Servizi streaming

Scritto da Febogamedev

Tra i principali contenuti che possono essere fruiti attraverso tecnologie di streaming troviamo:

- Audio.
- Video.
- Applicazioni interattive.

Audio

Lo streaming audio permette di ascoltare contenuti sonori attraverso una rete senza dover attendere il download completo del contenuto (Pantos & May, 2017).
A seconda del servizio, l'audio può essere già disponibile oppure prodotto e trasmesso in tempo reale.

Tipologia	Caratteristiche	Esempi
Musica On-Demand (Audio Streaming On-Demand, *AOD*)	L'utente sceglie un brano già disponibile in un catalogo.	Spotify, Apple Music, Tidal, YouTube Music
Radio Streaming (Live Audio Streaming)	Il contenuto viene trasmesso secondo una programmazione continua o live.	TuneIn, BBC Sounds.
Podcast	Episodi preregistrati disponibili quando l'utente desidera ascoltarli.	Apple Podcasts, Spotify, YouTube Music.
Social audio.	Conversazioni audio prodotte e ascoltate in tempo reale.	X Spaces, Discord.
Comunicazione vocale.	Audio bidirezionale a bassa latenza tra più utenti.	WhatsApp, Teams, Zoom, Discord.

Nota:
Una precisazione: un podcast non richiede necessariamente un sistema di streaming a segmenti. Gli episodi possono essere distribuiti come file audio tramite HTTP e riprodotti mentre vengono ricevuti; Apple Podcasts richiede ai server il supporto alle richieste HTTP HEAD e agli intervalli di byte per consentire questa riproduzione (Apple, s.d.).
Dal punto di vista dell'utente l'esperienza appare comunque simile allo streaming.

Nel caso dell'audio on demand, il funzionamento è concettualmente simile allo streaming video.

Un brano musicale è codificato attraverso codec come *AAC*, Opus o altri codec audio, distribuito attraverso server e *CDN* e ricevuto progressivamente dal player.
Una differenza importante è che, a parità di durata e in molte configurazioni d'uso, l'audio compresso richiede un bitrate inferiore rispetto a un flusso audiovisivo. Il fabbisogno effettivo dipende però da codec, qualità e numero di canali.
Un bitrate inferiore può favorire:

- Minore consumo di banda.
- Minore memoria per conservare la stessa durata di contenuto nel buffer.
- Tempi di caricamento ridotti, a parità di condizioni.
- Maggiore tolleranza alle connessioni lente.

Anche lo streaming audio può utilizzare tecniche adattive, scegliendo versioni con bitrate differenti quando cambiano le condizioni della rete (Pantos & May, 2017).

Nello streaming audio live il contenuto viene prodotto e distribuito mentre l'evento è ancora in corso.
In questo caso diventano particolarmente importanti:
Latenza, buffering e stabilità della rete.
Una web radio può tollerare alcuni secondi di latenza senza particolari problemi, mentre una conversazione tra due persone richiede tempi molto più bassi

Tale categoria si trova nel campo del ***VoIP* (Voice over Internet Protocol).**
Il *VoIP* permette di trasportare comunicazioni vocali attraverso reti *IP*, come Internet o una rete locale, invece di utilizzare esclusivamente la tradizionale rete telefonica (Alvestrand, 2021).
La voce viene:
Acquisita → Digitalizzata → Codificata → Trasportata in pacchetti → Decodificata → Riprodotta.

A differenza dello streaming musicale, il *VoIP* è normalmente bidirezionale e interattivo.
Entrambi gli utenti stanno contemporaneamente:
Trasmettendo audio ↔ Ricevendo audio.

Per questo motivo l'obiettivo principale non è accumulare molti secondi di contenuto nel buffer, ma mantenere la latenza il più bassa possibile.

Nel *VoIP* il processo è questo:

- Il segnale vocale acquisito dal microfono viene suddiviso in piccoli intervalli temporali e compresso attraverso un codec audio. Un codec molto utilizzato nelle comunicazioni Internet moderne è Opus, particolarmente adatto alla voce e alle comunicazioni in tempo reale (Valin et al., 2012).
- I dati vengono quindi trasportati attraverso la rete sotto forma di pacchetti.
- Il ricevitore deve ricostruire la sequenza temporale: Pacchetto 1 → Pacchetto 2 → Pacchetto 3 → Pacchetto 4.

Poiché Internet può introdurre ritardi differenti tra i pacchetti, il *VoIP* utilizza generalmente un jitter buffer.

Il jitter buffer cerca di compensare le irregolarità nei tempi di arrivo mantenendo contemporaneamente la latenza sufficientemente bassa per permettere una conversazione naturale (W3C, 2025).

Una canzone on demand può utilizzare diversi secondi di buffer.
Se la rete rallenta temporaneamente il buffer fa continuare la riproduzione.
In una telefonata questo approccio sarebbe problematico.
Immaginiamo un buffer di cinque secondi:
Utente A parla → 5 secondi → Utente B sente.

Una conversazione diventerebbe praticamente impossibile.
Per questo il *VoIP* è particolarmente sensibile a:

- Latenza.
- Jitter.
- Packet loss.
- Congestione della rete.

