

Multimedia Networking

Reti Multimediali e Requisiti QoS

Le **reti multimediali** si occupano della distribuzione di contenuti come video e audio su Internet. Questo è un compito impegnativo a causa di:

- **Numerose sottoreti e dispositivi** distribuiti geograficamente.
- **Requisiti di qualità variabili** per i flussi video.
- **Rischio elevato di degrado della qualità** poiché i fornitori di servizi Internet (ISP) non sempre possiedono l'intera rete di distribuzione dei contenuti.
- **Problemi comuni** come ritardo dei pacchetti, jitter e perdita di pacchetti. Il Jitter è importante e rappresenta la magnitudine della variazione del ritardo. Si risolverebbe salvando i dati che arrivano in un buffer e mandandoli compensando il jitter. Bisogna cercare di rendere costante il rate di trasmissione e il rate di ricezione. Rende poco fluide le videochiamate.
- Le **aspettative degli utenti** sono elevate, spesso influenzate dalla qualità offerta dalla TV via cavo e satellitare.

Il **Quality of Service (QoS)** è il framework per la gestione del traffico di rete nei sistemi di distribuzione, con l'obiettivo di garantire prestazioni e stabilizzare le condizioni della rete attraverso la gestione delle risorse e la prioritizzazione del traffico.

Tipi di Traffico Internet

Il traffico Internet si divide in due categorie principali:

1. Traffico Elastico:

- Si adatta facilmente ai cambiamenti di ritardo e throughput.
- È il tipo di traffico tradizionale per cui Internet è stato progettato (basato su TCP/IP).
- **Esempi:** trasferimento di file (FTP), e-mail (SMTP), login remoto (Telnet, SSH), gestione di rete (SNMP) e accesso web (HTTP).

2. Traffico Inelastico:

- Non si adatta facilmente ai cambiamenti di ritardo e throughput.
- **Esempi:** trasmissione multimediale (voce e video) e traffico interattivo ad alto volume (es. giochi online).
- **Requisiti specifici:**

- **Throughput minimo:** molte applicazioni richiedono un throughput minimo garantito.
- **Ritardo (o latenza) massimo:** fondamentale per lo streaming video in tempo reale.
- **Jitter (variazione del ritardo):** la magnitudine della variazione del ritardo è critica; i pacchetti vengono bufferizzati per compensare il jitter.
- **Perdita di pacchetti:** le applicazioni in tempo reale variano nella tolleranza alla perdita di pacchetti.
- **Sfide:** è difficile soddisfare questi requisiti in un ambiente con ritardi di coda variabili e perdite per congestione.
- **Traffico in tempo reale:** può essere da sorgenti continue (pacchetti a dimensione fissa a intervalli fissi), on/off (alterna periodi attivi e di inattività, es. voce), o a dimensione di pacchetto variabile (es. video MPEG).

Classi di Servizi Multimediali

Le applicazioni multimediali hanno diverse esigenze di QoS:

- **Traffico bidirezionale:** bassa latenza e jitter (es. telefonia vocale, video).
- **Latenza e jitter moderati:** ordine e sincronizzazione rigorosi.
 - **Traffico unidirezionale:** streaming.
 - **Traffico bidirezionale:** web surfing e giochi online.
- **Nessun requisito in tempo reale:** download o trasferimento di file di grandi dimensioni.

Protocolli Multimediali

L'implementazione dei servizi nelle reti IP richiede protocolli a livello di applicazione specifici, soprattutto per i dati in tempo reale che hanno requisiti diversi rispetto al trasferimento di massa.

QoS a livello IP

Il protocollo IP convenzionale offre solo un servizio "best-effort" senza differenziazione. Nel senso che **fa del suo meglio** per consegnare i pacchetti:

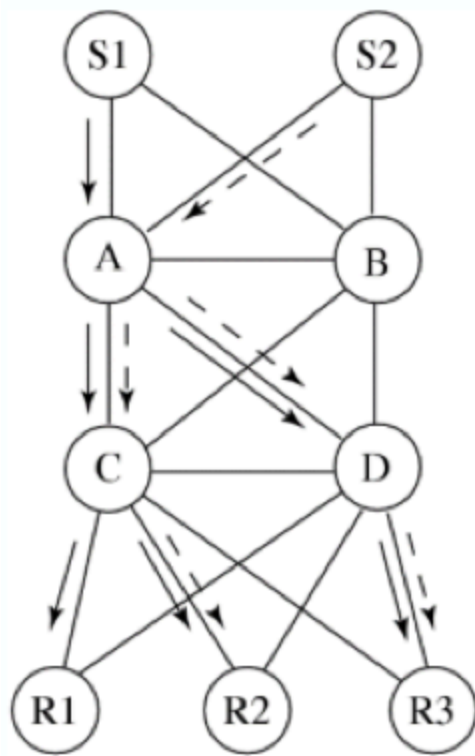
- **Non garantisce:**
 - che i pacchetti arrivino a destinazione,
 - né **quando** arrivano (nessuna garanzia di latenza),
 - né **in che ordine** arrivano,
 - né che **non vengano persi**.

