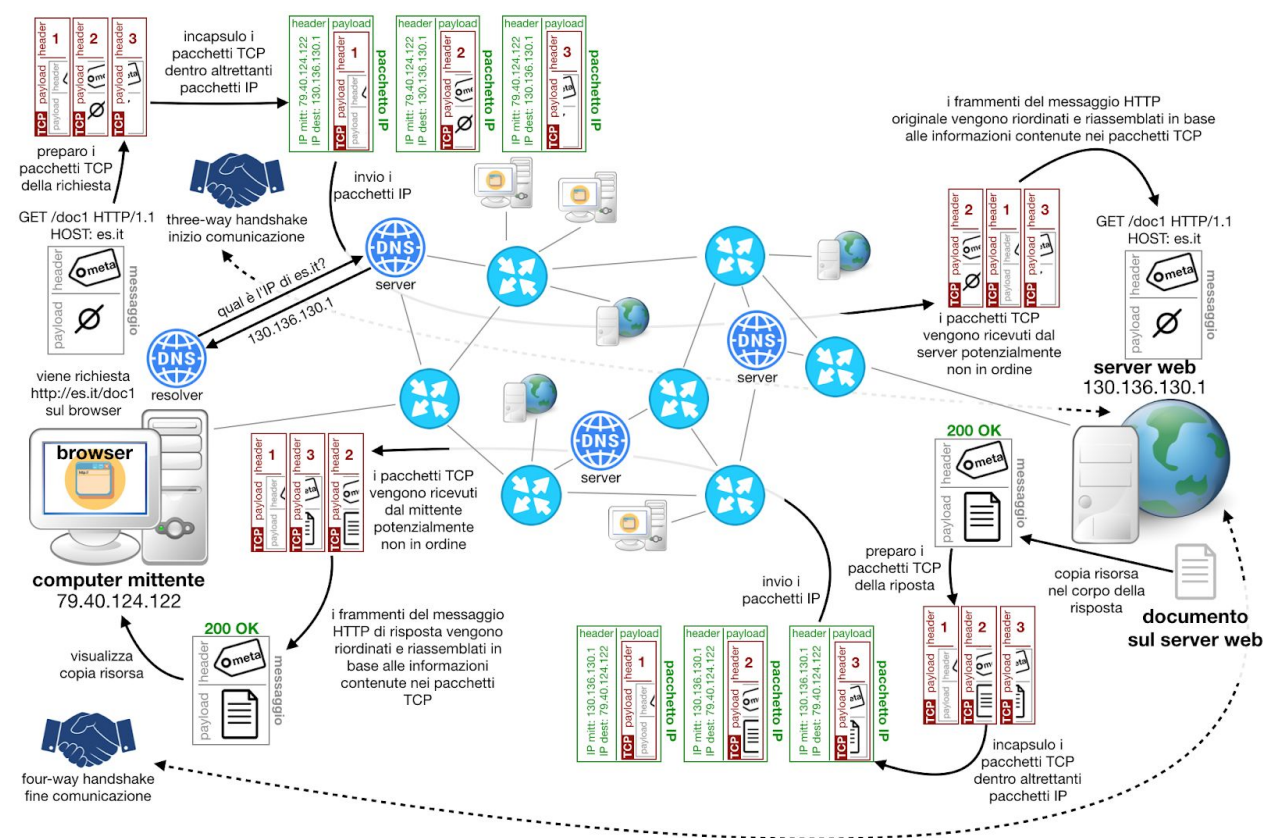
Quando un utente, attraverso il browser sul suo computer collegato a Internet (e, per questa ragione, identificato da uno specifico indirizzo IP, ad esempio `79.40.124.122`), clicca su un collegamento ipertestuale della pagina che sta navigando, quello che in realtà succede è richiedere delle informazioni associate ad uno specifico URL in qualche modo incapsulato in quel link – ad esempio `http://es.it/doc1`. Il browser interpreta questa richiesta, e prepara un pacchetto HTTP contenente il messaggio di richiesta, suddiviso in un *header* contenente informazioni sulla richiesta e un *payload*, che in questo specifico caso è vuoto.

A partire dall'informazione relativa all'*host* (la parte `es.it` dell'URL), viene demandato al DNS resolver il compito di recuperare l'indirizzo IP del server web che dovrebbe contenere informazioni sulla risorsa identificata dall'URL della richiesta, così da poterla instradare correttamente in rete. Una volta recuperata questa informazione (nell'esempio `130.136.130.1`), e capito (in qualche modo più o meno automatico) quale sia il massimo volume di dati per pacchetto che la rete può instradare, si iniziano a preparare i vari pacchetti per la comunicazione.