Il *VoIP* offre diversi vantaggi rispetto ai sistemi telefonici tradizionali:

- Integrazione con Internet: voce, video, messaggi e condivisione dello schermo possono essere integrati nella stessa applicazione.
- Flessibilità: è possibile comunicare utilizzando computer, smartphone e altri dispositivi connessi.
- Scalabilità: i sistemi aziendali possono aggiungere facilmente utenti e servizi.
- Funzionalità avanzate: conferenze, registrazioni, inoltro, presenza e integrazione con software aziendali possono essere gestiti digitalmente.

Tra i limiti troviamo invece:

- Dipendenza dalla rete: una connessione instabile può compromettere la qualità della comunicazione.
- Sensibilità a latenza e jitter: ritardi elevati rendono la conversazione innaturale.
- Dipendenza dall'alimentazione e dall'infrastruttura locale: dispositivi e apparati di rete devono essere alimentati per funzionare.
- Possibile perdita di pacchetti: una parte dell'audio può arrivare troppo tardi o non arrivare affatto.

Tra i servizi che utilizzano il *VoIP* troviamo Microsoft Teams, WhatsApp, Zoom, Discord e Cisco Webex.

Quindi in conclusione:

- Audio on demand → priorità alla continuità e alla qualità.
- Audio live broadcast → priorità alla continuità con latenza contenuta.
- *VoIP* → priorità alla latenza molto bassa e all'interattività.

Video

Lo streaming video è una tecnologia che permette di guardare contenuti audiovisivi attraverso una rete senza dover attendere il download completo del file prima della riproduzione e di avere il file completo in locale.
Il contenuto viene ricevuto progressivamente dal dispositivo, conservato temporaneamente nel buffer, decodificato e riprodotto dal player (Pantos & May, 2017).

Nei sistemi moderni, come *HLS* e *DASH*, il video viene normalmente suddiviso in segmenti.

Lo streaming video può essere utilizzato sia per contenuti on demand sia per contenuti live.

- Nel primo caso il contenuto è già stato prodotto e l'utente decide quando riprodurlo.
- Nel secondo caso il contenuto viene acquisito, codificato e distribuito mentre l'evento è ancora in corso.

Lo streaming video può essere classificato in base a questi due assi:

- Modalità di fruizione: On demand / Live / Real-time interactive.
- Modello di distribuzione: *OTT* / *IPTV*.

Tipologia	**Caratteristiche**	**Esempi**
OTT + *VOD*	Contenuti già disponibili distribuiti attraverso Internet.	Netflix, Disney+.
OTT + Live	Contenuti trasmessi in diretta attraverso Internet.	YouTube Live, Twitch.
Real-time interactive + Live	Comunicazione audiovisiva bidirezionale a bassa latenza attraverso Internet.	Google Meet, Zoom.
IPTV + Live	Canali live distribuiti attraverso una rete *IP* gestita dall'operatore.	Canali televisivi del servizio IPTV.
IPTV + VOD	Contenuti on demand distribuiti attraverso una rete *IP* gestita dall'operatore.	Catalogo VOD dell'operatore.

La maggior parte dei servizi di streaming video utilizzati sul Web appartiene alla categoria *OTT* (Over The Top).
Un servizio *OTT* distribuisce i propri contenuti attraverso Internet utilizzando l'infrastruttura di rete esistente.
Il provider del contenuto non deve necessariamente controllare la rete attraverso la quale passa il video.
Un servizio come Netflix, per esempio, distribuisce il proprio contenuto attraverso reti *IP* e *CDN*, ma l'utente può accedervi utilizzando diversi operatori Internet (Netflix, s.d.).
Il servizio è quindi "sopra" l'infrastruttura di rete.

***L'IPTV* (Internet Protocol Television)**, invece, utilizza anch'essa reti basate sul protocollo *IP*, ma segue un modello differente.
In un sistema *IPTV*, il servizio televisivo viene normalmente fornito attraverso una rete gestita dall'operatore.
Questo significa che lo stesso soggetto che fornisce il servizio può controllare maggiormente la quality of service della rete (ITU-T, 2009).

Il percorso può essere semplificato così:
Broadcaster → Rete IPTV dell'operatore → Set-top box / Smart TV → Utente.

A differenza dei normali servizi *OTT*, quindi, il contenuto non attraversa necessariamente Internet pubblico nello stesso modo.
L'operatore può gestire direttamente la rete e garantire determinati parametri di qualità.

Una caratteristica interessante dell'*IPTV* è la possibilità di utilizzare il multicast per i canali televisivi live.
Nel normale streaming unicast ogni utente riceve un flusso separato.
Nel multicast una stessa trasmissione può essere distribuita attraverso la rete a più utenti interessati.
Questo è particolarmente efficiente per un canale televisivo visto contemporaneamente da moltissime persone.
Il Video on Demand *IPTV*,invece, viene normalmente servito attraverso connessioni individuali, perché utenti differenti possono guardare contenuti differenti e in momenti differenti (ITU-T, 2009).

Quindi *IPTV* e *OTT* utilizzano entrambe tecnologie *IP*, ma differiscono soprattutto per il controllo dell'infrastruttura.

- *IPTV*: Rete gestita dall'operatore → maggiore controllo sulla qualità.
- *OTT*: Internet pubblico → maggiore indipendenza dall'operatore.