Due approcci comuni per il QoS:

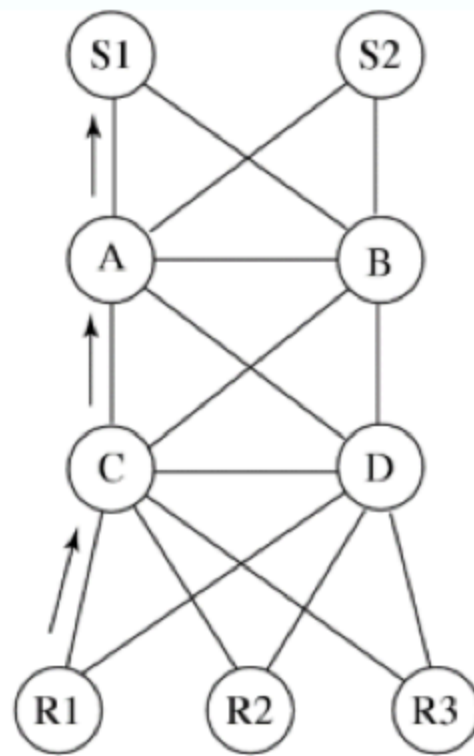
IntServ (Integrated Services):

- Modello QoS per reti a pacchetti che riserva un livello di servizio specificato da sorgente a destinazione.
- Utilizza il **Resource reSerVation Protocol (RSVP)** per la prenotazione delle risorse.
- **Classi di servizio (CoS):** Servizio garantito (limiti rigidi su ritardo e banda), Carico Controllato (stesso servizio come se la rete fosse scarica), Best-effort (nessuna garanzia).
- **Requisiti di implementazione:** ogni router deve supportare RSVP, controllo di ammissione della connessione, mantenimento dello stato e prioritizzazione del traffico per flusso.
- L'RSVP è un protocollo inizializzato dai ricevitori e la richiesta viaggia all'indietro verso il mittente ma non necessariamente seguendo la stessa strada.

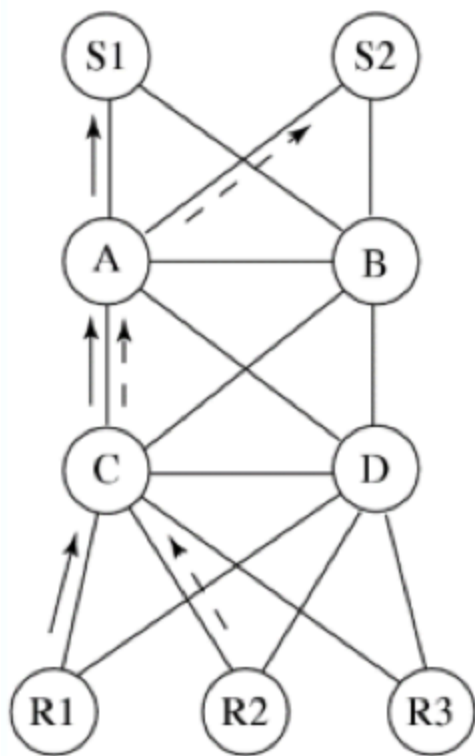
- RSVP crea solo "soft state" che devono essere periodicamente aggiornati dal ricevitore.



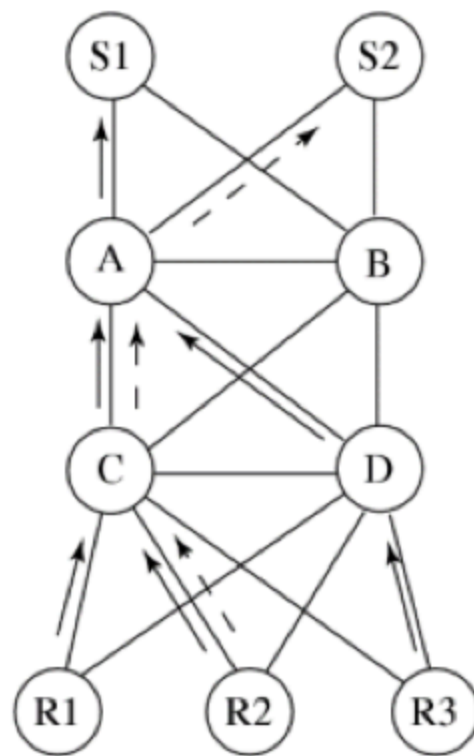
(a)



