

Università degli Studi di Roma "Tor Vergata"  
Laurea in Informatica

Sistemi Operativi e Reti  
(modulo Reti)  
a.a. 2024/2025

# Livello di applicazione (parte4)

dr. Manuel Fiorelli

[manuel.fiorelli@uniroma2.it](mailto:manuel.fiorelli@uniroma2.it)

<https://art.uniroma2.it/fiorelli>

Basate sulle slide del libro di testo:

[https://gaia.cs.umass.edu/kurose\\_ross/ppt.php](https://gaia.cs.umass.edu/kurose_ross/ppt.php)

# Livello di applicazione: panoramica

- Principi delle applicazioni di rete
- Web e HTTP
- E-mail, SMTP, IMAP
- DNS: il servizio di directory di Internet
- Applicazioni P2P
- Streaming video e reti di distribuzione di contenuti
- Programmazione delle socket programming con UDP e TCP



# Streaming video e CDN: contesto

- traffico video in streaming :  
grande consumatore di larghezza  
di banda Internet
  - Netflix, YouTube, Amazon Prime: 80% del  
traffico ISP residenziale (2020)
- *sfida*: scala - come raggiungere  
~1B di utenti?
- *sfida*: eterogeneità
  - utenti diversi hanno capacità diverse (ad esempio, cablati  
o mobili; ricchi di larghezza di banda o poveri di larghezza  
di banda)
- *soluzione*: infrastruttura distribuita a livello di  
applicazione

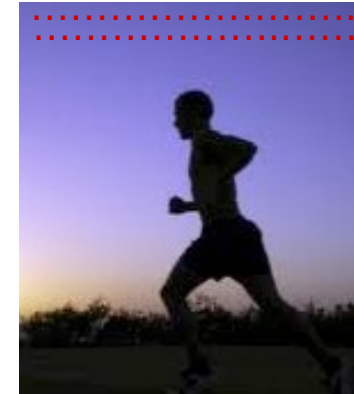


# Contenuti multimediali: video

- video: sequenza di immagini visualizzate a tasso costante (*frame rate*)
  - Esempio: 24 immagini al secondo
- immagine digitale: un array di pixel
  - ogni pixel rappresentato da bit
- codifica: utilizzare la ridondanza *all'interno* e *tra* le immagini per ridurre il numero di bit utilizzati per la codifica dell'immagine
  - spaziale (all'interno di una data immagine)
  - temporale (da un'immagine all'altra)

*esempio di codifica spaziale:*

invece di inviare N valori dello stesso colore (tutti viola), inviare solo due valori: valore del colore (viola) e numero di valori ripetuti (N)



frame *i*

*esempio di codifica temporale:*

invece di inviare il frame completo a  $i+1$ , invia solo le differenze dal frame  $i$



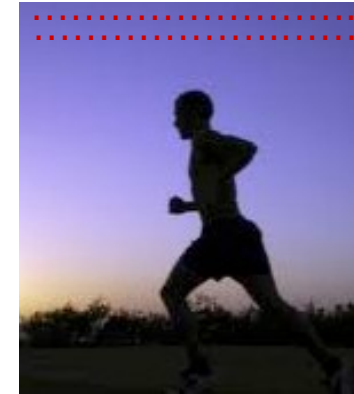
frame  $i+1$

# Multimedia: video

- **CBR: (constant bit rate):** bit rate costante
- **VBR: (variable bit rate):** bit rate cambia con la quantità di ridondanza spaziale e temporale
- **esempio:**
  - MPEG-1 (CD-ROM) 1.5 Mbps
  - MPEG-2 (DVD) 3-6 Mbps
  - MPEG-4 (spesso usato in Internet, 64kbps – 12 Mbps)

*esempio di codifica spaziale:*

invece di inviare N valori dello stesso colore (tutti viola), inviare solo due valori: valore del colore (viola) e numero di valori ripetuti (N)