Un altro caso particolare è quello delle videocomunicazioni, come videoconferenze e videochiamate.
Qui l'obiettivo cambia ancora, come abbiamo visto nell'audio.
Netflix può tollerare diversi secondi di buffer.
Durante una videochiamata, invece, un buffer di diversi secondi renderebbe impossibile conversare.
Per questo sistemi come *WebRTC* privilegiano la bassa latenza e quindi un buffer ridotto.
In questo, quindi, diventa importante gestire la latenza, il jitter e la perdita di pacchetti (Alvestrand, 2021).

Applicazioni interattive

Lo streaming di applicazioni interattive permette di eseguire un software su un computer remoto e di trasmetterne all'utente soprattutto l'output audiovisivo, mentre gli input dell'utente vengono inviati nella direzione opposta (Amazon Web Services, s.d.-b).
È il principio utilizzato in sistemi come:

- Cloud gaming.
- Desktop remoto.
- Application streaming.
- Software grafici o professionali eseguiti nel cloud.

In questo modello, il dispositivo dell'utente non deve eseguire localmente tutta l'elaborazione richiesta dall'applicazione.
Il processo fondamentale è:
Input utente → Server remoto → Elaborazione → Rendering → Codifica video/audio → Rete → Client → Decodifica → Visualizzazione.

Contemporaneamente il client continua a inviare nuovi input al server remoto tramite dispositivi di input come mouse, tastiera, controller, ...



L'applicazione viene eseguita su un server remoto, spesso dotato di *CPU* e *GPU* ad alte prestazioni.
Nel caso del cloud gaming, questo è il server che esegue e renderizza la sessione del giocatore. Qui lo chiameremo game server, senza confonderlo con un normale server multiplayer che gestisce lo stato della partita mentre il rendering avviene sui dispositivi dei giocatori (Amazon Web Services, s.d.-a).
Il server viene generalmente scelto in una regione o struttura relativamente vicina all'utente per ridurre la latenza.
In particolare il server serve a:

- Eseguire il gioco.
- Calcolare fisica e logica.
- Renderizzare i frame.
- Generare l'audio.

In questo scenario, il dispositivo dell'utente deve quindi principalmente:
- Ricevere il flusso.
- Decodificarlo.
- Visualizzarlo.
- Raccogliere e trasmettere gli input.

Per questo anche un dispositivo relativamente poco potente può utilizzare applicazioni che richiederebbero hardware molto più performante se eseguite localmente.
Amazon, per esempio, offre WorkSpaces Applications, un servizio che consente di eseguire applicazioni desktop in *AWS* e utilizzarle attraverso browser HTML5 o client dedicati (Amazon Web Services, s.d.-b).

Ma come funziona esattamente?
Immaginiamo un videogioco.
- L'utente preme sul controller il pulsante per saltare.
- L'input viene inviato attraverso la rete al server.
- Il server riceve l'input.
- Il server aggiorna il gioco.
- Il server renderizza il nuovo frame.
- Il server codifica il frame.
- Il risultato viene quindi inviato al dispositivo.
- Il dispositivo decodifica il frame.
- Il frame viene visualizzato sul display dell'utente.

Il ciclo ricomincia immediatamente con l'input successivo.
Possiamo rappresentarlo così:
Input → Rete → Server → Elaborazione → Rendering → Encoding → Rete → Decoder → Display.



Come abbiamo visto, a differenza dello streaming video tradizionale, il contenuto non è già disponibile perché ogni frame dipende anche dalle azioni dell'utente: infatti esso viene generato sul momento.
Nel cloud gaming uno dei parametri più importanti è l'**input lag**, cioè il tempo che intercorre tra un'azione dell'utente e il momento in cui la conseguenza di quell'azione appare sullo schermo (Alhilal et al., 2022).
Non dipende soltanto dalla rete.
Il ritardo complessivo può essere composto da diversi elementi:

- Input dispositivo.
- Trasmissione verso il server.
- Elaborazione del gioco.
- Rendering.
- Codifica video.
- Trasmissione verso l'utente.
- Decodifica.
- Visualizzazione sul display.

Questo ritardo complessivo viene spesso descritto anche come i**nput-to-display latency**.
Più è elevato, più l'applicazione sembra rispondere in ritardo.
Il problema è particolarmente evidente in:

- FPS competitivi.
- Picchiaduro.
- Racing game.
- Applicazioni VR.
- Software interattivi che richiedono risposta immediata.

Non è realistico richiedere una latenza “prossima allo zero”: l’obiettivo è mantenerla sufficientemente bassa da non compromettere l’interazione.
NVIDIA, per esempio, nella guida a GeForce NOW per Windows richiede meno di 80 ms di latenza di rete verso il proprio data center e raccomanda meno di 40 ms per una migliore esperienza. Questi valori riguardano la latenza di rete, non l’intero intervallo tra input e visualizzazione (NVIDIA, 2026).

Nel video on demand un buffer relativamente grande è utile: più buffer → maggiore stabilità.
Nel cloud gaming questo approccio non è possibile.
Se accumulassimo, per esempio, cinque secondi di video, il gioco diventerebbe inutilizzabile.
Lo streaming interattivo deve quindi utilizzare buffer molto piccoli e privilegiare una bassa latenza per una risposta rapida (Alhilal et al., 2022).

Questo rende il sistema più sensibile a:

- Jitter.
- Packet loss.
- Congestione.
- Variazioni della banda disponibile.

