Modelli di comunicazione nella rete

Scritto da Febogamedev

I dati viaggiano nella rete sotto forma di pacchetti.
Ora vediamo i modelli con cui questi pacchetti vengono scambiati tra due applicazioni.

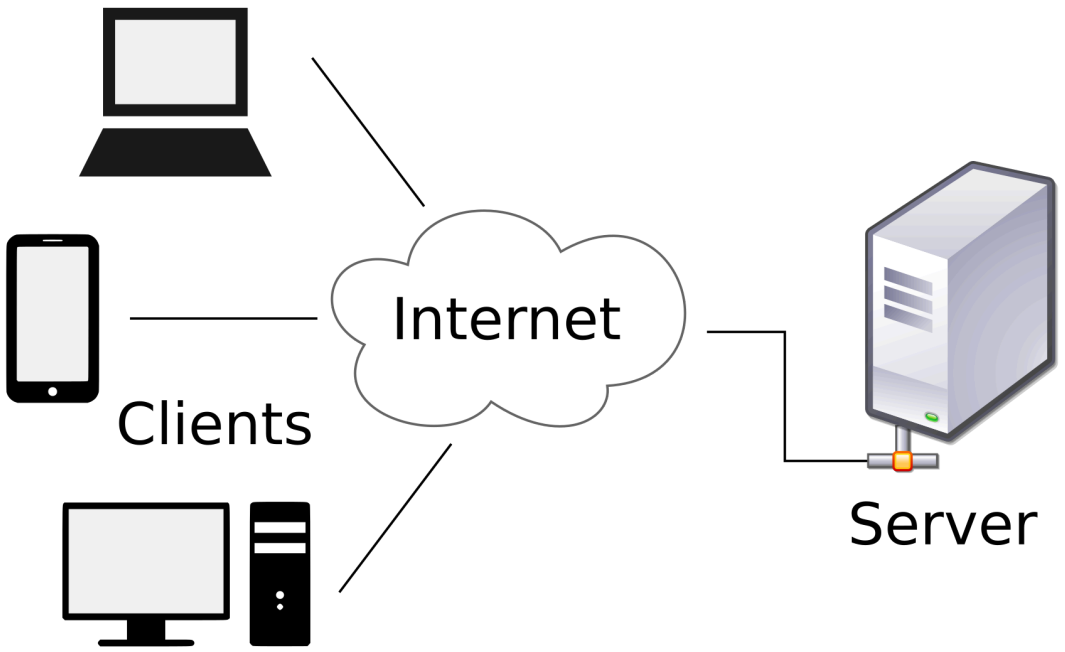
- Modello client-server.
- Modello peer-to-peer.

Modello client-server

Il modello client-server è un'architettura di rete in cui un dispositivo o processo software, chiamato client, invia richieste a un altro dispositivo o processo software, chiamato server, per ottenere un servizio (Kurose & Ross, 2026; Tanenbaum et al., 2022).

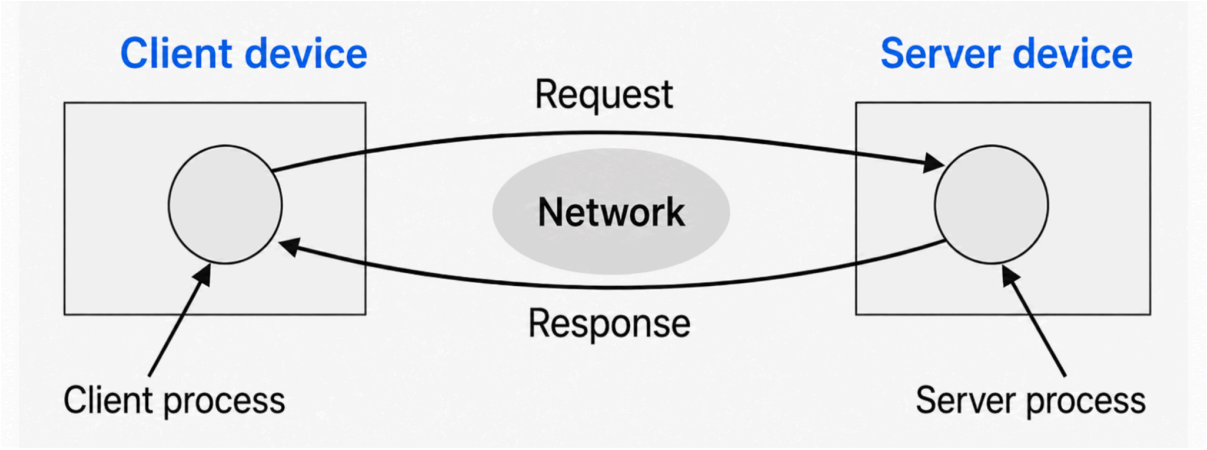
La presenza di un server permette a un certo numero di client di condividerne le risorse, lasciando che sia il server a gestire gli accessi alle risorse per evitare conflitti di utilizzazione.

Molti servizi disponibili su Internet funzionano secondo il modello client-server: per esempio, quando visitiamo un sito web, il browser invia una richiesta a un server web, che restituisce la pagina richiesta.