(b)



(c)



(d)

Sostanzialmente:

1. S1 e S2 mandano un messaggio Path ai riceventi per informarli del percorso e delle caratteristiche del flusso dei dati SENZA riservare nessuna risorsa.
2. R1 riceve il messaggio e manda un RESV a S1 allocando quindi le risorse esplicitate dal Path Message.
3. R2 manda un Resv a S2 per riservare risorse, dove nei nodi A e C si hanno due canali riservati uno per R1 e uno per R2.
4. R2 e R3 mandano poi un RESV a S1.

DiffServ (Differentiated Services):

- Approccio opposto basato sull'aggregazione e classificazione del traffico.
- I router implementano **per-hop behaviors (PHBs)** che definiscono le proprietà di inoltro dei pacchetti associate a una classe di traffico.
- Il primo passo è marcare i pacchetti con un valore nel **campo DS** (campo ToS in IPv4, Traffic Class in IPv6).
- **PHBs comuni:** Default PHB (best-effort), Expedited Forwarding (EF) per traffico a bassa perdita e bassa latenza, Assured Forwarding (AF) per garanzia di consegna sotto condizioni. L'AF può allocare la banda in categorie (es. Gold 50% BW, Silver 30% BW, Bronze 20% BW).
- I router utilizzano **shapers** (mettono i pacchetti in code diverse) e **droppers** (scartano i pacchetti, es. Random Early Detection) sulle loro code di input.
- La schedulazione può utilizzare **Weighted Fair Queuing (WFQ)**. Nel senso che ogni classe di traffico avrà un peso w_i e si calcolerà il rate di trasmissione di ogni queue associata alla classe tramite la seguente formula:

$$R_i = R \cdot \frac{w_i}{\sum_j w_j}$$

dove R è la potenza di trasmissione totale.

- **DiffServ vs. IntServ:** DiffServ ha una granularità di controllo più grossolana (classi aggregate vs. flussi individuali), è più semplice e scalabile. Possono lavorare insieme (IntServ all'edge, DiffServ nel core).
- Nella maggior parte dei casi, le applicazioni multimediali si affidano ancora al servizio "best-effort" di base, dato che IntServ e DiffServ sono poco diffusi nelle WAN per la loro complessità.

Protocolli TCP/IP

- **TCP (Transmission Control Protocol):**
 - Fornisce una **consegna affidabile e sequenziale** di flussi di pacchetti.
 - I pacchetti sono riconosciuti e ritrasmessi se necessario.

- Utilizza il **controllo della congestione** e adatta la velocità di invio alle condizioni della rete.
- È **orientato alla connessione** (richiede una handshake a tre vie).
- Non ideale per il real-time a causa della ritrasmissione e delle fluttuazioni di velocità.
- **UDP (User Datagram Protocol):**
 - Fornisce il **servizio IP "raw"** (grezzo).
 - È **senza connessione** e non offre garanzie di consegna.
 - I pacchetti UDP possono essere persi, ritardati o fuori ordine.
 - Adatto per **multicast e broadcast**.
 - Ideale per il real-time grazie alla sua velocità e bassa latenza, ma richiede meccanismi di controllo degli errori a livelli superiori.

Protocolli di Trasporto Audio/Video

La scelta del protocollo di trasporto è cruciale. TCP non è adatto per il real-time a causa dell'affidabilità (ritrasmissioni), mentre UDP è veloce ma richiede un controllo degli errori.