Il sistema deve quindi trovare un equilibrio tra:
Qualità video ↔ Stabilità ↔ Latenza.

Nelle applicazioni interattive il server deve codificare ogni nuovo frame quasi immediatamente.
Non può utilizzare tempi di elaborazione lunghi come quelli utilizzabili per preparare un film on demand.
Per questo vengono spesso utilizzati encoder hardware integrati nelle *GPU*, capaci di comprimere il video con un ritardo molto contenuto (NVIDIA, s.d.).
Anche il client deve poter decodificare rapidamente il flusso ricevuto.

Nel cloud gaming il modello di distribuzione è diverso da quello utilizzato nello streaming video on demand.
Una *CDN* tradizionale è particolarmente efficace quando molti utenti richiedono gli stessi contenuti. Un segmento di un film, per esempio, può essere memorizzato nella cache di un edge server e successivamente distribuito a migliaia di utenti senza dover essere recuperato ogni volta dall’origin.

Nel cloud gaming, invece, il contenuto viene generato in tempo reale per ogni singolo utente.
Il flusso audiovisivo risultante è quindi diverso per ogni sessione.
Non avrebbe normalmente senso memorizzarlo nella cache di una *CDN*, perché gli stessi frame non verranno richiesti da altri utenti.
Per questo, nello streaming interattivo, il problema principale non è portare vicino all’utente una copia del contenuto, ma portare vicino all’utente il server che sta producendo quel contenuto (Alhilal et al., 2022).
Il percorso ideale diventa quindi:
Utente ↔ Game Server vicino all’utente.

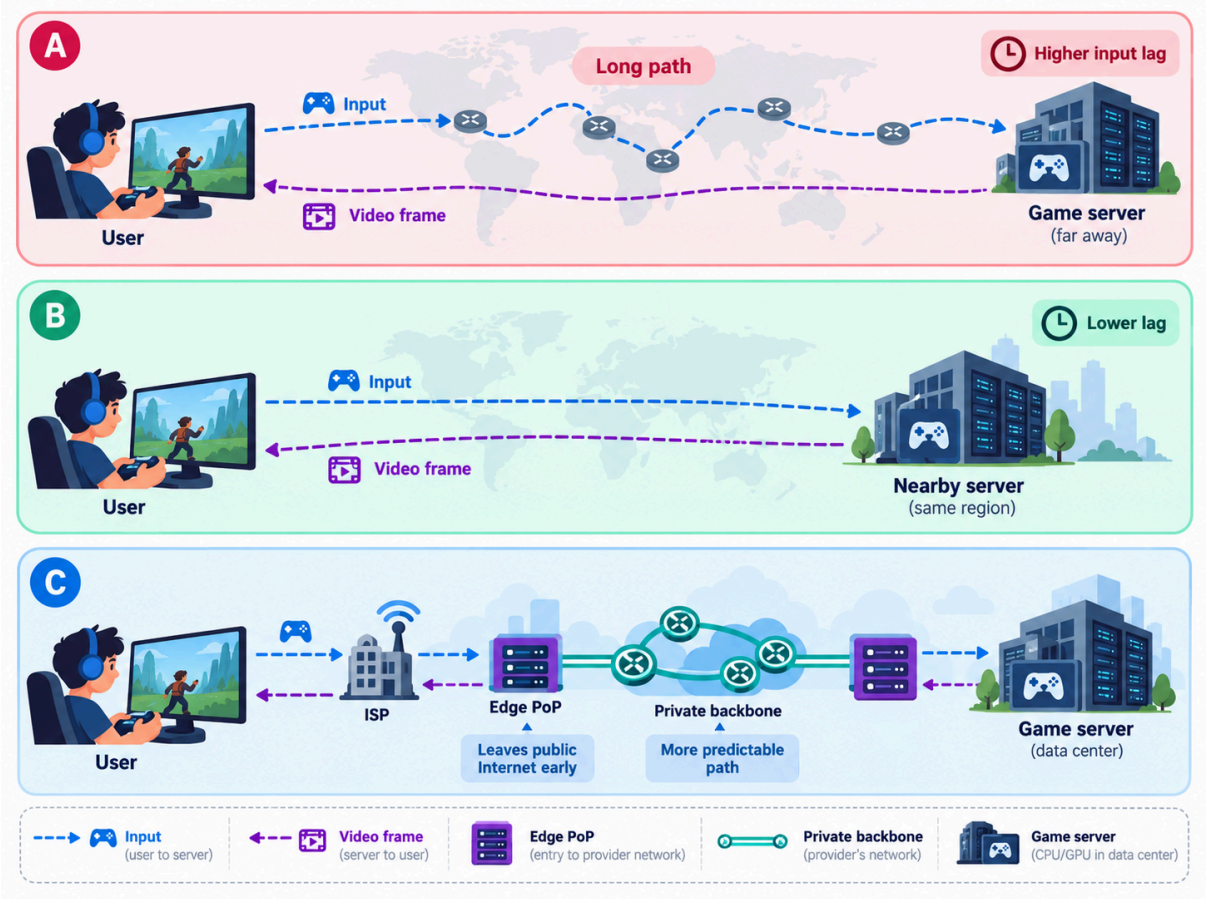
Se il server che esegue il gioco si trova molto lontano, ogni input dell'utente deve percorrere una distanza maggiore prima di essere elaborato e ogni nuovo frame deve compiere il percorso inverso per tornare al dispositivo.
Questo aumenta la latenza complessiva e quindi l'input lag.

