UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO
DI INGEGNERIA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

"APPROCCI RECENTI PER LO STREAMING VIDEO A BASSA LATENZA"

Relatore: Prof. Marco Cagnazzo

Laureando: Filippo D'Emilio

ANNO ACCADEMICO 2022 – 2023

Data di laurea: 27 Settembre 2023

ABSTRACT

Lo scopo di questo documento è di fornire al lettore le conoscenze di base che riguardano lo streaming video, il concetto di streaming adattivo e QoE, oltre che a cenni sulla codifica video, sul codec VVC e le metriche di valutazione di qualità. In seguito è trattato l'argomento centrale, ovvero lo streaming video a bassa latenza ed è descritta nel dettaglio la tecnica GDR, ovvero una recente tecnica di codifica a bassa latenza. Infine tramite delle sperimentazioni sono stati messi in pratica gli argomenti trattati: sono state effettuate delle codifiche tradizionali e delle codifiche GDR, e mettendole a confronto tramite i dati ottenuti sono stati prodotti dei risultati.

La percentuale di traffico internet degli utenti relativo allo streaming di contenuti video è costantemente aumentata negli anni, nel 2016 costituiva il 76% del traffico globale e nel 2021 l'82%. Lo streaming video ha visto una significativa evoluzione negli anni: con la nascita dello streaming video adattivo il bitrate a cui viene scaricato il contenuto viene scelto da appositi algoritmi in esecuzione lato client e la lo standard MPEG-DASH, nato nel 2013, ha fornito un modello universale per lo streaming video adattivo su http basato sull'architettura client/server.

La QoE (Quality of Experience) indica la qualità di un servizio di streaming video nel complesso e uno degli obiettivi principali delle piattaforme che offrono tali servizi è massimizzarla. I fattori che determinano la QoE variano a seconda del tipo di servizio offerto. In particolar modo nello streaming video a bassa latenza è fondamentale che il ritardo con cui il video viene visualizzato dall'utente sia limitato e di conseguenza la QoE è significativamente influenzata da questo aspetto.

Il codec VVC è stato rilasciato nel 2021 e permette di avere un'efficienza di codifica dal 40% al 50% maggiore del precedente HEVC. Il VVC supporta la tecnica GDR, ovvero una tecnica di streaming video a bassa latenza che distribuisce in più immagini aree codificate in modalità intra, rendendo il bitrate dello streaming significativamente più uniforme rispetto alle tecniche tradizionali.

Nel capitolo sperimentale sono state effettuate misurazioni che mettono a confronto una codifica VVC standard e una codifica VVC con GDR, con lo scopo di verificare le conseguenze che l'utilizzo di GDR comporta in termini di uniformità del bitrate ed efficienza della codifica.

INDICE

INTRODUZIONE

Il traffico internet su scala globale è da diversi anni dominato da contenuti video. Secondo [1] il traffico degli utenti relativo allo streaming video costituiva il 76% del traffico globale nel 2016, e nel 2021 ammontava all' 82%. Secondo [2] Youtube, la nota piattaforma di condivisione di contenuti video, conta nel 2023 1.8 miliardi di diversi utenti mensili.

Lo streaming video consiste in una trasmissione continua di file video attraverso internet in cui la risorsa multimediale può essere visualizzata senza avere la necessità di salvarla permanentemente nella memoria del dispositivo.

Una sessione di streaming video coinvolge due protagonisti, ovvero client e server. Il server possiede in memoria una versione opportunamente compressa della risorsa multimediale, ottenuta tramite operazioni di codifica (detta anche compressione). Il client richiede al server la risorsa video selezionata e questa viene spedita dal server attraverso internet, scaricata e decodificata dal client per essere poi visualizzata a schermo tramite un apposito video player. Questa è evidentemente una definizione semplice e poco dettagliata relativa allo streaming video, i vari elementi che lo costituiscono verranno approfonditi nelle sezioni successive.

Il continuo aumentare nel corso degli anni della percentuale del traffico internet che riguarda lo streaming video è dovuto a diverse ragioni, fra le quali:

- La nascita e la crescita di piattaforme di streaming, sia per contenuto "on-demand", ovvero video memorizzati in un server pronti per essere visualizzati da un dispositivo client su richiesta; sia per contenuto "dal vivo", in cui la visualizzazione di un evento che sta avendo luogo realmente in concomitanza con lo streaming, come eventi sportivi.
- Il miglioramento della larghezza di banda delle connessioni ha permesso il nascere e diffondersi di nuove qualità, dette Ultra High Definition, come le risoluzioni 3840 x 2160 e 7.680 x 4.320 (numero pixel costituenti le immagini del video, rispettivamente per riga e per colonna, dette anche risoluzioni 4K e 8K) . Come diretta conseguenza dell'utilizzo di alte risoluzioni il volume del traffico internet causato dallo streaming aumenta notevolmente.
- L'esigenza di poter comunicare da remoto in ambito professionale, e non solo, tramite applicazioni di videochiamate o videoconferenze. Negli ultimi anni si sono infatti comprese le potenzialità di questo tipo di applicazioni e affinché il servizio fornito sia valido è necessario che la comunicazione avvenga in tempo reale, quindi è necessario che il ritardo dello streaming video sia notevolmente ridotto.

- il continuo aumentare di popolarità dei creatori di contenuti, per esempio gamers che condividono in diretta le loro esperienze di gioco su apposite piattaforme, oppure utenti dei social media che producono contenuto video di diversa natura a seconda delle loro passioni o del settore di cui si occupano. Al giorno d'oggi è possibile trovare contenuti video che trattano argomenti di ogni tipo, e il mestiere del creatore di contenuti è diventato di fatto un lavoro vero e proprio visto le potenzialità di profitto che si hanno introducendo nei propri contenuti annunci pubblicitari e/o sponsorizzazioni.
- Il mondo dei videogiochi ha visto la nascita di una nuova concezione di gaming (giocare ai videogiochi), ovvero il cloud gaming. Grazie a quest'ultimo è possibile delegare ad un server tutte le operazioni necessarie per giocare ad un videogioco, a partire dal fatto che non è richiesto avere il videogioco installato sul proprio dispositivo. Il dispositivo client agisce come un desktop remoto, il videogioco infatti è in esecuzione sul server e l'output video prodotto da esso è inviato mediante uno streaming. Per poter giocare sul cloud è necessario possedere una connessione con prestazioni sufficientemente alte, sia in termini di larghezza di banda che di latenza. In compenso a questo requisito la parte hardware impiegata per eseguire il videogioco è delegata al server, ed è possibile selezionare qualunque videogioco compreso nel catalogo reso a disposizione e iniziare a giocarci immediatamente.

Le tecniche e le scelte implementative che riguardano lo streaming video attraverso internet hanno visto enormi cambiamenti e miglioramenti nel corso degli anni. Sono tali miglioramenti infatti che hanno permesso a questo fenomeno di diffondersi e raggiungere i numeri di oggi. Fra i cambiamenti più incisivi c'è la nascita dello streaming adattivo e la creazione dello standard DASH (Dynamic Adaptive Streaming over HTTP).