Nello specifico, nel modello client-server vediamo che sono coinvolti due processi (programmi in esecuzione) uno sulla macchina client e uno sulla macchina server.
Da queste premesse, la comunicazione è rappresentata da questa logica:

- Un processo client manda un messaggio attraverso la rete al processo server per poi restare in attesa di un messaggio di risposta.
- Quando il processo server riceve la richiesta, la esegue e restituisce una risposta.



È importante specificare che un server non è necessariamente una macchina fisica: può essere anche un processo software che rimane in ascolto su una porta di rete, in attesa di richieste da parte dei client. Una stessa macchina può quindi ospitare più server contemporaneamente, cioè più applicazioni o servizi in esecuzione, ciascuno raggiungibile dai client (Kurose & Ross, 2026).
Anche il client non è necessariamente una macchina fisica: può essere un processo software che invia richieste a un server per ottenere un servizio. Una stessa macchina può eseguire più processi client contemporaneamente, per esempio un browser web, un'app di posta, un videogioco online o un'app di messaggistica.

I client e il server sono in collegamento tramite un **protocollo di comunicazione,** cioè un insieme di regole condivise che consentono di dialogare per uno specifico compito (Kurose & Ross, 2026).

Client

Il termine **client** indica un componente o sottosistema informatico che accede ai servizi o alle risorse di un server.
Il client può essere hardware o software:

- Un dispositivo, collegato a un server tramite rete, al quale richiede uno o più servizi, utilizzando uno o più protocolli di rete, è un esempio di client hardware.
- Invece un programma di posta elettronica è un esempio di client software perché il software comunica continuamente con il server che riceve e invia i messaggi di posta elettronica.

Ad esempio uno dei più importanti client software è il **browser**, un software specializzato per recuperare e visualizzare pagine web recuperate da un server web.

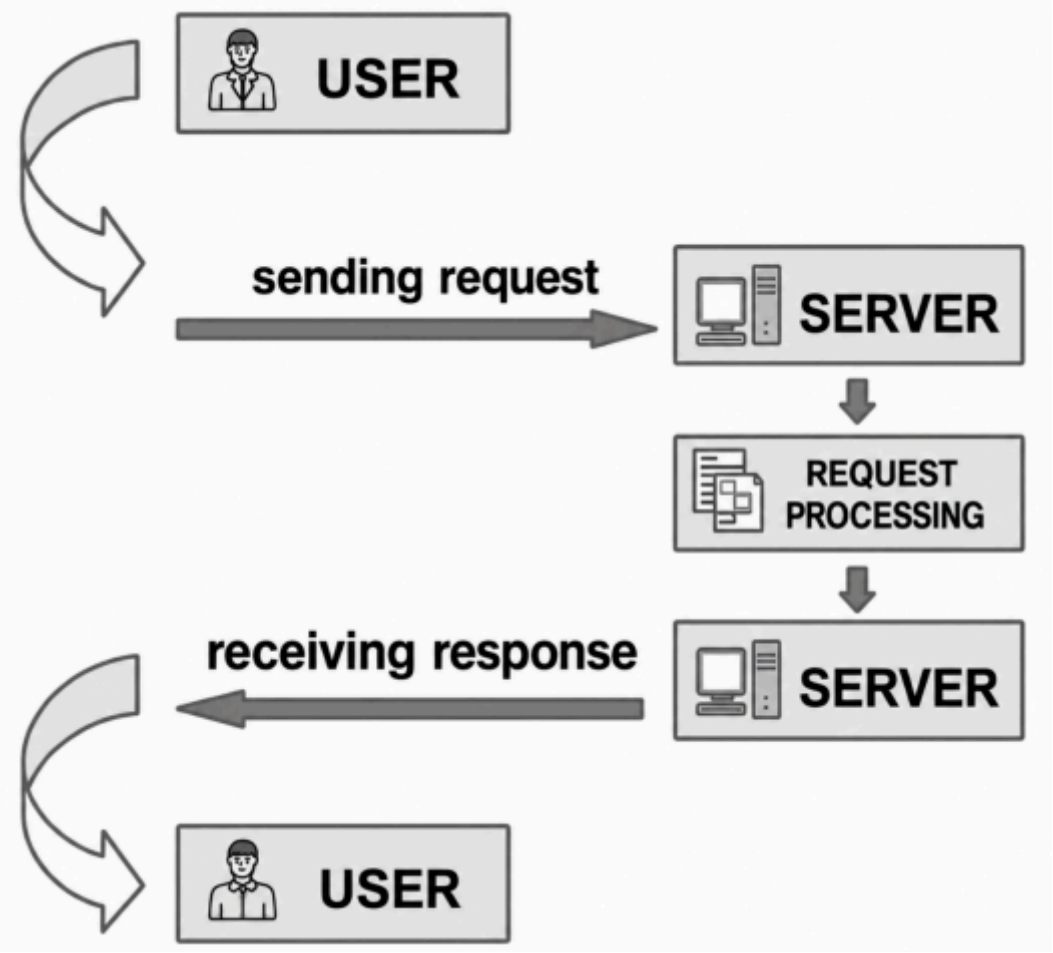
In alcuni sistemi il client svolge soprattutto il ruolo di interfaccia verso il server; in altri casi, invece, può essere un software molto complesso, capace di elaborare molti dati localmente.

Esistono alcuni software, come l'email, che sono divisi in una parte client (residente ed in esecuzione sul dispositivo client) e una parte server (residente ed in esecuzione sul server).