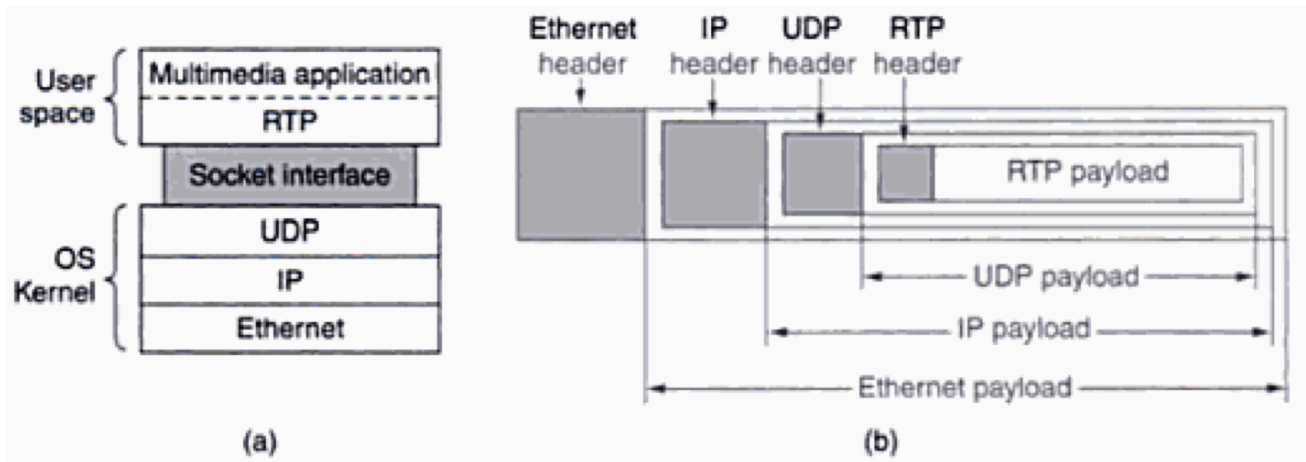
Real-time Transport Protocol (RTP)

RTP è un protocollo progettato per il **trasferimento in tempo reale di dati multimediali**, come:

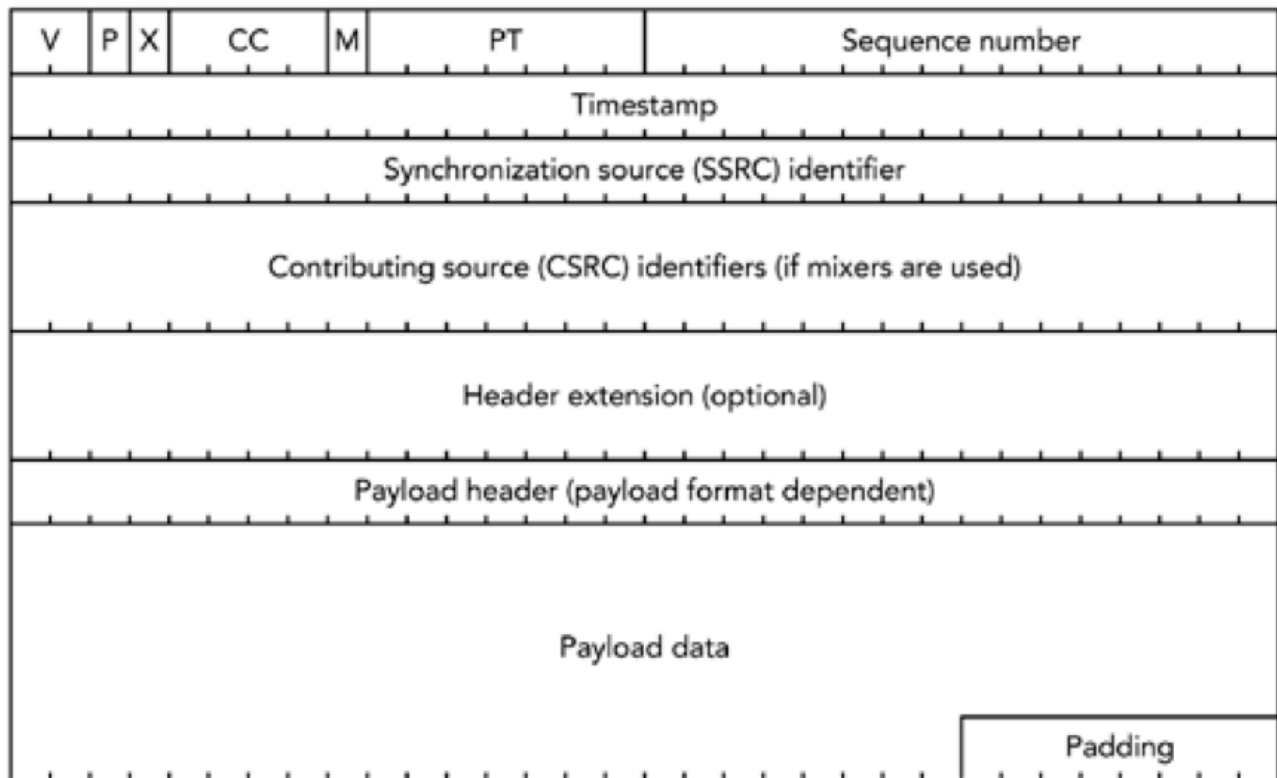
- **Audio** (es. VoIP),
- **Video** (es. videoconferenze, streaming live).