Un altro aspetto che ha portato ad un'evoluzione continua del fenomeno dello streaming video è il continuo sviluppo di nuove tecniche di codifica, che permettono di rappresentare la risorsa multimediale in maniera sempre più efficiente, garantendo qualità maggiore a parità di larghezza di banda a disposizione. Il codec (codificatore/decodificatore) HEVC [3] (High Efficency Video Coding o H-265), nato nel 2013 è stato esteso e migliorato sotto ogni aspetto dal codec VVC [4] (Versatile Video Coding o H-266), rilasciato nel 2021. Il VVC ha introdotto innovative modalità di codifica, permettendo secondo [12] una riduzione dal 40% al 50% del numero di byte per rappresentare una risorsa a parità di qualità percepita dall'utente.

Un concetto molto importante nello streaming video è la QoE (Quality of Experience), ovvero una valutazione complessiva del servizio di streaming video offerto agli utenti. La QoE è un aspetto di importanza critica per le piattaforme che offrono servizi di streaming video (dette anche OTT,

“Over The Top”), il cui scopo è quello di massimizzarla e fornire un servizio migliore della concorrenza. I fattori che determinando la QoE possono differire a seconda del tipo di servizio di streaming considerato, pertanto è necessario fare le correte scelte implementative che considerano tale aspetto.

Il codec VVC è detto “versatile” poiché fornisce una vasta gamma di modalità di codifica, con lo scopo di poter adattarsi ai diversi tipi di contenuti video emersi negli ultimi anni. Fra le modalità offerte dal VVC è presente anche una tecnica di codifica ideata per lo streaming video a bassa latenza, ovvero la tecnica GDR (Gradual Decoding Refresh). Negli ultimi anni è cresciuto l’utilizzo di applicazioni di streaming video a bassa latenza, che per garantire un’opportuna QoE necessitano di operare con valori di latenza al di sotto di una determinata soglia, a seconda di quanto stretto è tale requisito.

Lo scopo di questo documento è di fornire al lettore le conoscenze di base che riguardano lo streaming video, il concetto di streaming adattivo e QoE, oltre che a cenni sulla codifica video, sul codec VVC e le metriche di valutazione di qualità. In seguito è trattato l’argomento centrale, ovvero lo streaming video a bassa latenza ed è descritta nel dettaglio la tecnica GDR, ovvero una recente tecnica di codifica a bassa latenza. Infine tramite delle sperimentazioni sono stati messi in pratica gli argomenti trattati: sono state effettuate delle codifiche tradizionali e delle codifiche GDR, e mettendole a confronto tramite i dati ottenuti sono stati prodotti dei risultati.

Il documento è quindi strutturato nel seguente modo:

- Nel capitolo 1 è descritto il concetto di streaming adattivo e trattato in particolare lo standard MPEG-DASH.
- Nel capitolo 2 è trattata la QoE e i vari aspetti che la definiscono.
- Nel capitolo 3 sono dati dei cenni sulla codifica video e sul codec VVC, oltre che ad essere presentate alcune metriche di valutazione utilizzate per le risorse video.
- Nel capitolo 4 è trattato lo streaming video a bassa latenza e in particolare la tecnica di codifica GDR.
- Nel capitolo 5 è presentata l’analisi sperimentale che mette a confronto una tecnica di codifica tradizionale e una tecnica di codifica a bassa latenza (GDR), per verificare le conseguenze del suo utilizzo.

Capitolo 1 - LO STREAMING VIDEO ADATTIVO

1.1 Il concetto di streaming adattivo

La nascita e la diffusione di tecniche di streaming video adattivo ha rivoluzionato il mondo dello streaming video, infatti esse permettono di migliorare notevolmente la QoE ed hanno completamente sostituito da molti anni le modalità usate in precedenza.

Nello streaming adattivo la risorsa video è suddivisa in segmenti e ognuno di essi corrisponde a qualche secondo della risorsa video , per esempio da 1 a 15 secondi [9]. La lunghezza dei segmenti è una decisione presa a livello di applicazione.

Ogni segmento è codificato in più modi e ogni diversa codifica ha un proprio tasso di codifica, o bitrate, misurato in bit per secondo [bps], ovvero il numero di bit che devono essere impiegati per rappresentare e scaricare un secondo di tale segmento. Più alto è il tasso di codifica selezionato per un dato segmento maggiore è la qualità, ma allo stesso tempo il client deve scaricare più bit per visualizzarlo. Le diverse codifiche dei segmenti sono chiamate rappresentazioni e il server fornisce al client gli URL per accedere a tali risorse.

Il client dispone di un buffer, detto playout buffer, ovvero un'aerea di memoria in cui vengono salvati i segmenti scaricati e si svuota man mano che il video viene riprodotto a schermo. La dimensione del buffer tipicamente è espressa in numero di segmenti che esso può contenere. Prima di iniziare a visualizzare una risorsa video si attende che il buffer si popoli di un sufficiente numero di segmenti, per poi iniziare la visualizzazione della risorsa. Se il playout buffer non dispone di dati sufficienti durante una sessione di streaming video, la visualizzazione del video si interrompe e si crea una situazione di stallo, o freeze, in cui l'utente è costretto ad attendere che vengano scaricati dei segmenti per riprendere la visualizzazione della risorsa multimediale.

Quando il client inizia uno streaming video vengono scaricati i segmenti tramite delle richieste http (HTTP GET). La scelta di quale rappresentazione di un segmento scaricare è delegata agli algoritmi di ABR (adaptive bit rate), che sono in esecuzione lato client. L'algoritmo di ABR ideale seleziona sempre la migliore qualità della risorsa video senza mai far svuotare il buffer, consentendo quindi una visualizzazione continua della risorsa alla qualità migliore consentita dal throughput (la frequenza con cui avviene lo scambio di dati nell'applicazione di streaming in esecuzione, misurata in bit per secondo). Questa situazione ideale non è sempre possibile, e il compito di tali algoritmi è quello di selezionare un bitrate opportuno dei segmenti da scaricare fra quelli resi disponibili dal server, a seconda di determinati criteri [5] che possono essere:

- Una stima del throughput istantaneo dell'applicazione, basata su misurazioni del throughput raggiunto durante scaricamento dei segmenti più recenti. Tale misurazione permette di sceglie un bitrate appropriato che non sia superiore al throughput dell'applicazione, cosa che comporterebbe ad una tendenza del buffer di svuotarsi.
- Lo stato di riempimento del playout buffer, ovvero il numero di segmenti già scaricati e pronti per essere visualizzati dal client. Il numero di secondi *x* di video presenti nel buffer rappresenta infatti una garanzia che prima di *x* secondi il buffer non potrà essere vuoto, e quindi non potrà verificarsi uno stallo in tale arco di tempo. Provare a scaricare un segmento ad una qualità maggiore nell'arco di tempo *x* non comporta rischi, perché in caso il throughput dell'applicazione sia inferiore al bitrate selezionato l'algoritmo di ABR dispone di tempo a sufficienza per osservare che il buffer tende a svuotarsi e quindi correggere la qualità. Al contrario è rischioso fare un'operazione analoga se il buffer contiene pochi secondi di video, perché si potrebbe non avere tempo a sufficienza per correggere il bitrate, causando di conseguenza uno stallo.
- Una combinazione ponderata dei precedenti due criteri, tecnica spesso utilizzata nei casi reali.