Per ridurre questo problema, i provider di cloud gaming distribuiscono server dotati di *CPU* e *GPU* in più data center o regioni, cercando di assegnare ogni giocatore a una macchina relativamente vicina (Amazon Web Services, s.d.-c).
Il percorso può quindi essere:
Utente → Data center vicino → GPU Server → Utente.

La vicinanza geografica, tuttavia, non è l'unico elemento importante. Conta anche il percorso effettivo seguito dai pacchetti nella rete.
Una possibile soluzione consiste nell'utilizzare punti di ingresso edge e una rete privata del provider per ottimizzare il percorso. AWS Global Accelerator, per esempio, instrada il traffico applicativo attraverso la rete globale AWS. Lo schema seguente illustra questa possibilità, senza descrivere necessariamente l'architettura di ogni servizio di cloud gaming (Amazon Web Services, s.d.-d).
Per esempio:
Utente → ISP → Edge PoP → Backbone privato → Game Server.

L'Edge PoP (Point of Presence) è un nodo della rete del provider collocato relativamente vicino agli utenti. Può funzionare come punto di ingresso nella rete del provider: il traffico lascia il prima possibile Internet pubblico ed entra in un'infrastruttura controllata dal servizio. Una volta entrato nella rete del provider, il traffico può essere trasportato attraverso un backbone privato, cioè una rete ad alta capacità che collega *PoP*, data center e altri nodi dell'infrastruttura (Amazon Web Services, s.d.-d).
All'interno dell'infrastruttura il traffico viene necessariamente inoltrato attraverso router e altri apparati di rete.
Alcuni provider possono inoltre utilizzare **relay** dedicati, cioè nodi che ricevono il traffico della sessione e lo inoltrano verso il game server o verso un altro nodo della rete del provider. Il relay non esegue necessariamente il gioco: la sua funzione principale è facilitare l'inoltro del traffico lungo un percorso controllato o ottimizzato (Alvestrand, 2021).

- Utente → ISP → Edge PoP → Relay → Backbone privato → Game Server.
- Game Server → Backbone privato → Relay → Edge PoP → ISP → Utente.



È quindi importante distinguere tra *CDN* tradizionale ed edge networking.

- Una *CDN* tradizionale utilizza soprattutto: caching + distribuzione di contenuti già esistenti.
- L'infrastruttura utilizzata dal cloud gaming punta invece soprattutto a: server vicini + routing efficiente + backbone veloce + bassa latenza.

Una *CDN* può comunque essere utilizzata dal servizio di cloud gaming per distribuire elementi che sono realmente cacheabili, come file del client, immagini, pagine web, aggiornamenti o altri contenuti statici.
Il flusso interattivo della partita, invece, segue generalmente un percorso dedicato alla sessione dell'utente e non viene distribuito attraverso il normale meccanismo di caching della *CDN*.
La differenza fondamentale può quindi essere riassunta così:

- Video on demand → portare il contenuto vicino all'utente.
- Cloud gaming → portare l'elaborazione vicino all'utente.

Nel primo caso sono fondamentali *CDN* ed edge caching.
Nel secondo diventano fondamentali soprattutto la posizione del server che esegue il gioco e il percorso di rete. PoP, relay e backbone privati possono contribuire a ottimizzarlo, ma non sono componenti obbligatori in ogni architettura. L'obiettivo è ridurre il tempo necessario per completare continuamente il ciclo:
Input → Elaborazione → Rendering → Video → Utente.

Un’altra differenza fondamentale rispetto allo streaming video tradizionale è la direzione dei dati.

- Nel video on demand il traffico principale è “Server → Utente”.
- Nel cloud gaming abbiamo continuamente “Input: Utente → Server” e contemporaneamente “Server: Audio/Video → Utente”.

Quindi il sistema è realmente bidirezionale e interattivo.
La quantità di dati nelle due direzioni, però, non è uguale.

- Gli input possono occupare pochissima banda: tasto premuto, posizione del mouse, movimento del controller, ..
- Il flusso video, invece, può richiedere molti megabit al secondo.

Abbiamo quindi normalmente:

- Uplink → pochi dati ma molto sensibili alla latenza.
- Downlink → molti dati audio/video e anch’essi sensibili alla latenza.



Il cloud gaming è una delle applicazioni più importanti dello streaming interattivo.
Esempi attuali sono NVIDIA GeForce NOW e Xbox Cloud Gaming.
NVIDIA pubblica requisiti specifici di banda e latenza per GeForce NOW, mentre Xbox permette di avviare giochi in streaming direttamente dal proprio servizio cloud (NVIDIA, 2026; Microsoft, s.d.).

Lo stesso principio può essere applicato anche a software non videoludici.
Un programma professionale può essere installato ed eseguito su un server remoto mentre l’utente interagisce attraverso un browser o un client.
Il percorso rimane:
Input utente → Applicazione remota → Rendering → Stream → Utente.

Questo permette, per esempio, di utilizzare software complessi senza eseguirli direttamente sull’hardware locale.