Esempio
Quando si utilizza Gmail tramite browser o app, l'interfaccia utente è il client. I messaggi, invece, sono conservati sui sistemi server del servizio. Il client invia richieste al server, che recupera i dati e li restituisce per mostrarli all'utente.

Server

Un **server (**dall'inglese to serve, "servire", dunque letteralmente «serviente, servitore») è un dispositivo o un software che fornisce servizi ad altri dispositivi o programmi, chiamati client (Tanenbaum et al., 2022).

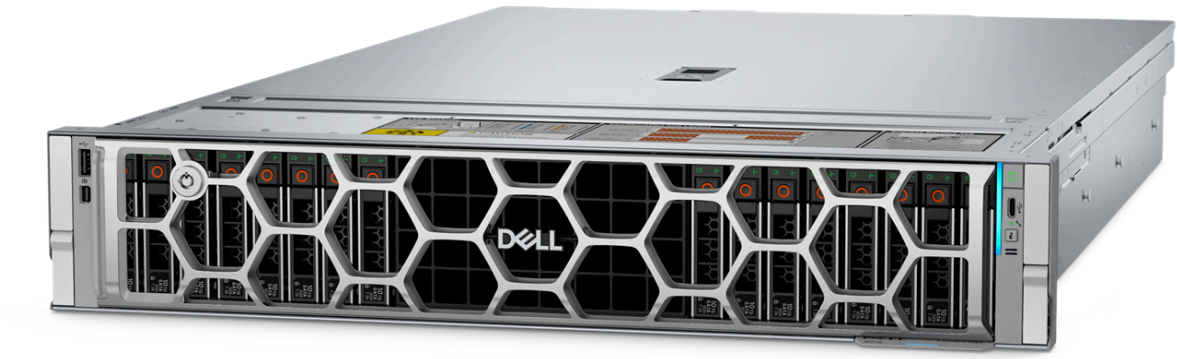


In base al contesto, il termine "server" può indicare quindi:

- Un computer "ordinario" utilizzato per fornire servizi ad altri computer, a prescindere dalle sue caratteristiche hardware.

- Un computer specifico appartenente alla fascia di mercato dedicata all'uso come server, caratterizzato da alta affidabilità, maggiori prestazioni e funzioni aggiuntive.
- Un software che fornisce servizi ad altri software.

Un server può occuparsi di gestire gli accessi degli utenti, assegnare e liberare risorse, condividere dati e proteggere le informazioni.



Server tipico

Esempio
Un server di posta elettronica è paragonabile ad un qualunque ufficio postale.
Gli utilizzatori per accedere via client alla loro casella di posta elettronica devono essere stati autorizzati.
In modo analogo un utente deve possedere la chiave della cassetta situata presso un ufficio postale dalla quale vuole prelevare la corrispondenza.

Tipi di server

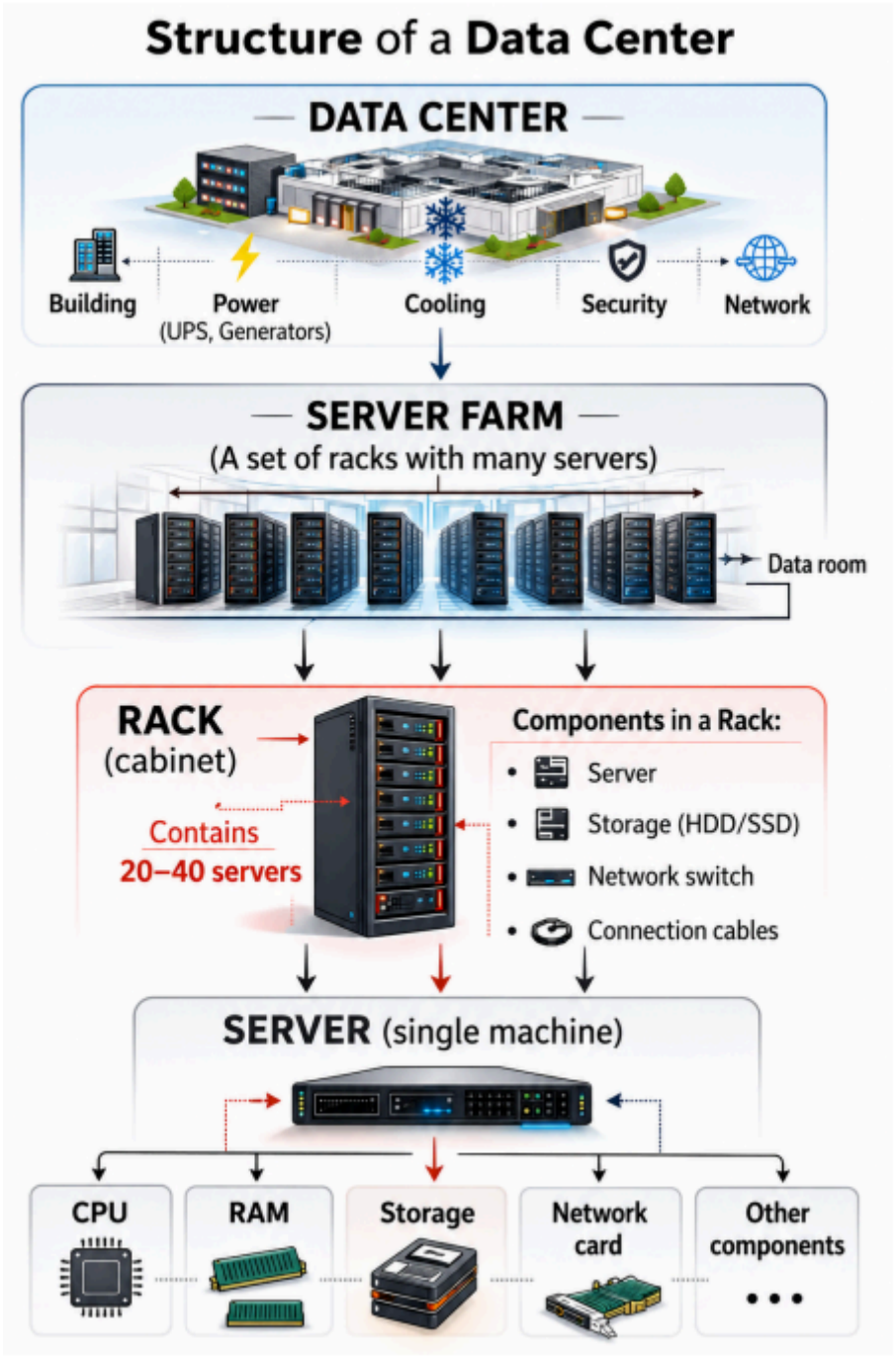
In molti casi, quando parliamo di server ci riferiamo a servizi software, cioè programmi che svolgono funzioni specifiche e rispondono alle richieste dei client.
Tra i principali tipi di server troviamo:

- File server: permette agli utenti di accedere ai file situati su un dispositivo di memorizzazione come se fossero sul proprio dispositivo, agevolando la condivisione di informazioni.
- Database server: gestisce database e risponde alle richieste di lettura, scrittura, modifica o eliminazione dei dati.
- *FTP* server: consente l'accesso a file e directory tramite il protocollo FTP
- Web server: usato per ospitare siti web.
- Application server: esegue la logica di un'applicazione web, per esempio gestendo funzioni, utenti, dati e comunicazioni tra interfaccia e database.
- Mail server: usato per la gestione della posta elettronica.
- Game server**:** ospita risorse per rendere possibili i giochi multiutente.

Data center

Un data center è una struttura specializzata progettata per ospitare server e tutti gli altri componenti ad essi collegati: dispositivi di archiviazione, apparati di rete, impianti di sicurezza e infrastrutture energetiche (Barroso et al., 2026).
La funzione di un data center è quella di archiviare, elaborare e distribuire grandi volumi di dati in modo continuo, affidabile e sicuro.
In sostanza sono le “fabbriche dell’informazione”: luoghi dove i dati vivono, transitano e vengono gestiti 24 ore su 24.

Una parte fondamentale di un data center è la **server farm,** cioè un insieme organizzato di server installati nello stesso luogo fisico.
Una server farm è composta da armadi metallici chiamati “**rack**”, ciascuno dei quali contiene decine di server.



Questi server ospitano siti web, database, applicazioni aziendali, video in streaming e perfino mondi virtuali dei videogiochi online.
Oltre alla server farm, un data center è dotato di:

- Sistemi di raffreddamento, per mantenere le temperature entro limiti accettabili (i server generano molto calore).
- Gruppi di continuità (*UPS*) e generatori, per garantire l'alimentazione anche in caso di blackout.
- Connessioni in fibra ottica ad altissima velocità.
- Sistemi di sicurezza fisica e monitoraggio digitale costante, per prevenire accessi non autorizzati o guasti.

L’intero sistema è pensato per garantire alta disponibilità, cioè un funzionamento continuo anche in condizioni critiche (Barroso et al., 2026).
Un’interruzione di pochi secondi può significare enormi perdite economiche o danni all'immagine per chi fornisce servizi digitali.

Senza i data center, non ci sarebbe Internet per come lo conosciamo.
La maggior parte dei servizi digitali che usiamo ogni giorno, come email, messaggistica, streaming video, social network e piattaforme cloud, si appoggia a uno o più data center.
Anche i sistemi informativi delle aziende, le banche, gli ospedali, le app di navigazione o i servizi pubblici digitali si appoggiano a queste infrastrutture.



Rack con server all’interno

Sistema peer-to-peer

Molti servizi disponibili su Internet utilizzano il modello client-server ma esiste un altro popolare modello: il **peer-to-peer** (*P2P*).

Una rete peer-to-peer è un modello di rete in cui i dispositivi collegati, chiamati **peer**, possono comunicare direttamente tra loro senza dipendere necessariamente da un server centrale (Kurose & Ross, 2026; Tanenbaum et al., 2022).
La parola inglese peer significa “pari”. Questo ci aiuta a capire il concetto principale: in una rete peer-to-peer i computer non sono divisi rigidamente in client e server, ma possono svolgere entrambi i ruoli.