La fig. 1 rappresenta un esempio di sessione di streaming adattivo. Il server contiene diverse sequenze di video chunks, o segmenti, che costituiscono le varie rappresentazioni fornite della risorsa multimediale. I rettangoli raffigurano i segmenti e le diverse colorazioni indicano un tasso di codifica differente, ovvero 0.1Mbps, 0.5 Mbps, 1 Mbps, 1.5Mbps. Il client inizia lo streaming richiedendo il file indice o manifesto, che contiene le informazioni necessarie al client riguardo alla risorsa video selezionata. Il client richiede quindi i segmenti in sequenza, ciascuno con un opportuno bitrate scelto da un algoritmo di ABR. Nella figura vengono selezionati prima i segmenti codificati a 0.1 Mbps, per poi richiedere i successivi a 0.5 Mbps e a 1Mbps; successivamente si ha un peggioramento del throughput dell'applicazione e viene abbassato il bitrate a 0.1 Mbps. Una volta che il throughput è ripristinato alla condizione precedente il client può quindi tornare a richiedere segmenti a 1 Mbps. Ogni qualvolta il client decide di cambiare il bitrate dei segmenti richiesti l'utente osserverà un cambio di qualità del video che sta guardando, quest'operazione è detta adattazione.

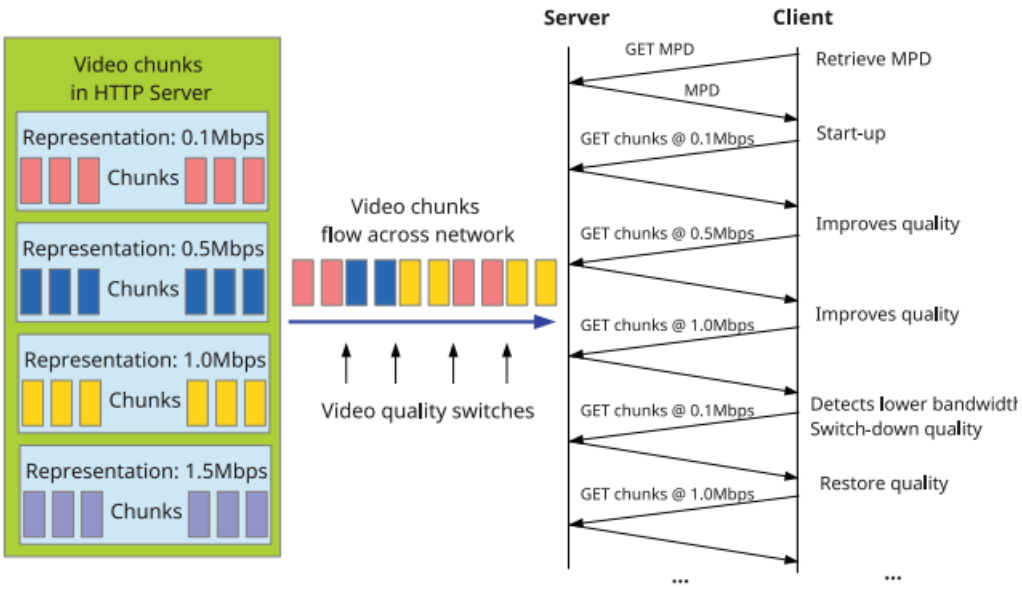


Fig. 1 [6]: Un esempio di sessione di streaming video adattivo.

Il motivo dell'esistenza del playout buffer e degli algoritmi di ABR è quello di minimizzare la probabilità che si verifichi una situazione di stallo, e a questo scopo è fondamentale avere una quantità sufficiente tempo di video nel buffer pronto per essere riprodotto, per intervenire se le condizioni di rete cambiano. Uno degli obiettivi principali delle applicazioni che offrono servizi di streaming video è quello di massimizzare la QoE, e da studi basati sui feedback degli utenti gli stalli sono il fenomeno che più degrada l'esperienza dell'utente, quindi l'adattazione è un aspetto centrale nello streaming video.

1.2 Streaming on-demand e dal vivo

Avendo introdotto il concetto di streaming adattivo, è ora possibile osservare quali sono le principali differenze fra streaming on-demand e streaming dal vivo.

Nello streaming on-demand, o su richiesta, la risorsa multimediale è già codificata nelle varie rappresentazioni e resa accessibile ai client in qualunque momento. Le priorità a livello server sono di fornire una scelta sufficientemente ampia di rappresentazioni, da bitrate molto alti a molto bassi, in maniera da rendere possibile ai client di selezionare il tasso ottimale. E' importante inoltre trovare la tecnica di codifica più efficiente, quindi con un alto tasso di compressione, per poter rappresentare la risorsa video con una qualità maggiore a parità di bitrate. In generale nello streaming on-demand si sceglie la lunghezza del segmento in modo che sia sufficientemente corto in modo che venga prodotta una reazione in tempo in caso lo stato della rete cambi, ma anche sufficientemente lungo da avere una codifica efficiente che fornisca un tasso di compressione elevato, infatti segmenti più lunghi permettono in generale una codifica migliore. Visto che questi

due aspetti sono contraddittori, la scelta deve essere fatta in base ad un trade-off, ovvero cercare il giusto compromesso che produca il risultato migliore.

Nello streaming dal vivo, o live streaming, il video da visualizzare corrisponde ad un evento che sta avendo luogo fisicamente in concomitanza con la visualizzazione. Il contenuto viene caricato quindi nel server in tempo reale mentre ne avviene la ripresa e non si ha quindi la possibilità di analizzare la codifica per scegliere la tecnica migliore. In determinati servizi di streaming dal vivo affinché il contenuto video sia reso disponibile al client in tempi utili è necessario minimizzare il ritardo end-to-end, o latenza, ovvero il tempo che trascorre fra la generazione del contenuto (nel server) e la presentazione del contenuto (al client). Tale ritardo comporta infatti ad uno sfalsamento fra l'evento reale e l'istante in cui tale evento è visualizzabile in streaming, elemento che spesso compromette l'esperienza dell'utente.

1.3 Lo standard MPEG-DASH

Il DASH (Dynamic Adaptive Streaming over Http) è uno standard che definisce un modello per lo streaming video adattivo basato sul protocollo http.