frame *i*

*esempio di codifica temporale:*

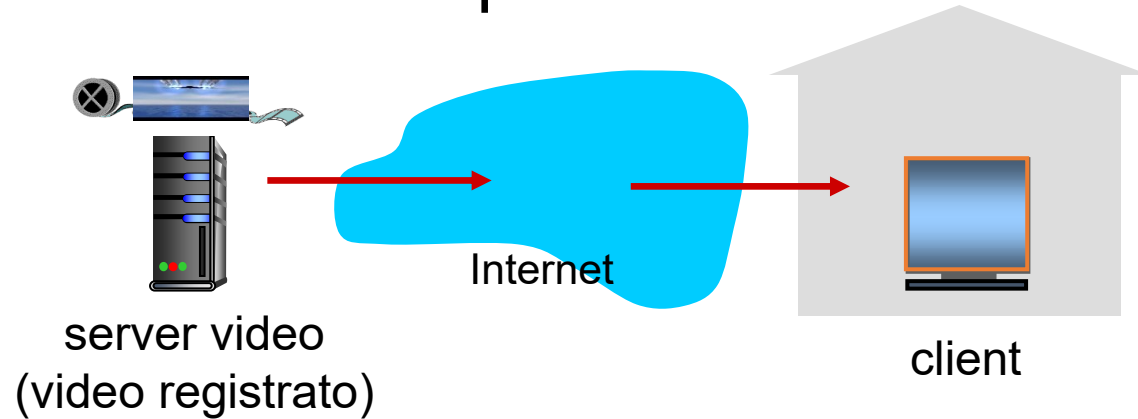
invece di inviare il frame completo a  $i+1$ , invia solo le differenze dal frame  $i$



frame *i+1*

# Streaming video di contenuti registrati

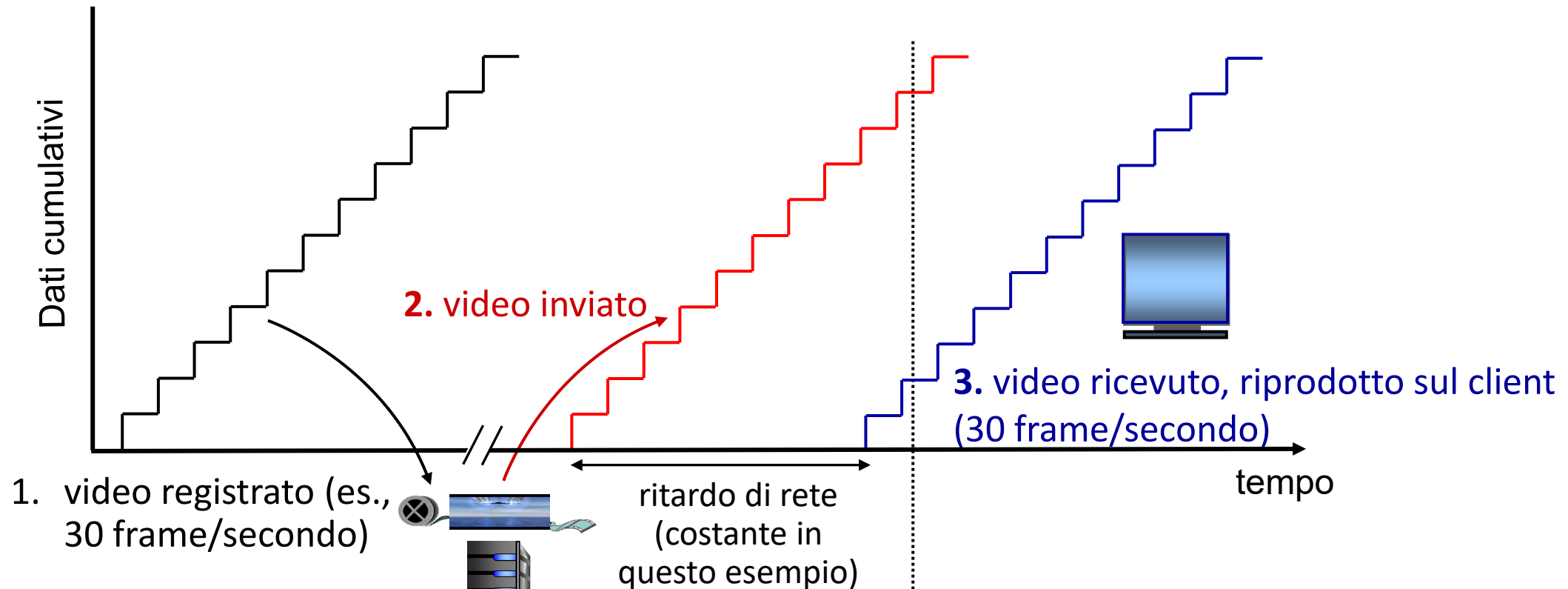
scenario semplice



Sfide principali:

- la larghezza di banda da server a client varia nel tempo, con il variare dei livelli di congestione della rete (rete residenziale, rete di accesso, nucleo della rete, server video)
- la perdita di pacchetti e i ritardi dovuti alla congestione ritardano la riproduzione o comportano una scarsa qualità video.