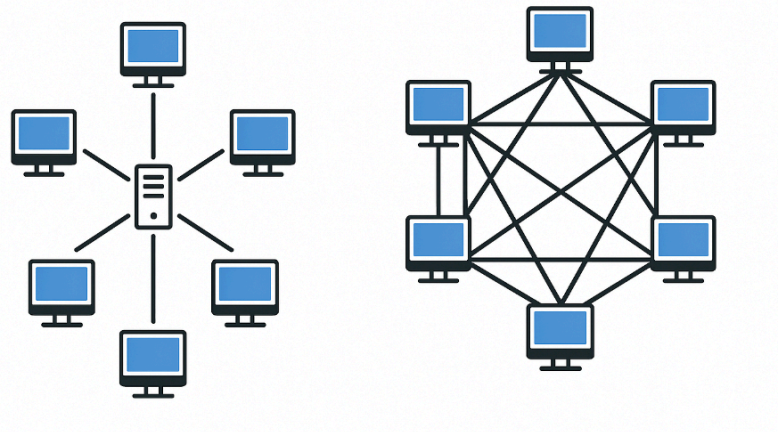
Un dispositivo può quindi:

- Richiedere dati a un altro dispositivo.
- Inviare dati ad altri dispositivi.
- Condividere file, risorse o servizi.
- Partecipare al funzionamento generale della rete.

Ogni peer può comportarsi sia da client, quando richiede una risorsa, sia da server, quando la fornisce.

Nel modello client-server, la comunicazione è organizzata intorno a uno o più server principali. I client non comunicano sempre direttamente tra loro, ma spesso passano attraverso il server.
Per esempio, quando visitiamo un sito web, il nostro browser è il client e il server web è il sistema che ospita le pagine richieste. Il client invia una richiesta, il server risponde inviando i dati necessari.

Nel modello peer-to-peer, invece, i dispositivi possono scambiarsi dati direttamente.
Non c'è necessariamente un server centrale che controlla tutta la comunicazione.
Questa è la differenza più importante: il client-server è più centralizzato, mentre il peer-to-peer è più distribuito (Kurose & Ross, 2026).



A sinistra un sistema client-server, a destra un sistema P2P

Ogni peer può mettere a disposizione una parte delle proprie risorse, per esempio: file, spazio di archiviazione, potenza di calcolo, informazioni utili al funzionamento del sistema, ecc.
Per questo motivo, in una rete peer-to-peer il funzionamento complessivo dipende dalla collaborazione tra molti peer.

Esempio
Immaginiamo una classe in cui tutti gli studenti hanno un file diverso sul proprio computer.

- Nel modello client-server, tutti i file sarebbero caricati su un unico computer centrale, per esempio quello del docente o un server della scuola. Gli studenti scaricherebbero i file da lì.

- Nel modello peer-to-peer, invece, ogni studente potrebbe condividere direttamente il proprio file con gli altri. Se uno studente ha bisogno di un file, può riceverlo direttamente dal computer di un compagno.

Non tutte le reti peer-to-peer funzionano nello stesso modo. Possiamo distinguere, in modo semplificato, tra reti ***P2P* pure** e **reti *P2P* ibride**.

- In una rete *P2P* pura, non esiste un server centrale. Tutti i peer hanno un ruolo simile e collaborano direttamente tra loro. La rete è completamente distribuita.
- In una rete *P2P* ibrida, invece, può esistere un server centrale che aiuta i peer a trovarsi, ma lo scambio dei dati avviene direttamente tra i dispositivi. Per esempio, un server potrebbe contenere l'elenco dei peer disponibili, ma non ospitare direttamente tutti i file scambiati. In questo caso il server non elimina la natura peer-to-peer della rete, perché i dati principali vengono comunque trasferiti tra i peer (Tanenbaum et al., 2022).

Uno degli usi più conosciuti delle reti peer-to-peer è la condivisione di file.
In un sistema di file sharing *P2P*, un file può essere diviso in tante parti.
Ogni peer può scaricare alcune parti del file da peer diversi e, allo stesso tempo, condividere con altri le parti che possiede già.
Questo meccanismo è molto diverso dal download tradizionale da un unico server. Nel modello client-server, se molti utenti scaricano lo stesso file dallo stesso server, quel server può sovraccaricarsi. Nel modello *P2P*, invece, più utenti partecipano alla rete e più aumentano le possibili fonti da cui scaricare (Kurose & Ross, 2026).
Per questo motivo il *P2P* può essere molto efficiente nella distribuzione di grandi quantità di dati.

Il modello peer-to-peer presenta diversi vantaggi:

- La distribuzione del carico: poiché non esiste necessariamente un unico server centrale, il lavoro può essere diviso tra molti dispositivi.
- La scalabilità: in alcuni sistemi *P2P*, quando entrano nuovi peer nella rete, non aumentano solo le richieste, ma anche le risorse disponibili. Ogni nuovo peer può infatti contribuire con banda, dati o capacità di elaborazione (Kurose & Ross, 2026).
- La resistenza ai guasti: se un server centrale smette di funzionare, un sistema client-server può diventare irraggiungibile. In una rete *P2P*, invece, la caduta di un singolo peer non blocca necessariamente tutta la rete (Tanenbaum et al., 2022).

Il modello peer-to-peer presenta però anche alcuni svantaggi:

- La gestione e il controllo: in un sistema client-server è più semplice controllare chi accede ai dati, quali risorse sono disponibili e come vengono gestite. In una rete P2P, invece, i nodi sono distribuiti e possono entrare o uscire dalla rete in qualunque momento.
- La sicurezza: poiché i dati possono provenire da peer diversi, è importante verificare che non siano stati modificati o che non contengano software dannoso.
- L'affidabilità delle risorse: se pochi peer condividono un certo file, quel file può diventare difficile da reperire. Inoltre, se molti peer si scollegano, la qualità del servizio può diminuire.