Il DASH è stato finalizzato nel 2011 da MPEG con partecipazione di esperti nel settore e associazioni il cui compito è la definizione di standard. Quando in passato lo streaming video si è spostato su http utilizzando il TCP come protocollo di trasporto sono nate diverse piattaforme di streaming video adattivo, fra cui le principali Apple's HTTP Live Streaming, Microsoft's Smooth Streaming e Adobe's http Dynamic Streaming [7].

Le piattaforme appena citate erano proprietarie, quindi usavano formati e tecniche diversi, sebbene il loro funzionamento era simile e il fine era identico. Per utilizzare una di queste piattaforme era richiesto il rispettivo dispositivo client che le supportasse, essendo queste proprietarie. Per esempio un dispositivo della Apple disponeva del client relativo alla piattaforma Apple's HTTP Live Streaming, che un dispositivo non Apple non poteva possedere. Il DASH ha fornito uno standard universale e utilizzabile da qualunque dispositivo, fornendo un modello che ha permesso un'enorme crescita dell'utilizzo di servizi di streaming video.

1.4 DASH lato server

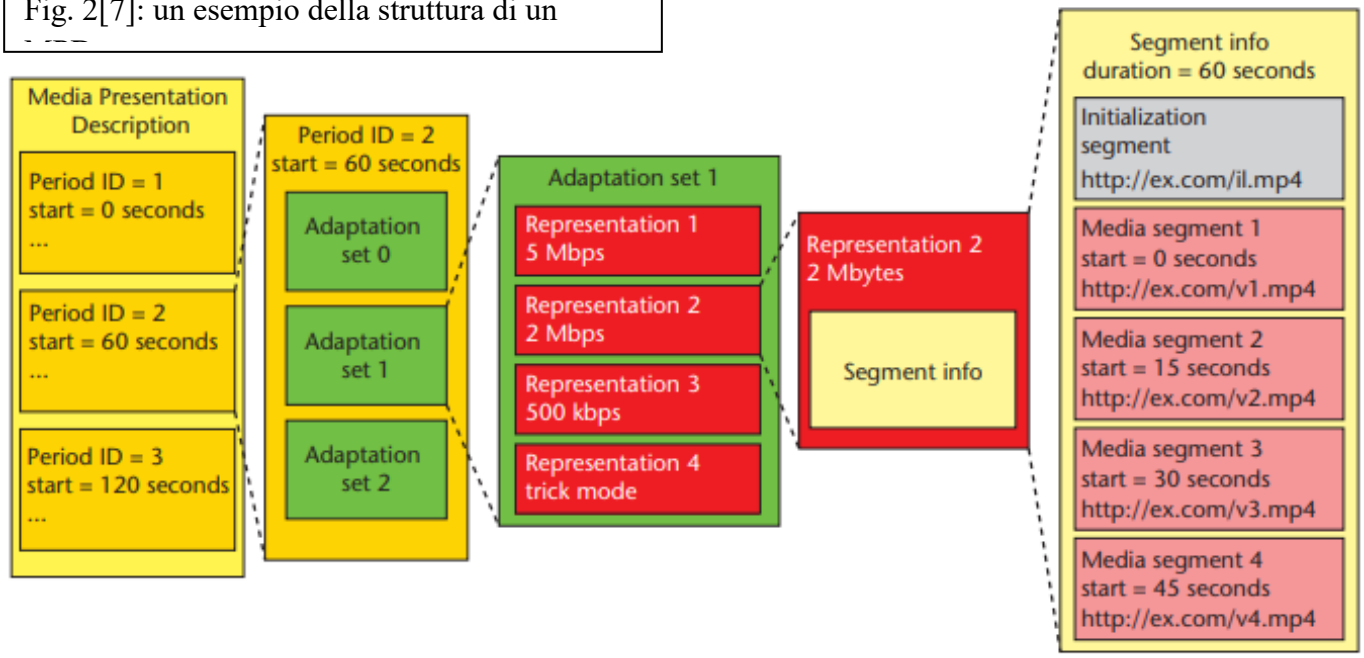
Lo standard DASH prevede che nel server siano presenti due elementi per ogni servizio di streaming offerto: il file MPD (Media Presentation Description) e i segmenti del relativo video, ovvero i bitstream (sequenze di bit che rappresentano il contenuto della risorsa multimediale).

Un contenuto multimediale può essere composto da più componenti (video, audio, testo) e ogni componente possiede le proprie rappresentazioni. Il file MPD, detto anche file indice o manifesto, ha il ruolo di fornire al client tutte le informazioni necessarie riguardanti la struttura della risorsa multimediale.

Il file MPD [7] è un file in formato XML organizzato in più periodi, ogni periodo rappresenta un arco temporale relativo alla risorsa multimediale, avente un istante d'inizio e una durata. I periodi sono divisi in adaptation sets, ognuno dei quali è relativo ad una componente multimediale (per esempio considerando il periodo 1: l'adaptation set 1 contiene informazioni del video relative al primo periodo, l'adaptation set 2 contiene informazioni dell'audio relative al primo periodo). Ogni adaptation set contiene le varie rappresentazioni relative alla componente multimediale interessata, ciascuna rappresenta lo stesso arco temporale di risorsa multimediale ma con un bitrate diverso. Infine ciascuna rappresentazione contiene gli indirizzi URL dei segmenti del video codificati con il rispettivo bitrate, quindi per esempio se ogni periodo è di 60 secondi e i segmenti sono lunghi 15 secondi, una rappresentazione contiene gli URL di 4 segmenti. Vale la pena notare che, essendo l'MPD un file xml, non contiene nessun bitstream della risorsa video, ma fornisce al client gli URL per accedere ad essi tramite delle richieste HTTP GET.

La fig. 2 [7] rappresenta un essempio di MPD. La risorsa è divisa in periodi, ognuno di lunghezza 60 secondi. Il periodo 2 ha tre adaptation set e l'adaptation set 1, relativo alla componente video, ha 4 rappresentazioni, con bitrate di 5 mbps, 2 mbps, 0.5 mbps e la trick mode (la modalità che permette di riavvolgere il video). La rappresentazione 2 contiene 4 segmenti, ognuno codificato a 2 mbps e lungo 15 secondi. Il primo è un segmento di inizializzazione, contenente dei metadati, mentre gli altri sono segmenti che rappresentano la risorsa multimediale; ogni segmento ha associato il rispettivo indirizzo URL.

Fig. 2[7]: un esempio della struttura di un



1.5 DASH lato client

La prima operazione che il client esegue per iniziare uno streaming video è richiedere al server il file MPD relativo alla risorsa multimediale selezionata. Una volta ottenuto l'MPD esso viene processato dall'MPD parser, un programma che sfruttando la struttura standard dell'MPD ne estrae tutte le informazioni, fra cui: durata del contenuto, tipi di risorse multimediali, risoluzioni disponibili, minima e massima larghezza di banda richiesta e indirizzi URL delle risorse da scaricare.

Per iniziare una sessione di streaming il client utilizza le informazioni ottenute tramite l'MPD per iniziare a richiedere in ordine, ad un opportuno bitrate, i segmenti di video tramite delle richieste HTTP GET. Una volta scaricati i segmenti il segment parser si occupa di estrarne il contenuto, operazione possibile visto che il formato dei segmenti è un formato standard definito dal DASH.