# Streaming video di contenuti registrati

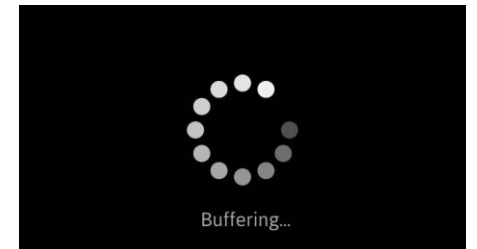


Limita il ritardo di riproduzione, evita spreco di banda se il client interrompe riproduzione

**streaming**: in questo momento, il client sta riproducendo la parte iniziale del video, mentre il server sta ancora inviando la parte successiva del video

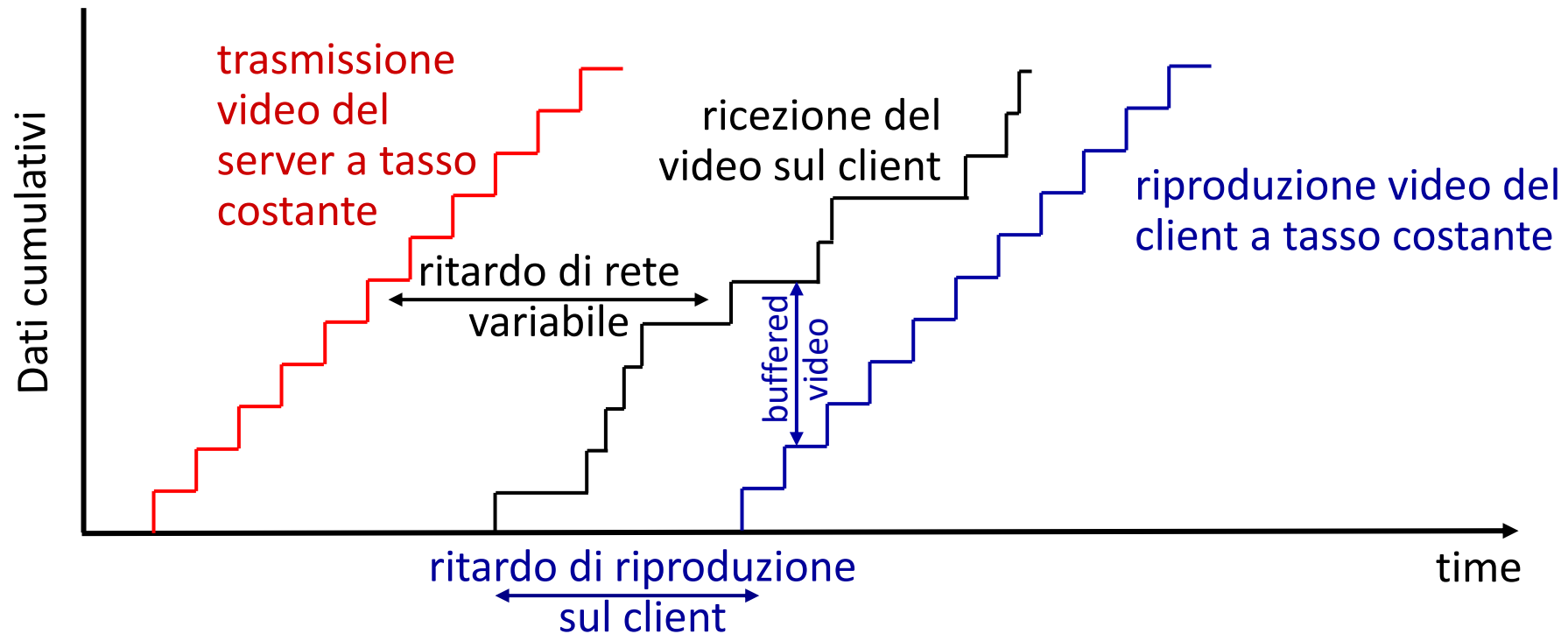
# Streaming video di contenuti registrati