Non significa però necessariamente che l'utente non debba installare nulla: alcuni servizi funzionano interamente nel browser, altri utilizzano un client dedicato.
Amazon WorkSpaces Applications, per esempio, permette l'accesso tramite browser HTML5 oppure tramite client dedicato (Amazon Web Services, s.d.-b).



Materiale di supporto

Esercizi

Esercizio 1 — Progettare un servizio di streaming audio
Una società vuole realizzare una piattaforma dedicata a contenuti audio. Il servizio dovrà offrire contemporaneamente:

- Album musicali già registrati.
- Una radio online con programmazione continua.
- Conversazioni vocali in tempo reale tra piccoli gruppi di utenti.

Il team propone di utilizzare la stessa configurazione tecnica per tutti e tre i servizi.

Analizza criticamente la proposta.

- Classifica i tre servizi distinguendo tra:
    - Audio on demand.
    - Audio live.
    - Comunicazione vocale interattiva.
- Per ciascun servizio stabilisci quale importanza assumono:
    - Qualità audio.
    - Continuità.
    - Latenza.
    - Buffering.
    - Jitter.
    - Packet loss.
- Spiega perché un buffer di diversi secondi potrebbe essere utile durante l'ascolto di un album ma problematico durante una conversazione.
- Per la comunicazione vocale ricostruisci il percorso: Voce → acquisizione → digitalizzazione → codifica → pacchetti → rete → decodifica → riproduzione.
- Spiega quale funzione può svolgere un jitter buffer.
- Una connessione consegna tutti i pacchetti audio, ma alcuni arrivano con forte ritardo e intervalli irregolari. Quale problema si sta verificando?
- Una seconda connessione mantiene tempi di arrivo regolari, ma perde occasionalmente alcuni pacchetti. È lo stesso problema? Motiva.
- Perché il bitrate richiesto dallo streaming audio è generalmente inferiore rispetto allo streaming video e quali vantaggi ne derivano?
- Immagina che la piattaforma disponga dello stesso brano in tre versioni: 320 kbit/s, 160 kbit/s e 64 kbit/s. Spiega come potrebbe utilizzare queste versioni quando cambiano le condizioni della rete.
- Concludi completando:
    - Audio on demand → priorità a …
    - Audio live → priorità a …
    - *VoIP* → priorità a …

Esercizio 2 — *OTT*, *IPTV* o comunicazione in tempo reale?

Una società vuole distribuire un nuovo canale dedicato al cinema attraverso tre servizi differenti.

- Servizio A: gli utenti scelgono film già presenti in catalogo utilizzando qualsiasi connessione Internet.
- Servizio B: un operatore di telecomunicazioni distribuisce un canale cinematografico continuo ai propri abbonati attraverso la propria rete IP gestita.
- Servizio C: dopo alcuni film vengono organizzati incontri video nei quali regista e spettatori possono parlarsi in tempo reale.

Analizza i tre casi.

- Classifica ciascun servizio secondo la modalità di fruizione:
  - On demand.
  - Live.
  - Real-time interactive.
- Classifica A e B secondo il modello di distribuzione: *OTT* o *IPTV*.
- Spiega perché *OTT*/*IPTV* e on demand/live descrivono due aspetti differenti dello stesso servizio.
- Quale elemento distingue principalmente *OTT* e *IPTV* dal punto di vista dell'infrastruttura?
- Nel Servizio B l'operatore controlla la rete di distribuzione. Quali vantaggi potrebbe ottenere nella gestione della Quality of Service.
- Se 100.000 utenti stanno guardando contemporaneamente lo stesso canale live all'interno della rete IPTV, perché il multicast potrebbe risultare interessante?
- Perché lo stesso meccanismo sarebbe meno adatto a un catalogo VOD nel quale ogni spettatore sceglie un film e un punto temporale differente?
- Nel Servizio C, perché un buffer di 10 secondi sarebbe inaccettabile?
- Quali problemi di rete diventano particolarmente importanti nel Servizio C?
- Confronta le priorità dei tre sistemi:

Servizio	Continuità	Qualità latenza	Controllo della rete
A			
B			
C			

- Immagina che il Servizio A venga distribuito attraverso diversi ISP. Il provider del contenuto controlla completamente il percorso di rete fino all'utente? Motiva.
- Il fatto che *IPTV* utilizzi il protocollo IP significa che sia semplicemente un altro nome per *OTT*? Spiega.
- Concludi formulando con parole tue la differenza tra:
  - Modalità di fruizione → quando e come il contenuto viene consumato.
  - Modello di distribuzione → attraverso quale infrastruttura arriva all'utente.

Esercizio 3 — Un software 3D eseguito in streaming

Una scuola vuole permettere agli studenti di utilizzare un software di modellazione 3D molto pesante anche sui computer meno potenti del laboratorio.
Il software viene quindi eseguito su potenti server remoti dotati di *GPU*.
Quando uno studente muove il mouse:

- Il movimento viene inviato attraverso la rete.
- Il server modifica la scena 3D.
- Produce il nuovo frame.
- Codifica l'immagine.
- La invia al computer dello studente.
- Il client la decodifica e la visualizza.