-
- Protocollo di rete per la **consegna di flussi audio e video** su reti IP.
 - Generalmente **esegue su UDP**.
 - Utilizzato con il **RTP Control Protocol (RTCP)**. Esso monitora le statistiche di trasmissione e il QoS.
 - RTP fornisce funzioni per la **compensazione del jitter**, la **gestione della perdita di pacchetti** e la **consegna fuori ordine**, non possibili con UDP.
 - È il **principale protocollo** di trasporto audio/video nelle reti IP.
 - **Motivi per usare UDP con RTP:** scalabilità in ambienti multicast (TCP è connection-oriented), tolleranza alla perdita di dati multimediali (la ritrasmissione non è sempre necessaria o desiderabile), e le fluttuazioni di velocità di TCP non sono adatte ai media continui.
 - **Scalabile:** da punto-punto a sessioni multicast con migliaia di utenti.
 - RTP non fornisce meccanismi per garantire la consegna tempestiva o altre garanzie di QoS.

- L'incapsulamento RTP è visto **solo nei sistemi finali**, poichè è del livello applicativo.



- **Limiti:** non specifica algoritmi per il playout dei media (non gestisce quando e come riprodurre i contenuti), la rigenerazione del timing (si ha il timestamp per dare un ordine temporale dei dati ma non sta al protocollo ricostruire il flusso temporale coerente), la sincronizzazione tra flussi (sincronizzazione tra video o audio, non sta all'applicazione gestire la sincronizzazione), l'occultamento/correzione degli errori o il controllo della congestione.
- Il **pacchetto di trasferimento dati RTP** è composto da un'intestazione obbligatoria (12 byte), estensione opzionale, intestazione payload opzionale e dati payload.
- **Elementi dell'Header:**
 - **V (Version):** Versione attuale è 2.
 - **P (Padding):** Indica che il payload è stato riempito.
 - **X (Header Extension):** Indica la presenza di estensioni all'header.
 - **M (Marker Bit):** Marca eventi di interesse (es. primo pacchetto dopo il silenzio, ultimo pacchetto in un I o P-frame).
 - **PT (Payload Type):** Identifica il tipo di media trasportato (7 bit). Il ricevente esamina il seguente valore per decidere come trattare il dato.
 - **SN (Sequence Number):** Identifica i pacchetti in una sequenza, aiuta a rilevare perdite o pacchetti fuori ordine (16 bit non firmato, incrementa di 1 ad ogni pacchetto).
 - **Timestamp:** Utilizzato per schedare e sincronizzare il playout del flusso media (32 bit non firmato, aumenta a una velocità dipendente dal media).



Per gestire differenti tipi di media, RTP specifica dei profili. Ogni profilo (Audio - Video) ha dei parametri relativi al media trasportato. I profili sono identificati mappando il tipo di payload con le specifiche. Ogni profilo definisce:

- dimensione payload nell'RTP header e il numero di bit usati per marcare eventi significativi.
- un header addizionale.
- l'intervallo di report del RTCP.
- meccanismi di security.
- Mappare RTP e RTCP in un protocollo di trasporto(UDP - TCP).

RTP Control Protocol (RTCP)

- Protocollo "gemello" di RTP, definito in RFC 3550.
- Fornisce **statistiche out-of-band** e informazioni di controllo per una sessione RTP.
- Non trasporta dati media, ma fornisce **reporting periodico** sullo stato della sessione/connessione (qualità di ricezione, informazioni per sincronizzare i flussi media, identificazione dei partecipanti).
- Tutti i partecipanti inviano pacchetti RTCP composti a intervalli regolari, ottenendo così conoscenza di tutti gli altri partecipanti (natura P2P).
- Le componenti dell'implementazione di questo protocollo sono:
 - regole di temporalizzazione: intervalli entro i quali vengono mandati messaggi di reporting.
 - database dei partecipanti

- un insieme di tipi di pacchetti e regole per aggregarli in pacchetti.
- Ogni sessione RTP è tipicamente un indirizzamento del tipo single to multicast.
- I pacchetti RTP - RTCP sono distinti ognuno da porte distinte (una per RTP e una per RTCP). (5004- 5005)

Servizi di Streaming

Session Description Protocol (SDP)

- Formato di messaggio per descrivere i **parametri di inizializzazione di un flusso multimediale**.
- Non è progettato per la negoziazione del contenuto o della codifica dei media, ma per **annunciare una sessione**, invitare partecipanti e negoziare parametri come tipo e formato dei media.
- Formato: `<type>=<value>` .
- Tre parti principali:
 - **Session Description**: descrive la sessione, come l'host della sessione.
 - **Time Description**: provvede a dare info sul tempo della sessione(inizio, fine).
 - **Media Description**: provvede a dare info sul tipo di media utilizzato.