- **vincolo di riproduzione continua**: quando la riproduzione inizia, dovrebbe procedere secondo i tempi di registrazione originali
  - ... ma **i ritardi di rete sono variabili** (jitter), quindi avrà bisogno di un buffer **lato client** per soddisfare i vincoli di riproduzione continua
- altre sfide:
  - interattività del client: pausa, avanzamento veloce, riavvolgimento, salti attraverso il video
  - i pacchetti video possono essere persi, ritrasmessi





# Streaming video di contenuti registrati



- ***buffering lato client e ritardo di riproduzione:*** compensare il ritardo aggiunto dalla rete, il jitter (variazione) del ritardo

# Streaming video di contenuti registrati

## ■ Streaming UDP:

- Il server invia pacchetti video in modo da eguagliare il bit rate del video stesso, traendo vantaggio dall'assenza di controllo della congestione (es. bit rate 2 Mbps e pacchetti da 8000 bit → il server invia un pacchetto ogni 4 ms)
- Buffer lato client equivalente a pochi secondi di video
- Connessione di controllo separata attraverso il quale il client può inviare comandi, quali pausa, salto, etc.
- Incapace di rispettare il vincolo di riproduzione continua, se la banda disponibile scende periodicamente sotto al bit rate del video (che è il tasso di consumo)

## ■ Streaming HTTP

- Il server trasmette alla massima velocità consentita (es. dal controllo della congestione): se la velocità di ricezione è maggiore del bit rate del video, il buffer del client continua a crescere anche durante la riproduzione (*prefetching*) fino al riempimento, dopodiché il controllo del flusso limiterà il tasso di trasmissione al tasso di consumo del client
- Può assorbire fluttuazioni del throughput, finché il throughput medio sia maggiore del bit rate
- Riga di intestazione *Range* nelle richieste per saltare

## ■ streaming dinamico adattativo su HTTP

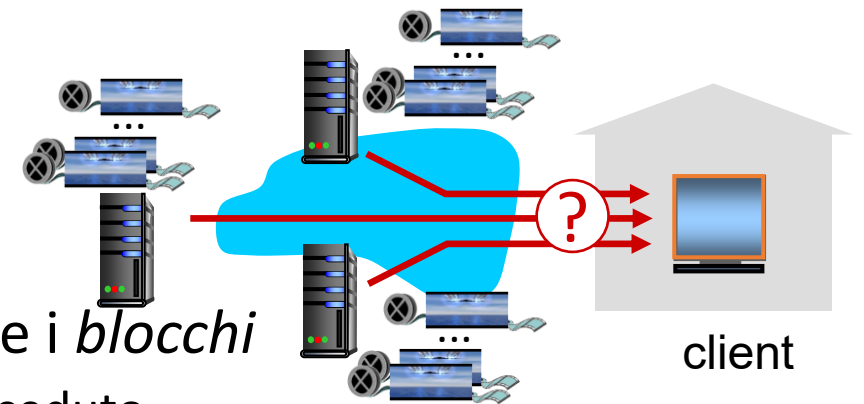
- Consente di scegliere tra versioni con livelli di qualità differenti anche durante la riproduzione

# Streaming multimediale: DASH

Dynamic, Adaptive  
Streaming over HTTP

## server:

- il video viene codificato in più *versioni* con bit rate differenti (quindi, qualità differenti)
- ogni versione è suddivisa in *blocchi* (segmenti)
- *manifest file (file manifesto)*: file XML che descrive le versioni e come richiedere i *blocchi*
  - un *blocco* può avere una propria URL oppure essere acceduto attraverso una *HTTP range request* per la stessa URL associata alla versione

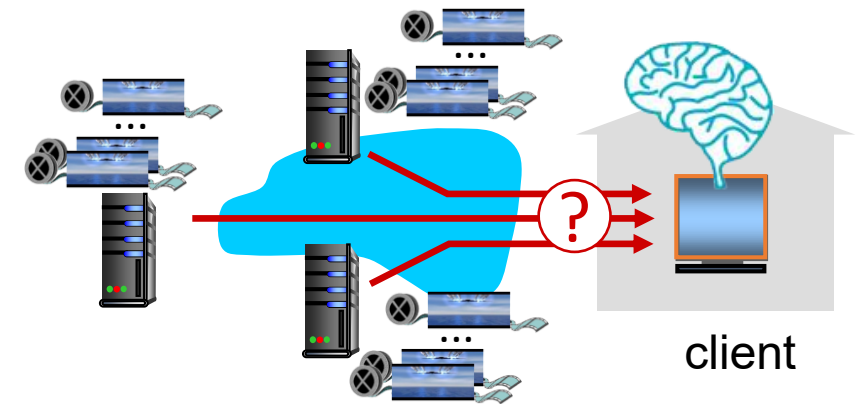


## client:

- stima periodicamente la banda da server a client
- consultando il manifesto, richiede un *blocco* alla volta
  - sceglie versione con il bit rate più alto sostenibile data la larghezza di banda corrente
  - può scegliere versioni con bit rate differenti in momenti diversi (a seconda della larghezza di banda disponibile in quel momento), e da server diversi