Durante la sessione di streaming video adattivo possono esserci dei cambi del bitrate selezionato con conseguenti variazioni della qualità della risorsa, tuttavia le euristiche che si occupano di fare adattazioni sono esterne al modello DASH, che si occupa solamente di fornire il formato del file MPD e il formato dei segmenti.

La fig. 3 [7] riassume quanto detto riguardo al modello proposto dal DASH, schematizzandone i componenti sia lato client che lato server. Sono colorati in rosso gli elementi che sono definiti e standardizzati dal DASH: il formato dei segmenti e dell'MPD (e come conseguenza i relativi parser

che ne estraggono i dati). Elementi come le euristiche adattive, il video player e il codec utilizzato sono esterni al DASH e la loro scelta è a discrezione della singola implementazione.

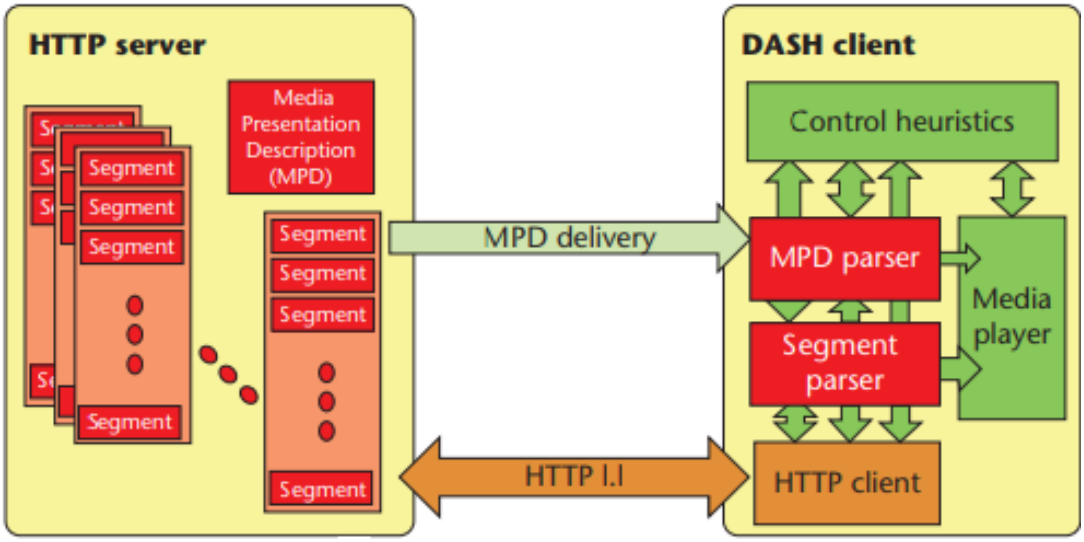


Fig. 3: schematizzazione dello standard DASH.

1.6 Il DASH e lo streaming dal vivo

Lo standard DASH supporta anche sessioni di streaming dal vivo, infatti il file MPD dispone di appositi campi in grado di fornire al client tutte le informazioni aggiuntive necessarie. Le problematiche [6] sono introdotte dal fatto che il contenuto è generato nel corso della sessione dal server, di conseguenza non è possibile disporre all'inizio di un file MPD che descriva l'intera struttura della risorsa multimediale.

- Il campo minimumUpdatePeriodMPD istruisce il client sulla lunghezza del periodo in cui usare l'MDP attuale, per cui alla scadenza di tale periodo è necessario richiedere al server il nuovo MDP e aggiornare il precedente. In genere questo intervallo di tempo è della durata di qualche segmento.
- Il campo availabilityStartTime contiene l'orario UTC relativo all'istante di inizio del primo periodo dell'MPD.
- Il campo availabiliyEndTime contiene l'orario UTC in cui l'evento dal vivo terminerà.

- Il campo minBufferTime garantisce che il client manterrà almeno il numero specificato di secondi di video nel playout buffer.

Il client dovrà quindi richiedere l'MPD molteplici volte, a differenza di una sessione di streaming on-demand in qui questo passaggio viene effettuato solamente una volta all'inizio. Il DASH non prevede che l'aggiornamento dell'MPD avvenga molto frequentemente, infatti secondo questa tecnica in esso possono essere descritti segmenti futuri, ovvero il cui contenuto deve ancora realmente manifestarsi, e forniti i relativi URL. Il client può infatti determinare qual è il segmento più recente contenente parti di video utilizzando l'orario attuale e sottraendolo all'orario riportato nel campo availabilityStartTime, capendo quindi quali segmenti sono stati popolati e quali invece sono relativi ad eventi che devono ancora accadere. Tale meccanismo permette utilizzare il DASH anche per supportare sessioni di streaming dal vivo.

Capitolo 2 - QUALITY OF EXPERIENCE (QoE)

2.1 Panoramica sulla QoE

La QoE indica la qualità relativa al servizio di streaming video usufruito che l'utente percepisce nel suo complesso. Non va confusa con la QoS (Qualità del Servizio), ovvero la qualità della rete a disposizione dell'utente, espressa da parametri come larghezza di banda, tasso di perdita dei pacchetti, latenza, jitter (variazioni considerevoli della latenza) [9].

I criteri di valutazione della QoE sono soggettivi, quindi determinati utenti possono dare più importanza a determinati fattori, altri utenti a fattori differenti. Nonostante questo, sono sicuramente individuabili elementi che in generale degradano la QoE percepita ed è possibile adottare delle soluzioni implementative per evitare il manifestarsi di tali fattori, o quantomeno minimizzarne la probabilità.

L'obiettivo di massimizzare la QoE è uno dei goal principali dei fornitori di servizi di streaming video, molto spesso complicato dal fatto che adottare una soluzione comporta un miglioramento di un determinato aspetto ma un peggioramento di un altro. Si parla quindi di trade-off, ovvero trovare una soluzione che rappresenta il giusto compromesso fra i vari fattori che possono incidere positivamente o negativamente sulla QoE.

Alcune applicazioni di streaming video necessitano di valori di latenza particolarmente bassi per fornire un servizio valido, pertanto questo costituisce un elemento aggiuntivo da tenere in considerazione per tali applicazioni. Nel capitolo 4 viene trattata la latenza in maniera approfondita e come questa incide significativamente sulla QoE nello streaming video a bassa latenza.

2.2 Fattori che influenzano la QoE

a) **Ritardo iniziale.**

   Esso è sempre presente durante uno streaming video perché è necessario che venga trasferita una certa quantità di dati affinché l'utente possa iniziare a visualizzare il video. Nella pratica per iniziare a riprodurre la risorsa si attende di più di quanto strettamente necessario, infatti si aspetta che venga scaricata una certa porzione di video nel playout buffer, misurata in numero di segmenti. Una maggiore quantità di video scaricata inizialmente comporta ad un ritardo iniziale maggiore, ma garantisce anche un maggior riempimento del buffer che riduce la probabilità che in futuro si manifesti uno stallo. Si ha quindi un trade-off relativo al numero di segmenti da scaricare prima di iniziare la visualizzazione. In generale il ritardo iniziale non degrada eccessivamente la QoE, infatti è

preferibile attendere per visualizzare il video piuttosto che avere un'interruzione di esso durante la visualizzazione.

b) Stalli (o freeze).