Real-Time Streaming Protocol (RTSP)

- **Protocollo a livello di applicazione** per il controllo della consegna di dati con proprietà in tempo reale (audio e video).
- Può essere pensato come un **telecomando** per lo streaming, con capacità di avviare, fermare, mettere in pausa, ecc..
- È **senza connessione** (connectionless) (lo stato è mantenuto sul server tramite un identificatore di sessione).
- Sintassi e operazioni sono **simili a HTTP/1.1**; SDP può essere usato con RTSP per descrivere i flussi media.
- Le **operazioni chiave** includono OPTIONS, DESCRIBE, SETUP, PLAY, PAUSE e TEARDOWN.
- Si inizia sequenza con URL:

entire sequence is normally initiated
 [rtsp:|rtspu:]//host [:port] path

dove rtsp si appoggia a TCP mentre rtspu si appoggia al rtspu.

- La porta è 554

Caching

Il **caching distribuito** è essenziale per la scalabilità dei servizi multimediali, data la grande quantità di dati e la necessità di consegna tempestiva.

Elementi di Caching

- **Strutturazione del contenuto:** il video può essere memorizzato a diverse risoluzioni e larghezze di banda. La cache può fornire prima flussi di qualità inferiore e poi adattarsi a qualità migliori.
- **Caching temporizzato:** non sempre pratico o conveniente memorizzare tutto il contenuto. La cache deve essere in grado di interpretare RTSP, RTCP e altri messaggi.
- **Posizione della cache:**
 - **Proxy:** al confine di una rete aziendale/ISP per servizi di caching.
 - **Reverse proxy:** vicino al fornitore di contenuti.
 - **Server di cache/contenuti dedicati:** in località geografiche diverse.
 - **Client:** cache locale per contenuti già visti, o in P2P per distribuire ad altri client.
- **Operazioni con caching parziale:** il proxy può sfruttare il messaggio `Range` di RTSP per richiedere una porzione specifica del video.
- **Algoritmi di Proxy Caching:**
 - **Sliding-Interval Caching:** memorizza un intervallo scorrevole per facilitare accessi consecutivi.
 - **Prefix Caching:** memorizza la porzione iniziale (prefisso), poi recupera il resto (suffisso). Non è adatta a streaming real time. È diviso in Prefix Caching(salvo una scena iniziale del video per utenti che vedono i primi secondi e questo diminuisce il delay di setup e nel mentre viene fatta richiesta per gli altri contenuti) e Segment Caching(Salvo scene specifiche).
 - **Segment Caching:** generalizza il prefix caching, suddividendo un oggetto multimediale in segmenti.
 - **Rate-Split Caching:** partiziona il media lungo l'asse della velocità, utile per lo streaming VBR (Variable Bit Rate).

HTTP Streaming

C'è stato uno spostamento significativo da RTP a HTTP per lo streaming, dovuto a:

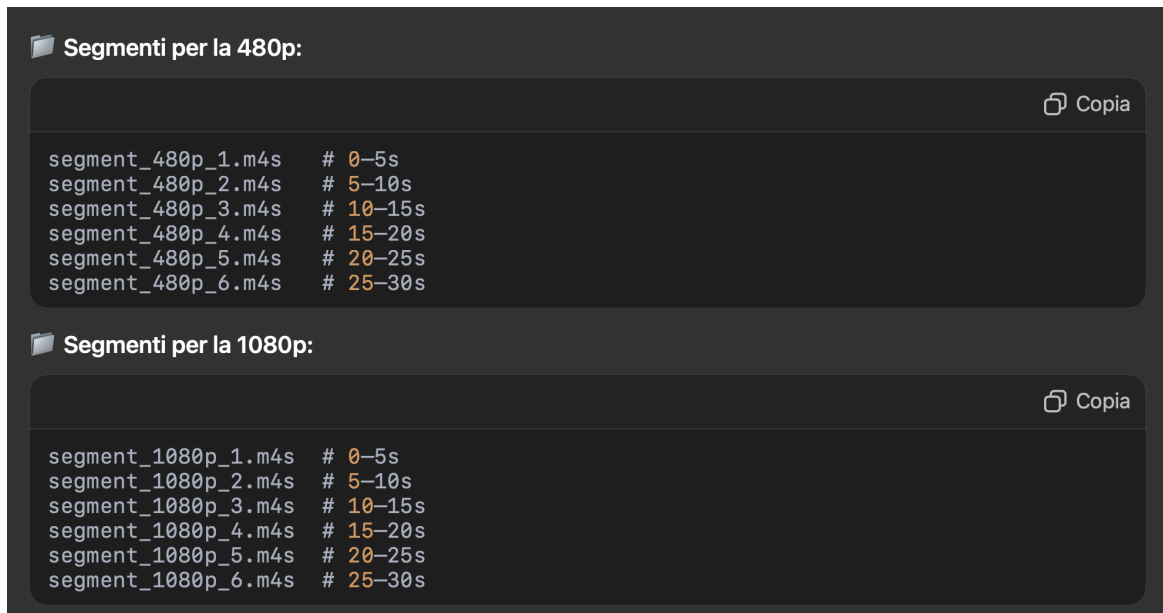
- **Aumento della larghezza di banda** disponibile per il consumatore medio.
- **Impatto minimizzato dell'uso di TCP** (ritrasmissione meno critica con banda sufficiente).
- **Eliminazione dei problemi di NAT traversal** usando HTTP su TCP.