# Streaming multimediale: DASH

- “*intelligenza*” sul client: il client determina
  - *quando* richiedere un *blocco* (in modo che non si verifichi la starvation del buffer o l'overflow)
  - *che bit rate richiedere* (qualità più alta quando c'è più banda disponibile)
  - *dove* richiedere il *blocco* (può richiedere dal server che è "vicino" al client o ha banda larga)



Lato server: è sufficiente un server Web in grado di servire contenuti statici (file manifesto e blocchi)

Streaming video = codifica + DASH + buffering di riproduzione

# Reti per la distribuzione di contenuti - Content distribution networks (CDNs)

*sfida*: come trasmettere contenuti in streaming (selezionati tra milioni di video) a centinaia di migliaia di utenti simultanei?

- *opzione 1*: unico, enorme data center
  - singolo punto di rottura (single point of failure)
  - punto di congestione della rete
  - percorso lungo (e possibilmente congestionato) verso i clienti lontani

.... molto semplicemente: questa soluzione *non è scalabile*

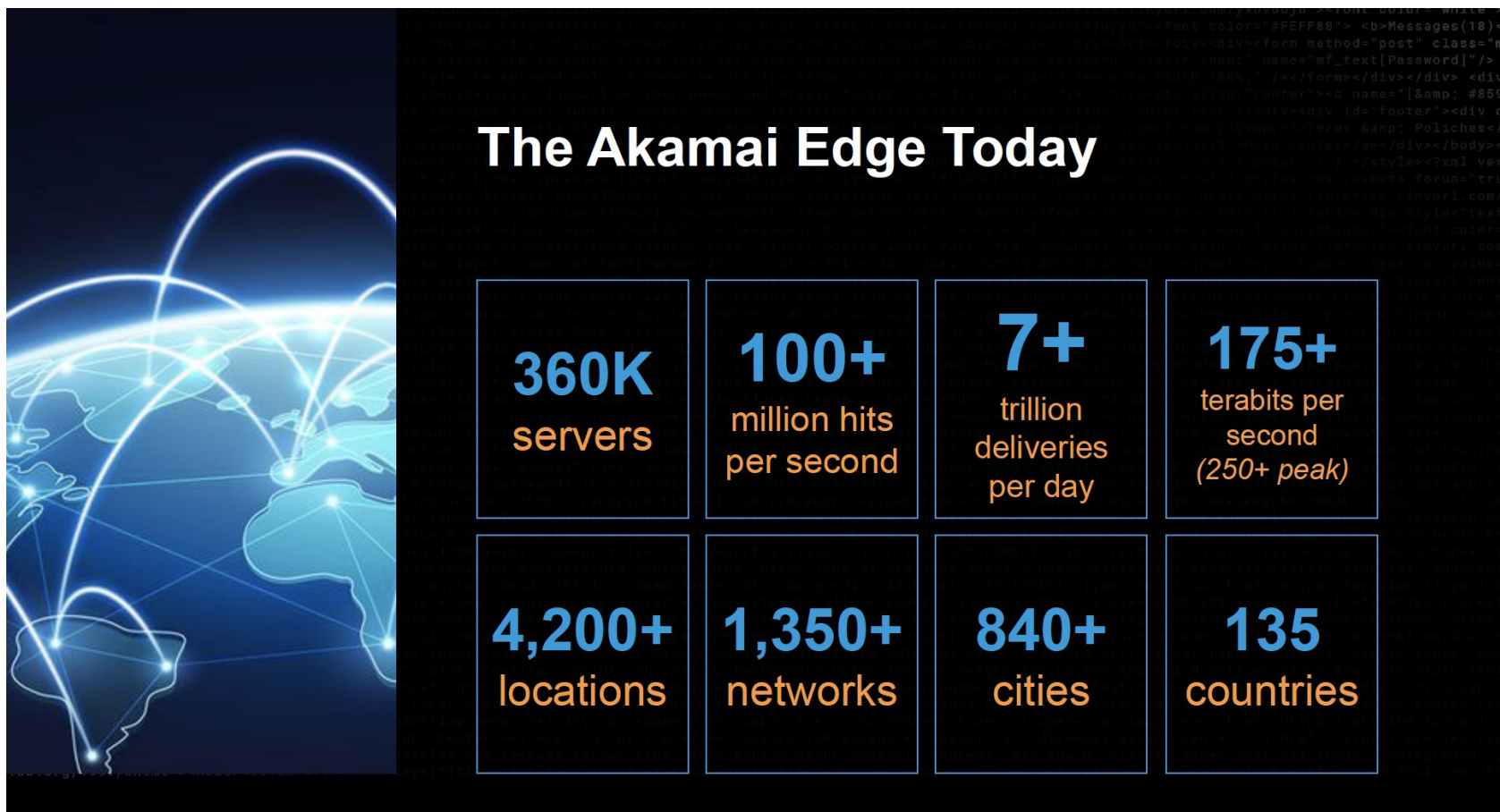
# Reti per la distribuzione di contenuti - Content distribution networks (CDNs)