Come anticipato in precedenza, uno stallo è un interruzione imprevista della riproduzione del video e si tratta del fenomeno percepito più negativamente dagli utenti, va quindi cercato sempre di evitare, nei limiti del possibile [9]. La causa di uno stallo è l'insufficiente quantità di dati nel buffer per continuare la riproduzione del contenuto video. Quando si manifesta uno stallo tipicamente un simbolo rappresentante l'azione di caricamento compare a schermo, e si attende che il buffer venga riempito di una certa quantità di video. Si ha quindi un trade-off che riguarda quanti secondi di video attendere che vengano scaricati nel buffer e la durata dello stallo. Infatti un riempimento del buffer maggiore prima di riprendere la visualizzazione, come nel caso di ritardo iniziale, garantisce una minore probabilità che si manifestano futuri stalli, ma un'attesa eccessivamente lunga viene percepita negativamente da parte dell'utente. In genere l'attesa iniziale viene scelta più lunga dell'attesa che si ha durante uno stallo, in quanto l'utente medio è meno infastidito da un ritardo prima della visualizzazione della risorsa piuttosto che durante. Una scelta ragionevole è per esempio attendere che due segmenti vengano scaricati prima di iniziare la riproduzione del video e che un solo segmento venga scaricato per riprendere la riproduzione dopo uno stallo.

c) Adattazioni.

Come trattato precedentemente il concetto di streaming adattivo prevede delle scelte riguardanti lo scaricamento dei segmenti. Diversi bitrate sono disponibili per lo stesso segmento e degli algoritmi lato client decidono opportunamente che bitrate selezionare a seconda delle condizioni attuali. I cambi del bitrate selezionato sono dette adattazioni e sono uno strumento fondamentale per evitare gli stalli. In generale le adattazioni, sebbene siano necessarie, comportano ad un degrado della QoE, che può risultare contenuto se eseguite secondo determinati criteri.

Frequenza e ampiezza delle adattazioni indicano rispettivamente quanto frequentemente essere avvengono e quanto cambiano la qualità rispetto a quella precedente. Se le adattazioni avvengono molto frequentemente, per esempio ogni uno o due secondi, l'utente le percepisce in maniera molto negativa ed è preferibile tenere costantemente una qualità più bassa. Per quanto riguarda l'ampiezza invece, cambi di qualità con ampiezza contenuta sono difficilmente percepibili dall'osservatore e non degradano quindi la QoE. Le

adattazioni quindi, se possibile, devono avvenire con una frequenza opportuna e cambiare la qualità del video gradualmente.

d) **Qualità del video.**

Infine la qualità del contenuto video visualizzato gioca un ruolo cruciale nella QoE. Naturalmente l'utente preferisce visualizzare il video a qualità alte, quindi se la qualità dovesse essere considerata bassa dall'utente questo fattore inciderebbe molto negativamente sulla QoE. La qualità di un video è definibile matematicamente tramite metriche di valutazione oggettive, per esempio PSNR e SSIM (vedi sezione 3.3), siccome questa è strettamente legata alla dimensione del bitrate, approssimativamente si può indicare la qualità tramite quest'ultimo. La qualità di un video è definita da diversi fattori, trattati in seguito come "dimensioni di adattazione".

2.3 Dimensioni di adattazione

Una rappresentazione di un contenuto video può differire da un'altra in base a tre aspetti, ovvero quantizzazione, risoluzione spaziale e numero di immagini al secondo. Questi tre aspetti sono detti dimensioni di adattazione, è infatti possibile variare il bitrate operando su una o più dimensioni. Queste tre dimensioni nel complesso costituiscono la qualità di un video.

a) **Quantizzazione.**

La quantizzazione è un operazione fondamentale di ogni codificatore con perdite, essa è esprimibile da un parametro che indica il livello di distorsione che verrà introdotto dal codificatore nelle immagini del video. Denominando QP il parametro di quantizzazione utilizzato nella codifica, ponendo QP = 0 si ha una codifica senza perdite, mentre per esempio ponendo QP = 50 si hanno delle immagini di pessima qualità. Al crescere di QP si ha una diminuzione del bitrate, e scegliendo un valore di QP appropriato è possibile ridurre notevolmente il bitrate producendo comunque delle immagini la cui qualità è percepita positivamente dagli osservatori.

Secondo [9] infatti si è sperimentalmente dedotto che, partendo da QP = 0, la qualità percepita diminuisce molto lievemente all'aumentare iniziale di QP in quanto è difficile osservare le distorsioni introdotte. Quando QP raggiunge un valore sufficientemente alto invece la qualità percepita inizia a diminuire in maniera significativa all'ulteriore aumento di QP, perché le distorsioni introdotte diventano sempre più evidenti. Va pertanto evitato l'utilizzo di valori di QP troppo alti, in quanto peggiorano la qualità del video in maniera eccessiva. Il valore del parametro QP è opportunamente scelto dal codec, l'utente non ha quindi la possibilità di personalizzarlo.

b) Risoluzione spaziale.

La risoluzione spaziale indica rispettivamente il numero di pixel per riga e per colonna delle immagini che costituiscono un video. Le risoluzioni più utilizzate negli streaming video sono la risoluzione HD (1280 x 720 pixel), full HD (1920 x 1080 pixel) e 4K (o ultra HD 3480 x 2160 pixel). Sebbene la risoluzione spaziale indica la dimensione delle immagini di un video, gli schermi dei dispositivi applicano operazioni di downscaling o upscaling per adattare la dimensione delle immagini alla dimensione reale dello schermo. Per esempio non si può visualizzare una risorsa in 4K se la risoluzione dello schermo è minore, ma tale risorsa verrà visualizzata nella massima risoluzione supportata dallo schermo del dispositivo (downscaling). Se invece la risoluzione della risorsa video è per esempio HD, ma dispongo di uno schermo full HD, la risoluzione del video verrà adattata alla dimensione dello schermo (upscaling), con una conseguente introduzione di distorsioni.

Uno streaming video adattivo dovrebbe adattare la risoluzione del video alla risoluzione dello schermo dei dispositivo che lo visualizza, in modo da evitare la trasmissione di informazioni che poi verranno scartate e l'introduzione di distorsioni che degradano la QoE. Utilizzare risoluzioni video considerevolmente più basse a quella dello schermo dell'utente va evitato, infatti rende particolarmente evidente l'introduzione di artefatti nelle immagini.