- **Minimizzazione o eliminazione del bisogno di software server speciali** (utilizzo di web server, proxy, server di caching standard).

Piattaforme di HTTP Streaming includono HTTP Live Streaming (Apple), Smooth Streaming (Microsoft), HTTP Dynamic Streaming (Adobe) e **DASH (Dynamic Adaptive Streaming over HTTP)**.

Dynamic Adaptive Streaming over HTTP (DASH)

- Termine generale per qualsiasi metodo di streaming HTTP che permette al client di **selezionare la qualità** di un flusso media.
- **ISO/IEC FCD 23009-1 (MPEG-DASH)** è uno standard aperto.
- **Proprietà:**
 - Utilizza HTTP per il trasporto, consentendo l'uso di componenti web standard.
 - Il client sceglie e cambia la qualità/bit rate del flusso.
 - I segmenti sono memorizzati come file individuali, usati con file di indice e descrizione.
- **Livelli SVC (Scalable Video Coding):** MPEG-DASH consente la regolazione della qualità del flusso offrendo al client più livelli SVC, che sono additivi (il client riceve sempre il livello base e aggiunge strati di miglioramento per aumentare la qualità).
- **Non è un protocollo o un sistema**, ma un insieme di formati.
- **Due componenti principali:**
 - **Media Presentation Description (MPD):** un file XML che definisce:
 - Periodi (segmenti più grandi): È una **sezione autonoma del flusso video/audio**, che può avere **propri segmenti, rappresentazioni, codec, lingue, ecc.** E' il contenitore temporale delle rappresentazioni di un certo numero di segmenti.
 - Rappresentazioni (diverse codifiche/risoluzioni/frame rate): sono diverse tipologie di periodo che si diversificano in base alla risoluzione, frame rate ed encoding.
 - Segmenti: rappresentano porzioni di un contenuto multimediale(audio o video) che ha un intervallo temporale (2-10 s). Ogni segmento è tracciabile tramite URL.



- Il formato dei segmenti media.
- **Procedura:** Il client richiede un MPD, riceve l'MPD con i dettagli sul contenuto e le URL. Il cliente usa l'URL per scaricare i segmenti. Il range di inizializzazione e di indice nella sezione del segmento base dirà al cliente dove trovare informazione e contenuto nella stream e può cambiare rappresentazione alla fine di un segmento.
- **Live Streams:** considerazioni speciali per i flussi live includono il ritardo end-to-end e la generazione/aggiornamento dell'MPD.

Consegna Multi-utente

Per gestire un gran numero di partecipanti (es. milioni di utenti per un evento sportivo, come l'esempio dei 700 Gb/s della BBC alle Olimpiadi), sono considerati due meccanismi principali:

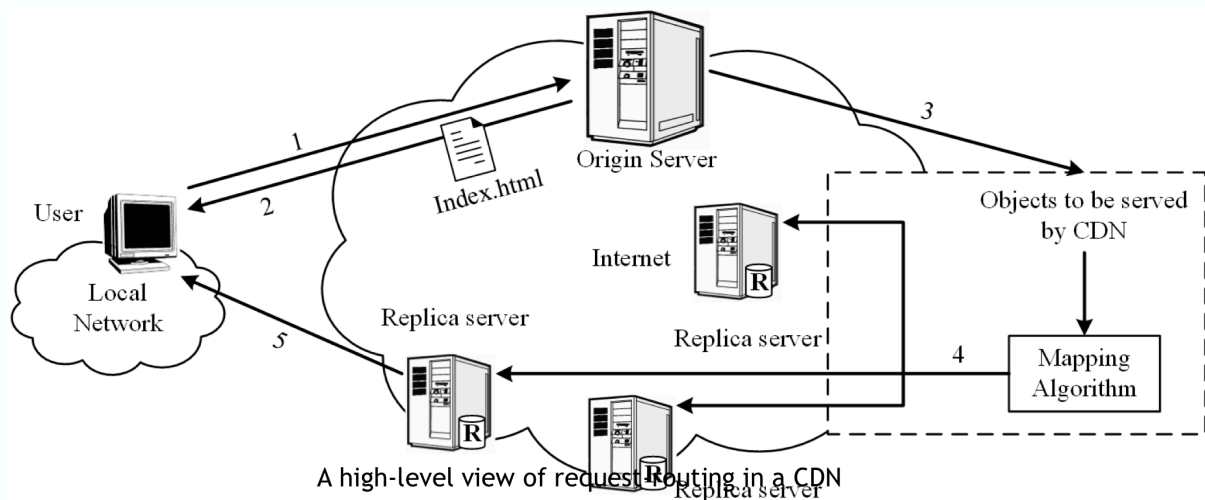
1. Content Distribution/Delivery Networks (CDN):

- Raccolta di **server geograficamente dispersi** per fornire contenuti ad alte prestazioni e affidabilità agli utenti finali.
- Riduce il tempo di download, il carico sui server del fornitore di contenuti, la perdita di pacchetti, la latenza e il jitter.
- Aumenta l'affidabilità e la resistenza agli attacchi DoS.
- Facilita la scalabilità a un gran numero di utenti.
- **Componenti tecnologici (bricks):**
 - **Caricamento del contenuto:** identificazione, precaricamento o caching tradizionale, algoritmi basati sulla popolarità.
 - **Distribuzione/Sincronizzazione del contenuto:** push, pull dal server originale o scambio P2P tra server CDN.
 - **Selezione del server CDN:** dirigere il client al server appropriato basandosi su criteri come geolocalizzazione, numero di hop di rete, latenza, disponibilità, carico

(gli algoritmi possono combinare questi criteri).

- **Reindirizzamento del client:** tramite HTTP 30x redirect o manipolazione del DNS.
- **Streaming live:** richiede ottimizzazione dei percorsi dall'ingest ai server edge per la consegna in tempo reale.
- **Flusso di richiesta in una CDN:** l'utente richiede contenuto all'origine, che reindirizza la richiesta al CDN (per contenuti a banda larga e frequenti), il CDN seleziona il server replica e serve il contenuto.

Per servire il contenuto con alta larghezza di banda, il server origine ridireziona la richiesta al provider CDN utilizzando un algoritmo di mapping per rintracciare il CDN più vicino o meno congestionato o con costo minore e il CDN serve l'utente.



- **Esempio: Akamai Streaming CDN:** Il CDN più grande al mondo (circa 240.000 server), con reti di consegna parallele adattate a servizi specifici. La sua architettura ha **tre livelli**: Entry points (memorizzazione fisica), Edge servers (consegna agli utenti), e Reflectors (strato intermedio per migliorare la scalabilità, connettendo più entry point ed edge server).
- **Case Study: Netflix:** Netflix ha spostato quasi tutta la sua infrastruttura multimediale sul cloud pubblico di Amazon, utilizzando **EC2** per la conversione dei contenuti, **S3** per l'archiviazione e **CDN di terze parti** (Akamai, Limelight, Level-3) per la distribuzione.
- **Netflix OpenConnect:** Netflix ha introdotto la propria infrastruttura CDN nel 2011 per gestire la crescente scala dello streaming. È una rete globale che distribuisce contenuti via HTTP/HTTPS. I **motivi** includono la necessità di collaborare direttamente con gli ISP e di creare una soluzione di caching proattiva più efficiente rispetto alle CDN tradizionali. I **blocchi costruttivi** sono gli **Open Connect Appliances (OCAs)**, server proprietari che memorizzano file video/immagine e li servono ai dispositivi client. Gli OCAs possono essere installati in punti di scambio Internet (IXPs)(switch tra ISP e Cloud Center) o direttamente nelle reti degli ISP, spesso gratuitamente per gli ISP(vodafone, tim, wind, ecc) qualificati.

2. Tecnologie Multicast: Alternative per la distribuzione multi-utente.