Una delle domande fondamentali in una rete peer-to-peer è: come fa un peer a trovare gli altri peer?
Nel modello client-server il client conosce l'indirizzo del server, per esempio tramite un URL o un indirizzo IP. Nel modello *P2P*, invece, i peer possono essere molti, cambiare nel tempo e collegarsi da reti diverse.

Per risolvere questo problema, le reti *P2P* possono usare diversi meccanismi di scoperta.

- Alcune usano server di supporto, che aiutano i peer a ottenere una lista di altri peer disponibili.
- Altre reti usano sistemi più distribuiti. In questo caso ogni nodo conosce solo una parte della rete e può indicare altri nodi da contattare. Quando un nuovo peer entra nella rete, parte da alcuni nodi già conosciuti. Da questi ottiene informazioni su altri peer e amplia progressivamente la propria conoscenza della rete.

Torrent

Uno degli esempi più conosciuti di tecnologia peer-to-peer è **BitTorrent**.
Nel linguaggio comune, però, si usa spesso la parola "**torrent**" per indicare un sistema di download basato sul protocollo BitTorrent.

BitTorrent è un protocollo di comunicazione pensato per distribuire file tra molti dispositivi collegati in rete (Cohen, 2008).
A differenza del download tradizionale, in cui un client scarica un file da un unico server, nel sistema torrent il file viene distribuito tra molti utenti.
Ogni utente che partecipa allo scambio è un peer. Questo significa che può sia ricevere dati sia inviarli agli altri.

Nel download tradizionale, il funzionamento è abbastanza semplice: il client invia una richiesta a un server e il server invia il file richiesto.
Per esempio, se scarichiamo un programma dal sito ufficiale del produttore, il nostro computer si collega al server dell'azienda e riceve da lì il file completo.

Nel torrent, invece, il file non arriva necessariamente da un unico server. Il file viene diviso in tante piccole parti, chiamate pezzi o blocchi. Il nostro computer può scaricare alcuni pezzi da un peer, altri pezzi da un altro peer, e così via (Cohen, 2008).

Allo stesso tempo, mentre scarichiamo, possiamo anche condividere con altri utenti i pezzi che abbiamo già ricevuto.
Questo è il punto fondamentale: in un sistema torrent, chi scarica può anche contribuire alla distribuzione del file.

Per avviare un download torrent, spesso si usa un piccolo file con estensione ".torrent".
Questo file non contiene il contenuto vero e proprio che vogliamo scaricare. Per esempio, se vogliamo scaricare un video, il file .torrent non contiene il video completo.
Il file .torrent contiene invece informazioni utili al programma torrent, come:

- Il nome del file da scaricare.

- La dimensione del file.
- La divisione del file in pezzi.
- Informazioni per controllare che i pezzi scaricati siano corretti.
- Indicazioni su come trovare altri peer.

Quindi il file .torrent è una specie di “mappa” che permette al programma di sapere cosa deve cercare e come ricostruire il file finale (Cohen, 2008).

Per usare un torrent serve un programma apposito, chiamato **client torrent**.
Esempi di client torrent sono: qBittorrent, Transmission, µTorrent, BitTorrent Web.

Il client torrent legge il file “.torrent” oppure un collegamento chiamato **magnet link**, cerca altri peer disponibili e inizia a scaricare i pezzi del file (Hazel & Norberg, 2008).
Il client si occupa anche di ricomporre i pezzi scaricati nell’ordine corretto, fino a ottenere il file completo.

Un magnet link è un collegamento che contiene le informazioni essenziali per identificare il file da scaricare. Invece di scaricare prima un file “.torrent”, l’utente può aprire direttamente il magnet link con il client torrent (Hazel & Norberg, 2008).
Il client usa queste informazioni per cercare altri peer e recuperare i dati necessari per iniziare il download.
Dal punto di vista dell’utente, il magnet link è spesso più pratico perché basta cliccare su un collegamento.

Nel linguaggio dei torrent si usano spesso due termini: **seed** e **leech**.
- Un seed è un peer che possiede già il file completo e lo sta condividendo con gli altri.
- Un leech, o più precisamente un peer in fase di download, è un utente che sta ancora scaricando il file. Anche se non possiede ancora il file completo, può comunque condividere con altri i pezzi che ha già ricevuto.

Più seed ci sono, più è facile e veloce scaricare il file, perché esistono più copie complete disponibili nella rete.
Se invece ci sono pochi seed, il download può essere lento o bloccarsi, perché alcuni pezzi del file potrebbero essere difficili da trovare.

L’insieme dei peer che stanno condividendo uno stesso file prende il nome di **swarm**, cioè “sciame”.

Uno swarm è quindi formato da tutti gli utenti che partecipano alla distribuzione di uno specifico contenuto: alcuni hanno il file completo, altri ne possiedono solo una parte.

Nei sistemi torrent esistono principalmente due modi per trovare i peer dello swarm di uno specifico contenuto: **il tracker** e la ***DHT***.