La risoluzione spaziale è particolarmente importante in video in cui è necessario osservare dettagli di piccole dimensioni, per esempio un video ripreso in 4K che mostra dettagliatamente un paesaggio, in cui la componente più importante della qualità è giocata dalla risoluzione spaziale. Generalmente durante la visualizzazione di un video l'utente ha la possibilità di scegliere manualmente la risoluzione spaziale, qualora avesse bisogno per esempio di osservare le immagini del video nel dettaglio.

c) Immagini al secondo

Il numero di immagini al secondo visualizzate durante la riproduzione di un video (fps, frame per second) determina la fluidità dei movimenti rappresentati nel video. Un alto numero di fps comporta ad un video in cui l'illusione del movimento è perfetta e non si percepisce la discontinuità delle immagini, un basso numero di fps comporta invece ad un video i cui movimenti sono discontinui. Un tentativo di adattazione che riduce il numero di immagini al secondo in generale comporta ad una diminuzione considerevole della QoE. Nello specifico fare adattazione sui fps ha un impatto differente in base al tipo di contenuto rappresentato nel video: in video che rappresenta scene molto dinamiche è fondamentale avere un adeguato numero di fps, mentre in video statici dove c'è poco movimento i fps non

giocano un ruolo importante. Due esempi opposti possono essere un video che rappresenta un videogioco sparatutto, in cui la visuale si sposta continuamente e un video che inquadra una persona che sta parlando, in cui la telecamera è sempre ferma e i movimenti della persona sono minimi.

L'impatto sulla QoE del numero di immagini al secondo dipende quindi dal contenuto video nello specifico, fattore di cui si deve tener conto nei processi di adattazione in modo da non gravare eccessivamente sulla dimensione di adattazione più rilevante.

Per adattare il bitrate di un video durante la sua visualizzazione è possibile operare su una o più delle dimensioni di adattazione descritte, con una conseguente variazione della qualità causata dal variare di una o più dimensioni. Si è sperimentato [9] che in generale fare adattazione considerando più dimensioni produce una qualità percepita maggiore rispetto che su una sola dimensione, a parità di bitrate del video.

Data una particolare risorsa video e fissato il bitrate esiste una combinazione ottima [9] dei valori delle dimensioni tale che massimizza la qualità del video percepita dell'utente. Per trovare una valida ponderazione delle dimensioni è necessario analizzare la risorsa video nel dettaglio per dedurre l'importanza delle singole dimensioni, e quindi per ogni bitrate reso a disposizione regolare ad hoc tali dimensioni. Quest'operazione è chiaramente possibile nel caso di risorse video on-demand, in cui si ha tutto il tempo necessario per eseguire analisi di questo tipo. Per quanto riguarda invece contenuto video dal vivo, quindi generato sul momento, è necessario ponderare le dimensioni in maniera che in generale sia prodotto un video di buona qualità.

Capitolo 3 - LA CODIFICA VIDEO

3.1 La codifica con o senza perdite

Esistono due tipi di codec: con perdite (lossy) e senza perdite (lossless). Codificare, o comprimere, una risorsa multimediale produce una rappresentazione di essa che utilizza meno bit dell'originale e tramite il processo di decodifica, o decompressione, è possibile visualizzare una versione più o meno alterata della risorsa originale.

Il rapporto di compressione di una codifica è dato dal rapporto fra bit usati per rappresentare il contenuto codificato e il numero di bit con cui è rappresentato il contenuto all'ingresso. Nella codifica senza perdite il contenuto ottenuto eseguendo la decodifica è identico al contenuto originale, non si hanno quindi perdite di qualità, ma nemmeno un elevato rapporto di compressione. Nella codifica con perdite il contenuto ottenuto eseguendo la decodifica non coincide con l'originale, infatti vengono introdotte delle distorsioni, che comportano ad una perdita di qualità, ma si ha un ottimo rapporto di compressione (due ordini di grandezza nei contenuti video).

Per eseguire la codifica di risorse video si usano quasi esclusivamente tecniche di codifica con perdite, in quanto permettono di rappresentare il contenuto con molti meno bit rispetto alla risorsa originale e allo stesso tempo è possibile avere una buona qualità percepita da parte dell'utente. Regolando determinati parametri di un codificatore con perdite è possibile regolare la distorsione introdotta dal processo di codifica, e quindi la qualità del video. Il numero di bit al secondo necessari per rappresentare una risorsa video codificata è detto tasso di codifica (o bitrate), e in base a parametri di codifica si può scegliere se rappresentare la risorsa video con un alto o con un basso bitrate, con una conseguente migliore o peggiore qualità.

Una misura meno usata per descrivere il tasso di codifica sono i bit per pixel (bpp), ovvero il rapporto fra il numero totale di bit usati per rappresentare la risorsa codificata e il numero di pixel totali (pixel per immagine moltiplicato per numero di immagini) del video.

3.2 Nozioni di base sulla codifica

Lo scopo di questa sezione è di fornire al lettore una visione generale e semplificata del processo di codifica, per introdurre elementi che verranno utilizzati in seguito nell'elaborato.

Le immagini di un video sono rappresentate quasi esclusivamente nello spazio YCbCr, ovvero ogni pixel è descritto da tre elementi: luminanza, crominanza del rosso e crominanza del blu. La rappresentazione YCbCr 4:2:0 esegue un sotto campionamento della crominanza, utilizzando sia

per la componente Cb che per la componente Cr un quarto dei bit usati per la luminanza, che è la componente che contiene più informazione. Tale rappresentazione è largamente preferita alla rappresentazione RGB perché permette di rappresentare un'immagine con metà dei bit rispetto ad essa, inoltre permette ai codec di essere notevolmente più efficienti: nella rappresentazione RGB le tre componenti hanno la stessa quantità di informazione, mentre la rappresentazione YCbCr è più sparsa del momento che la luminanza costituisce la componente più importante.

Le varie immagini che compongono un video possono essere codificate in modalità 'intra' o in modalità 'inter', quest'ultima può consistere in una codifica predittiva o bidirezionale. Le immagini intra (I) sono codificate utilizzando esclusivamente informazioni contenute nell'immagine stessa, mentre le immagini predittive (P) e bidirezionali (B) possono avere come blocchi di riferimento blocchi di altre immagini.

Nel processo di codifica un'immagine inter questa è suddivisa in blocchi, e ad un blocco può essere associato un vettore di stima del movimento (MV, Motion Vector). Un MV punta ad un altro blocco simile al blocco considerato, detto blocco di riferimento. Dato un blocco è possibile eseguire una predizione spaziale, quindi cercare un blocco simile fra i blocchi spazialmente vicini al blocco corrente, oppure una predizione temporale, ovvero cercare un blocco simile nelle immagini immediatamente precedenti e/o successive. L'obiettivo di una predizione è che il blocco puntato dal MV sia sufficientemente simile al blocco corrente in modo che la differenza fra i due, o residuo, sia un segnale sparso, ovvero un blocco in cui la maggior parte dell'informazione è trascurabile o addirittura nulla. E' quindi possibile rappresentare solo la parte di informazione non trascurabile del residuo, con una conseguente efficienza nella codifica.