*sfida:* come trasmettere contenuti in streaming (selezionati tra milioni di video) a centinaia di migliaia di utenti simultanei?

- *opzione 2:* memorizzare/servire più copie di video in più siti geograficamente distribuiti (*CDN*)
  - *enter deep (entrare in profondità):* installare server della CDN in profondità dentro molte reti di accesso
    - essere vicini agli utenti -> minore ritardi e maggior throughput, ma maggiore complessità di gestione e manutenzione
    - Akamai: 240,000 server dispiegati in > 120 paesi (2015)
  - *bring home (portare a casa):* pochi grandi cluster (decine, per esempio) in IXP vicino alle reti di accesso
    - usato da Limelight



# Akamai oggi:



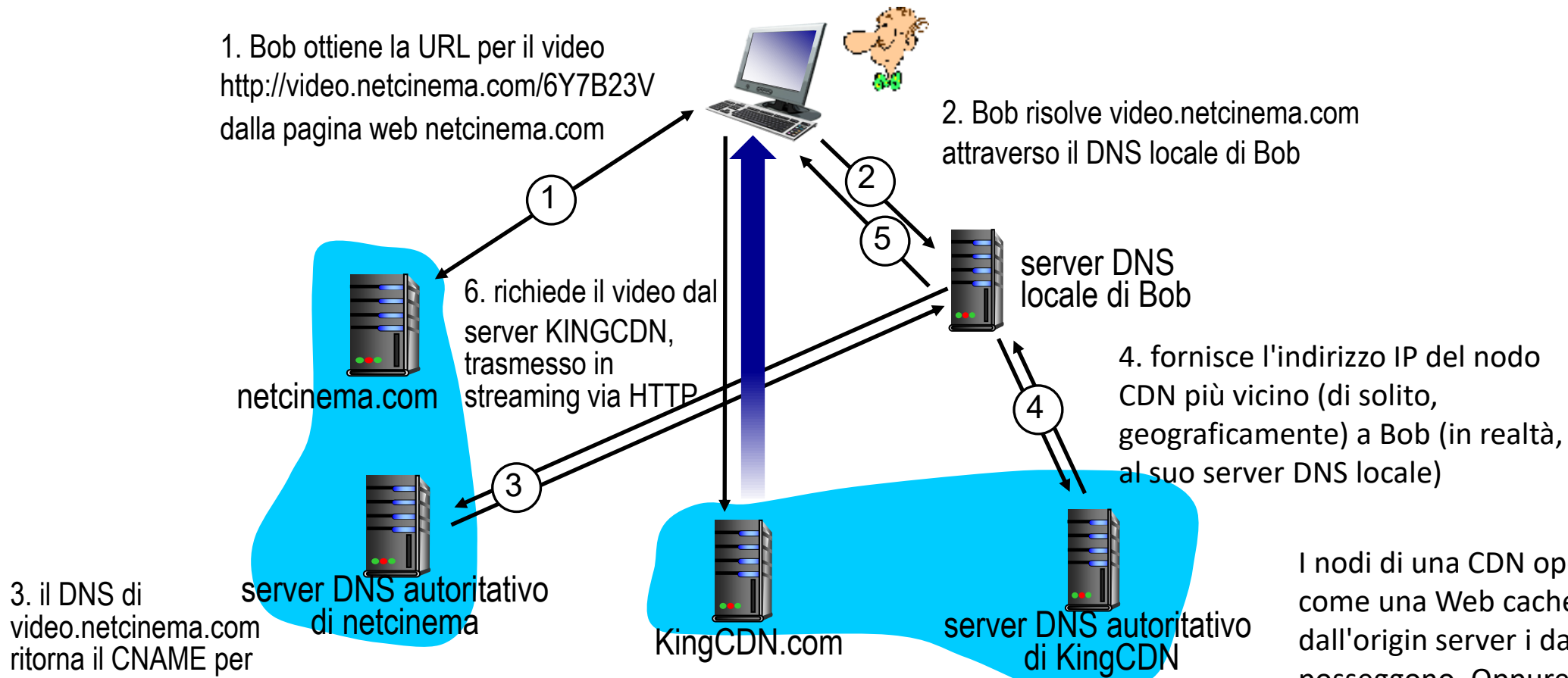
Fonte: <https://networkingchannel.eu/living-on-the-edge-for-a-quarter-century-an-akamai-retrospective-downloads/>