Il tracker è un server che aiuta i peer a trovarsi tra loro. Non contiene necessariamente il file da scaricare, ma mantiene informazioni sui peer che stanno partecipando allo swarm.
Quando un client torrent vuole scaricare un file, può contattare il tracker per ottenere una lista di altri peer disponibili (Cohen, 2008).

La *DHT*, sigla di Distributed Hash Table, cioè “tabella hash distribuita”, serve a trovare altri peer senza dipendere da un unico server centrale (Loewenstern & Norberg, 2008).
Invece di chiedere a un tracker chi sta condividendo un file, il client torrent interroga una rete distribuita composta da molti peer.
Possiamo immaginarla come una grande rubrica divisa tra tanti computer: ogni peer conosce solo una parte delle informazioni, ma collaborando con gli altri permette di trovare i peer che stanno condividendo un certo torrent.
Tramite *DHT*, il client non interroga tutti i peer della rete.
Usa invece le informazioni del torrent per contattare progressivamente alcuni nodi della rete distribuita, fino a trovare peer che stanno condividendo quel contenuto (Loewenstern & Norberg, 2008).

Materiale di supporto

Esercizi

Cosa sta facendo davvero il client?
Uno studente usa contemporaneamente sul proprio computer:

- Un browser per visitare il sito della scuola.
- Un programma di posta elettronica.
- Un videogioco online.
- Un'applicazione di messaggistica.

Lo studente sostiene:
«Il mio computer è il client e tutti i computer a cui si collega sono server.»

Analizza la sua affermazione.

- Spiega perché parlare semplicemente di “computer client” e “computer server” può essere impreciso.
- Individua quali processi client potrebbero essere in esecuzione contemporaneamente sul computer dello studente.
- Spiega come una singola macchina possa eseguire più client nello stesso momento.
- Descrivi, scegliendo una delle applicazioni indicate, la sequenza richiesta → elaborazione → risposta.
- Spiega quale funzione svolge il protocollo di comunicazione tra client e server.

Progettare i servizi di una scuola
Una scuola vuole realizzare una propria infrastruttura informatica.

Deve offrire questi servizi:

- Conservare documenti condivisi tra docenti.
- Ospitare il sito web della scuola.
- Gestire un database con studenti, classi e voti.
- Gestire la posta elettronica interna.
- Eseguire la logica di una nuova applicazione web scolastica.

Per ciascun servizio:

- Individua il tipo di server più adatto tra quelli descritti nella dispensa.
- Spiega quale servizio fornisce ai client.
- Indica un possibile client che potrebbe utilizzarlo.
- Spiega perché non è necessario utilizzare cinque computer fisici differenti per avere cinque server differenti.
- Descrivi una possibile situazione in cui uno stesso computer svolga contemporaneamente più funzioni server.

Un servizio che non può fermarsi

Una piattaforma di streaming viene utilizzata da milioni di utenti. I suoi responsabili pensano di conservare tutti i servizi necessari su alcuni server installati in una normale stanza dell'azienda.

La stanza dispone di connessione Internet e corrente elettrica, ma non possiede sistemi particolari di raffreddamento, generatori, UPS o sistemi avanzati di sicurezza.

Analizza la soluzione proposta.

- Individua almeno quattro problemi che potrebbero verificarsi.
- Spiega perché un semplice insieme di server non costituisce necessariamente un'infrastruttura adeguata per un servizio di queste dimensioni.
- Proponi quali componenti di un data center sarebbero necessari per migliorare l'affidabilità del servizio.
- Spiega quale funzione avrebbero server farm e rack all'interno della struttura.
- Un blackout dura cinque minuti: spiega quali sistemi dovrebbero permettere al servizio di continuare a funzionare.
- Spiega perché, per una piattaforma di questo tipo, è importante garantire alta disponibilità.

Client-server o peer-to-peer?
Una scuola deve distribuire un file video da 4 GB a 500 studenti.
Sono state proposte due soluzioni.

- Soluzione A: tutti gli studenti scaricano il file da un unico server della scuola.
- Soluzione B: inizialmente alcuni studenti ricevono parti del file e successivamente gli studenti possono scambiarsi direttamente tra loro le parti già possedute.

Confronta le due soluzioni.

- Indica quale soluzione segue principalmente il modello client-server e quale il modello peer-to-peer.
- Descrivi come viaggerebbe il file nei due casi.
- Spiega cosa potrebbe accadere al server nella soluzione A se tutti gli studenti iniziassero contemporaneamente il download.
- Spiega perché nella soluzione B l'arrivo di nuovi peer può aumentare anche le risorse disponibili.
- Individua almeno un vantaggio e uno svantaggio della soluzione B.
- Scegli quale soluzione utilizzeresti e motiva la scelta considerando carico, affidabilità, controllo e sicurezza.

P2P puro o P2P ibrido?
Un'applicazione permette agli utenti di condividere direttamente file tra i loro computer.

Quando un nuovo utente apre il programma, però, il software contatta un server centrale che gli fornisce una lista di altri utenti disponibili. Una volta trovati gli altri dispositivi, i file vengono trasferiti direttamente tra gli utenti e non passano attraverso il server centrale.

Uno studente sostiene:
«Non può essere peer-to-peer perché esiste un server.»

Valuta questa affermazione.