Rispondi alle seguenti domande

- Spiega perché questo sistema può essere considerato streaming di un'applicazione interattiva.
- Individua ciò che viene realmente eseguito:
    - Sul server.
    - Sul dispositivo dello studente.
- Ricostruisci il ciclo completo: Input → rete → server → elaborazione → rendering → encoding → rete → decoder → display.
- Spiega perché il contenuto video non può essere preparato anticipatamente come avviene con un film on demand.
- Ogni studente osserva una scena differente e compie azioni differenti. Perché i flussi prodotti sono quindi specifici per ciascuna sessione?
- Definisci input lag.
- Individua almeno sei componenti che possono contribuire alla input-to-display latency.
- Perché non sarebbe possibile utilizzare un buffer di diversi secondi come in un normale servizio *VOD*?
- Il laboratorio è a Milano ma il server remoto si trova in Australia. Anche con un'ottima connessione a banda larga l'esperienza potrebbe essere problematica. Perché?
- Quale strategia sarebbe più utile, motiva la risposta:
    - Conservare copie dei frame in una CDN tradizionale.
    - Eseguire il software su un server più vicino agli studenti.
- Spiega la differenza tra:
    - Video on demand → portare il contenuto vicino all'utente.
    - Streaming interattivo → portare l'elaborazione vicino all'utente.
- Il sistema utilizza: Studente → ISP → Edge PoP → backbone privato → server GPU. Spiega la funzione generale di:
    - Edge PoP.
    - Backbone privato.
    - Server GPU.
- Perché una *CDN* tradizionale può comunque essere utile per distribuire aggiornamenti, immagini o file dell'applicazione, anche se non è particolarmente utile per mettere in cache il flusso interattivo?
- Analizza la direzione del traffico:
    - Uplink: quali dati vengono inviati?
    - Downlink: quali dati vengono ricevuti?
    - Quale dei due utilizza normalmente più banda?

    - Perché entrambi rimangono sensibili alla latenza?
- Concludi individuando il compromesso principale: Qualità video ↔ stabilità ↔ latenza.

Bibliografia

Alhilal, A., Braud, T., Han, B., & Hui, P. (2022). *Nebula: Reliable Low-latency Video Transmission for Mobile Cloud Gaming*.

Alvestrand, H. (2021). *Overview: Real-Time Protocols for Browser-Based Applications. RFC 8825*.

Amazon Web Services. (s.d.-a). *Amazon GameLift Streams FAQs*.

Amazon Web Services. (s.d.-b). *Amazon WorkSpaces applications FAQs*.

Amazon Web Services. (s.d.-c). *Manage streaming with an Amazon GameLift Streams stream group*.

Amazon Web Services. (s.d.-d). *What is AWS Global Accelerator?*.

Apple. (s.d.). *Podcast RSS feed requirements*.

ITU-T. (2009). *Requirements for the support of IPTV services. Raccomandazione Y.1901*.

Microsoft. (s.d.). *Xbox Cloud Gaming*.

Netflix. (s.d.). *Open Connect Overview*.

NVIDIA. (2026). *How can I set up my Windows PC with my GeForce NOW Ultimate or Performance Membership?*.

NVIDIA. (s.d.). *NVIDIA Video Codec SDK*.

Pantos, R., & May, W. (2017). *HTTP Live Streaming. RFC 8216*.

Valin, J.-M., Vos, K., & Terriberry, T. (2012). *Definition of the Opus Audio Codec. RFC 6716*.

W3C. (2025). *Identifiers for WebRTC's Statistics API. Candidate Recommendation Draft, 25 settembre 2025*.

Approfondimenti

- James F. Kurose e Keith W. Ross, Reti di calcolatori e Internet. Un approccio top-down, 8ª edizione, Pearson, 2022 — Un manuale universitario dedicato al funzionamento delle reti Internet e delle applicazioni distribuite. Gli autori approfondiscono i protocolli di comunicazione, le applicazioni multimediali, la trasmissione audio-video e i meccanismi attraverso cui le reti gestiscono traffico con requisiti differenti. Il volume è particolarmente pertinente alla dispensa perché fornisce le basi per comprendere come servizi di streaming musicale, videocomunicazione e applicazioni interattive possano condividere l'infrastruttura Internet pur richiedendo strategie differenti per latenza, affidabilità e qualità della

trasmissione. Permette inoltre di approfondire i principi che distinguono le comunicazioni in tempo reale dalla distribuzione di contenuti preregistrati.