# Accesso ai contenuti CDN: uno sguardo da vicino

Bob (client) richiede il video `http://video.netcinema.com/6Y7B23V`

- video memorizzato sulla CDN a `http://netcinema.KingCDN.com/6Y7B23V`

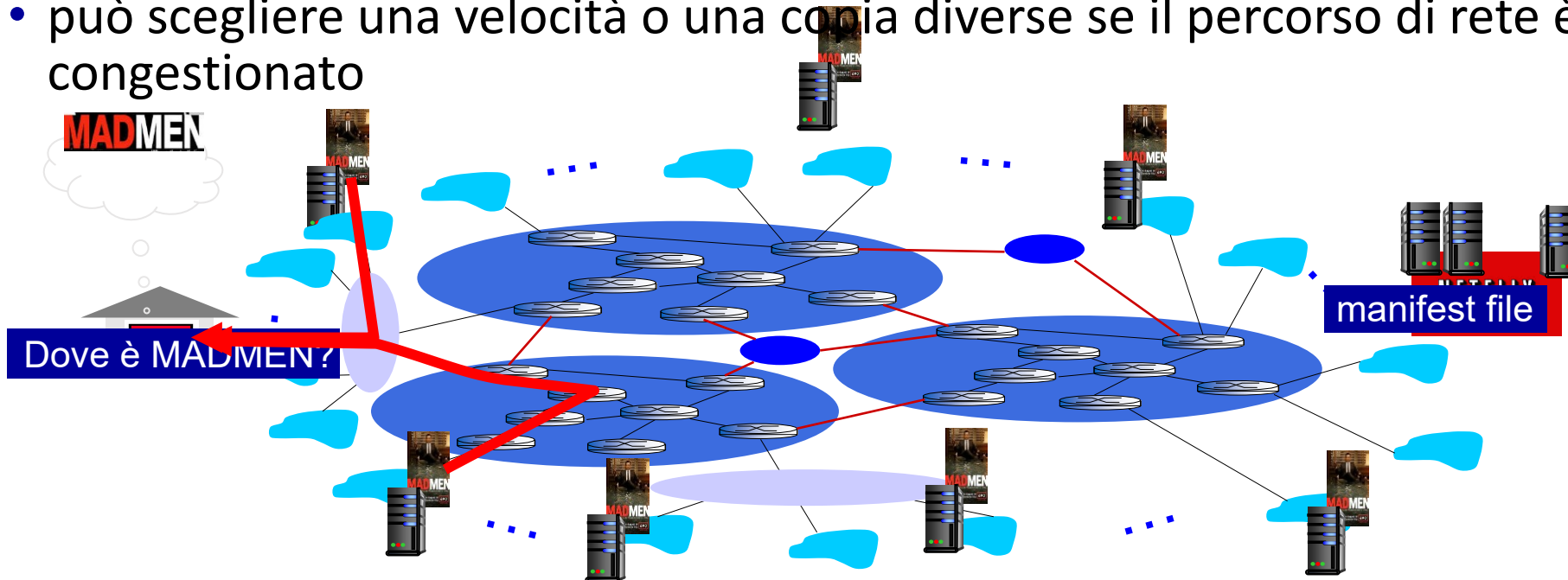


I nodi di una CDN operano solitamente come una Web cache recuperando dall'origin server i dati che non posseggono. Oppure, i dati possono essere precaricati.