- Spiega se il sistema descritto può essere considerato P2P.
- Stabilisci se si tratta più precisamente di P2P puro o P2P ibrido.
- Descrivi il ruolo svolto dal server centrale.
- Descrivi il ruolo svolto dai peer.
- Spiega cosa cambierebbe se anche i file fossero conservati sul server e tutti gli utenti dovessero scaricarli esclusivamente da lì.
- Immagina invece un sistema completamente privo di server centrale: descrivi quale problema dovrebbe comunque essere risolto affinché i peer riescano a comunicare tra loro.

Analizzare un download torrent

Vuoi scaricare legalmente un'immagine ISO di una distribuzione Linux attraverso BitTorrent.

Apri un magnet link con un client torrent. Il programma trova uno swarm formato da:

- 8 seed.
- 35 peer che stanno ancora completando il download.

Durante il download, il tuo computer riceve pezzi differenti del file da diversi peer e contemporaneamente invia ad altri alcuni pezzi che possiede già.

Analizza ciò che sta avvenendo.

- Spiega perché il tuo computer si comporta contemporaneamente da client e da server.
- Spiega la differenza tra seed, peer in fase di download e swarm.
- Descrivi perché il file può essere ricevuto da più dispositivi anziché da un unico server.
- Spiega come il client torrent può controllare che i pezzi ricevuti siano corretti e ricostruire il file finale.
- Spiega a cosa servirebbe un file .torrent e perché non contiene necessariamente il file che vuoi scaricare.
- Spiega quale funzione può svolgere un tracker.
- Spiega come la DHT permette invece di trovare peer senza dipendere da un unico server centrale.
- Immagina che tutti i seed si disconnettano mentre nessuno degli altri peer possiede un particolare pezzo del file: prevedi cosa potrebbe accadere al download e motiva la risposta usando i principi del P2P.

Bibliografia

Barroso, L. A., Hölzle, U., & Ranganathan, P. (2026). *The data center as a computer: Designing warehouse-scale machines* (4th ed.). Springer. https://doi.org/10.1007/978-3-031-99489-0

Cohen, B. (2008). *The BitTorrent protocol specification* (BEP 3). BitTorrent Enhancement Proposals. https://www.bittorrent.org/beps/bep_0003.html

Hazel, G., & Norberg, A. (2008). *Extension for peers to send metadata files* (BEP 9). BitTorrent Enhancement Proposals. https://www.bittorrent.org/beps/bep_0009.html

Kurose, J. F., & Ross, K. W. (2026). *Computer networking: A top-down approach* (9th ed.). Pearson.

Loewenstern, A., & Norberg, A. (2008). *DHT protocol* (BEP 5). BitTorrent Enhancement Proposals. https://www.bittorrent.org/beps/bep_0005.html

Tanenbaum, A. S., Feamster, N., & Wetherall, D. J. (2022). *Computer networks* (6th ed.). Pearson.

Approfondimenti

- James F. Kurose e Keith W. Ross, Reti di calcolatori e Internet. Un approccio top-down, 8ª ed., Pearson, 2022 — Un testo chiaro e progressivo per comprendere applicazioni di rete, modello client-server, protocolli e scambio di dati.
- Andrew S. Tanenbaum, David J. Wetherall e Nick Feamster, Reti di calcolatori, 6ª ed., Pearson, 2023 — Una trattazione sistematica delle architetture di rete, dei servizi server e dei principali protocolli di comunicazione.
- Larry L. Peterson e Bruce S. Davie, Reti di calcolatori, Apogeo Education, 2012 — Approfondisce le reti peer-to-peer e overlay, collegandole al funzionamento generale di Internet.
- Andrew S. Tanenbaum e Maarten van Steen, Sistemi distribuiti. Principi e paradigmi, 2ª ed., Pearson, 2007 — Utile per studiare decentralizzazione, coordinamento tra nodi e architetture peer-to-peer.
- Domenico Trisciuoglio, Data Center. Progetto, realizzazione, gestione, Tecniche Nuove, 2025 — Un approfondimento pratico su progettazione, alimentazione, raffreddamento, continuità operativa e gestione dei data center.

Nota editoriale

Per la realizzazione di questi materiali ho utilizzato strumenti di intelligenza artificiale generativa come supporto alla scrittura, in particolare per migliorare la forma del testo, riorganizzare i contenuti, correggere la formulazione e velocizzare alcune attività redazionali.

Lavorando autonomamente alla produzione di questi materiali, cerco infatti di automatizzare tutte le attività che possono esserlo, così da poter dedicare più tempo alla ricerca, alla progettazione e allo sviluppo dei contenuti.

L'intelligenza artificiale non determina però i contenuti dell'opera: la scelta degli argomenti, la struttura, le idee, le interpretazioni, gli esempi e l'impostazione didattica sono elaborati da me. L'IA viene quindi utilizzata principalmente come strumento di supporto alla produzione e alla revisione formale, mentre la responsabilità autoriale e progettuale dei contenuti rimane mia.

Salvo diversa indicazione, questo materiale è distribuito con licenza Creative Commons Attribuzione – Non commerciale – Condividi allo stesso modo 4.0 Internazionale (CC BY-NC-SA 4.0).

È quindi possibile condividere, ridistribuire, modificare e creare opere derivate a partire dal materiale, a condizione di attribuirne correttamente la paternità, di non utilizzarlo per scopi commerciali e di distribuire eventuali versioni modificate o derivate con la stessa licenza.