- Cloudflare, Cos'è il Voice over Internet Protocol (VoIP)?, Cloudflare Learning Center — Un approfondimento sul funzionamento delle comunicazioni vocali attraverso le reti IP e sulle differenze rispetto alla telefonia tradizionale. L'articolo descrive come la voce venga trasformata in dati digitali e trasportata attraverso Internet, illustrando alcuni dei protocolli e delle condizioni tecniche necessarie per mantenere una conversazione in tempo reale. È particolarmente pertinente alla sezione della dispensa dedicata allo streaming audio, perché permette di approfondire la distinzione tra ascolto di contenuti preregistrati e comunicazioni vocali bidirezionali. Esamina inoltre perché latenza, jitter, perdita di pacchetti e affidabilità della connessione assumano un'importanza particolare nei servizi VoIP.
- Amazon Web Services, In cosa consiste il trunking SIP?, AWS — Una spiegazione del funzionamento del Session Initiation Protocol e del suo utilizzo nelle infrastrutture di telefonia basate su IP. L'articolo chiarisce la distinzione tra VoIP, inteso come insieme di tecnologie per trasportare comunicazioni vocali attraverso reti IP, e SIP, utilizzato per stabilire e gestire sessioni di comunicazione. È particolarmente pertinente alla trattazione del VoIP presente nella dispensa, perché introduce il problema della gestione delle chiamate oltre al semplice trasporto dei pacchetti audio. Permette inoltre di comprendere come la comunicazione vocale digitale possa integrarsi con sistemi telefonici e infrastrutture aziendali, estendendo il quadro oltre le applicazioni di messaggistica e videoconferenza.
- Cloudflare, Cos'è il live streaming? Come funziona il live streaming, Cloudflare Learning Center — Un approfondimento sulle tecnologie utilizzate per distribuire contenuti audiovisivi mentre vengono prodotti. L'articolo esamina i processi di acquisizione, codifica, segmentazione, distribuzione e riproduzione che permettono di trasmettere un evento in diretta a numerosi spettatori. È particolarmente pertinente alla sezione della dispensa dedicata ai servizi video, perché permette di distinguere le caratteristiche delle trasmissioni live dalla distribuzione di contenuti on demand. Aiuta inoltre a comprendere perché uno streaming destinato a un grande pubblico abbia esigenze differenti rispetto a una videoconferenza, nella quale gli utenti devono poter comunicare reciprocamente con ritardi molto contenuti.
- Cloudflare, Cos'è HTTP Live Streaming? HLS Streaming, Cloudflare Learning Center — Un approfondimento su HTTP Live Streaming, una delle tecnologie più utilizzate per distribuire contenuti audiovisivi attraverso Internet. L'articolo descrive il funzionamento della segmentazione, delle playlist e dello streaming adattivo, spiegando come un player possa ricevere progressivamente il contenuto e modificarne la qualità in relazione alle condizioni della rete. È particolarmente pertinente alla distinzione tra servizi video OTT e comunicazioni interattive in tempo reale presente nella dispensa. Permette di comprendere perché le piattaforme di video on demand e molte trasmissioni live possano utilizzare infrastrutture HTTP e CDN, mentre applicazioni che richiedono una conversazione immediata adottano spesso tecnologie differenti.
- Cloudflare, Che cos'è un server TURN?, Cloudflare Learning Center — Un approfondimento sulle comunicazioni audiovisive in tempo reale basate su WebRTC e sui meccanismi che permettono di stabilire connessioni tra dispositivi attraverso Internet. L'articolo introduce il ruolo di STUN e TURN nel superamento delle difficoltà provocate da NAT e firewall, mostrando perché una comunicazione diretta tra due

dispositivi non sia sempre possibile. È particolarmente pertinente alle sezioni della dispensa dedicate alle videochiamate e alle comunicazioni vocali interattive. Permette di comprendere come l'architettura di una videoconferenza differisca da quella di un servizio di streaming video tradizionale e perché i sistemi in tempo reale possano richiedere server intermedi anche quando privilegiano connessioni dirette tra partecipanti.

- NVIDIA, Requisiti di sistema per GeForce NOW, Documentazione NVIDIA — Una documentazione tecnica sui requisiti hardware, di rete e di latenza necessari per utilizzare un servizio commerciale di cloud gaming. NVIDIA specifica le condizioni di connessione richieste per differenti risoluzioni e frequenze di aggiornamento, evidenziando l'importanza della latenza verso i propri data center. Il documento è particolarmente pertinente alla sezione della dispensa dedicata allo streaming interattivo e all'input lag. Permette di osservare concretamente come la qualità dell'esperienza dipenda non soltanto dalla banda disponibile, ma anche dal tempo necessario per inviare gli input al server, elaborare il gioco e restituire i frame al dispositivo dell'utente.
- Amazon Web Services, Che cos'è Amazon WorkSpaces Applications?, Documentazione AWS — Una documentazione tecnica dedicata allo streaming di applicazioni desktop eseguite su infrastrutture cloud. Il servizio permette agli utenti di utilizzare applicazioni remote attraverso un browser o un client dedicato, mentre l'elaborazione principale viene effettuata sui server AWS. È particolarmente pertinente alla sezione della dispensa dedicata all'application streaming, perché presenta un caso concreto nel quale l'utente interagisce con un software senza eseguirne localmente tutte le operazioni. Permette inoltre di comprendere il rapporto tra elaborazione remota, distribuzione dell'output, trasmissione degli input e capacità dell'infrastruttura di gestire contemporaneamente più sessioni individuali.
- Microsoft, Cos'è Desktop virtuale Azure?, Microsoft Learn — Una panoramica delle tecnologie che consentono di eseguire desktop e applicazioni Windows su infrastrutture remote, rendendoli accessibili attraverso dispositivi differenti. La documentazione descrive i principi della virtualizzazione dei desktop, della distribuzione delle applicazioni e della gestione centralizzata delle risorse. È particolarmente pertinente alla distinzione sviluppata nella dispensa tra streaming di contenuti audiovisivi e streaming di applicazioni interattive. Permette di comprendere come il principio dell'elaborazione remota possa essere applicato non soltanto ai videogiochi, ma anche a software professionali, ambienti di lavoro e applicazioni che richiedono risorse hardware superiori a quelle disponibili sul dispositivo dell'utente.

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.