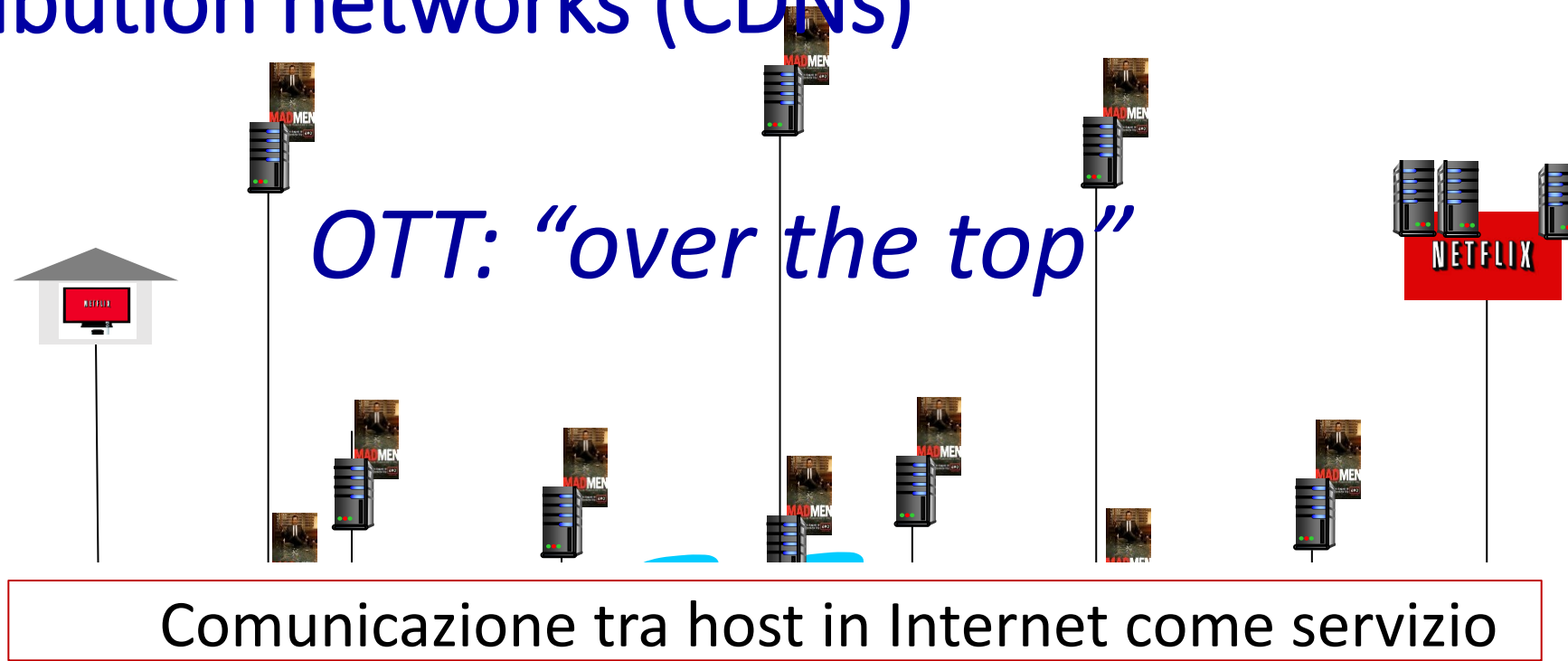


# Come funziona Netflix?

- Netflix: memorizza copie dei contenuti (ad esempio, MADMEN) sui nodi (in tutto il mondo) della sua CDN OpenConnect
- l'abbonato richiede il contenuto, il fornitore di servizi restituisce il manifesto
  - utilizzando il *manifest*, il client recupera i contenuti alla massima velocità supportabile
  - può scegliere una velocità o una copia diverse se il percorso di rete è congestionato



# Reti per la distribuzione di contenuti - Content distribution networks (CDNs)



*Sfide OTT:* affrontare un Internet congestionato dal "bordo"

- quale contenuto inserire in quale nodo CDN?
- da quale nodo CDN recuperare i contenuti? A quale velocità?