Lo strato di trasporto è responsabile per la creazione dei pacchetti TCP. Una volta effettuato il *three-way handshake* per iniziare la comunicazione con il server web, il messaggio HTTP viene frammentato (se serve) e incapsulato nel *payload* di uno o più pacchetti TCP, che specificano nell'*header* un numero di sequenza che verrà usato poi dal server web per ricostruire il messaggio HTTP originale nel caso in cui i frammenti non pervengano nell'ordine corretto. La



dimensione di ogni pacchetto TCP è già conforme alla Maximum Transmission Unit (MTU) della rete su cui i pacchetti verranno instradati.



**Figura 6.** Il diagramma completo, con tutte le tecnologie finora introdotte, che mostra l'interazione tra un browser e il server web a seguito di un click su un link ipertestuale.

A questo punto, i pacchetti TCP vengono presi in carica dallo strato di rete, che li incapsula nuovamente all'interno del *payload* di altrettanti pacchetti IP, che nell'*header* specificano l'indirizzo IP di computer mittente e l'indirizzo IP del server web destinatario del messaggio, quest'ultimo recuperato in precedenza grazie al DNS. Fatto ciò, i pacchetti vengono instradati (passando dallo strato di accesso alla rete) al router più vicino, e successivamente ai vari router presenti in Internet in modo da poter essere recapitati al server web.

Lo strato di rete del server web riceve i pacchetti IP, ne estrae il *payload* (ovvero, i relativi pacchetti TCP) e lo passa così com'è allo strato di trasporto. Lo strato di trasporto interpreta l'*header*, e in particolare il numero di sequenza dei pacchetti TCP, in modo da ricostruire correttamente il messaggio HTTP di richiesta, che viene poi passato allo strato di applicazione.

A questo punto, il server web, leggendo le informazioni contenute nell'*header* del messaggio HTTP, cerca localmente il documento (nell'esempio, una pagina web) che corrisponde al

percorso richiesto, e specificato nell'URL iniziale. Una volta trovato, prepara un nuovo messaggio HTTP di risposta che include, nel *payload*, una copia della pagina web richiesta.

Il messaggio HTTP di risposta viene passato allo strato di trasporto del server web, che incapsula nuovamente il messaggio HTTP in uno o più pacchetti TCP della dimensione appropriata per essere trasmessi correttamente in rete. A loro volta, come prima, i pacchetti TCP vengono passati allo strato di rete, così da incapsularli in altrettanti pacchetti IP, che vengono poi instradati in Internet, passando dallo strato di accesso alla rete, partendo dal router più vicino. Ovviamente, la connessione TCP è già attiva, avendo fatto in anticipo il *three-way handshake*, e quindi non è necessario ripeterlo. Inoltre, l'indirizzo IP e la porta su cui effettuare la comunicazione del mittente sono già noti, essendo stati specificati nei pacchetti IP e TCP della richiesta iniziale.

Analogamente a quel che ha fatto il server web, il computer mittente ricostruisce il messaggio HTTP di risposta, estraendo le informazioni rilevanti dai vari pacchetti IP e TCP ricevuti. Fatto ciò, il *payload* del messaggio HTTP di risposta, ovvero la pagina web richiesta, viene estratta e visualizzata sul browser – e, se serve, vengono fatte altre richieste HTTP per recuperare informazioni aggiuntive, come le immagini o i video che di solito non sono contenuti nella pagina web ma sono solo riferiti tramite i loro URL, come vedremo nel prossimo capitolo. Infine, la comunicazione TCP a livello di trasporto viene chiusa utilizzando il *four-way handshake*.

## Ringraziamenti

Gli autori ringraziano Eugenia Galli per le preziose correzioni proposte al testo.

## Bibliografia

Bemer, R. W. (1959). A proposal for a generalized card code for 256 characters. *Communications of the ACM*, 2(9), 19–23. <https://doi.org/10.1145/368424.368435>

Cerf, V., & Kahn, R. (1974). A Protocol for Packet Network Intercommunication. *IEEE Transactions on Communications*, 22(5), 637–648. <https://doi.org/10.1109/TCOM.1974.1092259>

Shannon, C. E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal*, 27(3), 379–423. <https://doi.org/10.1002/j.1538-7305.1948.tb01338.x>