Le immagini intra, essendo codificate in maniera autonoma, costituiscono dei punti di accesso di un video, in quanto non hanno bisogno di informazioni relative ad altre immagini per essere decodificate. Una caratteristica importante delle immagini intra è che, essendo limitate nell'utilizzo di tecniche di predizione ai soli blocchi dell'immagine stessa, sono rappresentate con molti più bit delle immagini codificate in modalità inter e per questo le immagini intra giocano un ruolo estremamente importante nella latenza, argomento approfondito nel capitolo successivo.

Un video è composto da una ripetizione periodica di GOP (Group Of Pictures), ovvero una sequenza di immagini che inizia con un'immagine I e finisce l'immagine prima della successiva immagine I. L'inizio di un GOP rappresenta quindi un punto di accesso della risorsa video. La lunghezza dei GOP è a discrezione dell' implementazione. La fig. 4 rappresenta una

schematizzazione di un GOP, una freccia uscente da un immagine X ed entrante in un immagine Y indica che i MV dei blocchi di X fanno riferimento ai blocchi di Y.

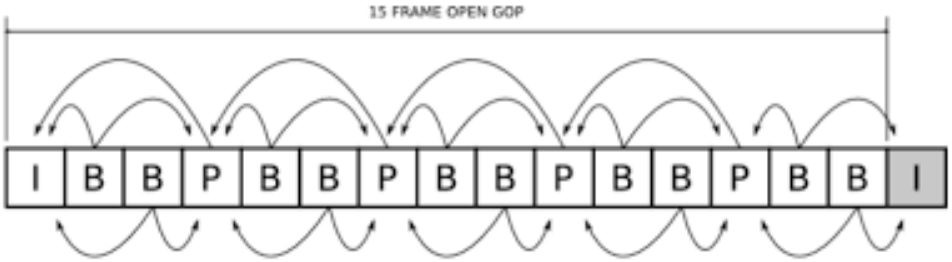


Fig. 4 [16]: un GOP di dimensione 15.

3.3 Metriche di valutazione MOS, MSE, PSNR, SSIM

E' importante avere a disposizione dei criteri per valutare la qualità di una risorsa video, per capire se la codifica ha introdotto una quantità di distorsione accettabile o meno. Esistono due tipi di criteri di valutazione: soggettivi e oggettivi.

Una tecnica di valutazione soggettiva consiste nel far visualizzare ad un opportuno insieme di osservatori un'immagine o un video, e far assegnare ad ogni osservatore un punteggio relativo alla qualità soggettivamente percepita. Tali punteggi verranno poi mediati per ottenere il MOS (Medium Opinion Score), che descrive la qualità percepita mediamente della risorsa multimediale.

I criteri oggettivi sono invece definiti da una funzione matematica che confronta il video originale con il video ottenuto tramite il codec. I più comuni criteri oggettivi sono l'MSE, ovvero l'errore quadratico medio e il PSNR, ovvero rapporto segnale rumore di picco. MSE e PSNR sono spesso usati come tecniche di valutazione perché rispecchiano sufficientemente bene la percezione di un osservatore umano e sono relativamente facili da calcolare, ovviamente utilizzando appositi software.

L'MSE si calcola mediante la seguente formula [8]:

$$MSE = \frac{1}{M\,N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |I(i,j) - K(i,j)|^2$$

Dove le due immagini messe a confronto sono entrambe di dimensione M x N. L'immagini originale è chiamata I, quella compressa K. Un pixel è definito da un indice di riga *i* e un indice di

colonna *j*. I(*i,j*) – K(*i,j*) costituisce la differenza fra il pixel in posizione (*i,j*) dell'immagine originale e lo stesso pixel nell'immagine compressa.

Il PSNR si calcola mediante la seguente formula:

$$PSNR = 10 \cdot log_{10}(\frac{(2^b - 1)^2}{\sqrt{MSE}})$$

Il PSNR è espresso in decibel (dB) e indica la somiglianza fra l'immagine compressa e quella originale, più alto è il PSNR minore è quindi la distorsione introdotta dal codec. Il termine $(2^b - 1)$ indica il numero massimo di valori possibili per la componente di luminanza, dove b è il numero di bit utilizzati per rappresentare la luminanza. $(2^b - 1)^2$ è quindi il massimo valore possibile, o valore di picco, della potenza del segnale. Il PSNR è quindi definito come il rapporto tra la massima potenza del segnale di luminanza e la potenza di rumore, espresso in decibel. Per fornire un riferimento relativo al valore del PSNR si può dire in grandi line che un PSNR=20 dB corrisponde ad una qualità scarsa, un PSNR=30 dB corrisponde ad una qualità media e infine un PSNR=40 dB corrisponde ad una qualità alta.

Il risultato espresso dal PSNR potrebbe distaccarsi dalla percezione umana della qualità di un'immagine. Per esempio se nell'immagine compressa un oggetto è traslato di pochi pixel il PSNR verrà significativamente influenzato da questo errore, mentre l'occhio umano percepirebbe tale traslazione come poco rilevante.

La metrica SSIM (Structural Similarity Index Measure) misura la somiglianza fra l'immagine di riferimento e l'immagine compressa in maniera da rispecchiare la percezione dell'occhio umano. A differenza del PSNR l'SSIM considera la struttura complessiva dell'immagine e non esclusivamente la differenza fra ogni singolo pixel. L'ISSM divide le due immagini in regioni ed esegue il confronto fra due regioni corrispondenti sotto tre aspetti: luminanza, contrasto e struttura. Questi tre elementi sono calcolati tramite formule matematiche e il valore dell'SSIM relativo all'intera immagine si ottiene mediando il risultato di ogni regione dell'immagine.

I tre indici di luminanza, contrasto e struttura relativi ad una regione di un'immagine *x,* contenente N pixel sono calcolati come segue [13]:

- Luminanza – la luminanza di una regione *x* è calcolata mediando il campione di luminanza x_i di ogni pixel fra gli N pixel della regione, secondo la formula $\mu_x = \sum_{i=1}^{N} x_i$

- Contrasto - il contrasto è espresso dalla deviazione standard dei valori di luminanza di ogni pixel della regione, secondo la formula $\sigma_x = \left(\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_i)^2\right)^2$
- Struttura – la struttura è calcolata secondo la formula $s_x = \frac{x-\mu_x}{\sigma_x}$

Le tre funzioni di confronto sono applicate a due regioni *x* e *y* corrispondenti (dell'immagine riferimento e dell'immagine compressa) in base ai tre indici descritti.

Sia b il numero di bit usati per rappresentare un pixel, la costante $L = 2^b - 1$ rappresenta il numero massimo di possibili valori di luminanza per pixel. $k_1 = 0.01$ e $k_2 = 0.03$ sono valori costanti di default. Le costanti c1, c2 e c3, usate nelle funzioni di confronto, sono quindi rispettivamente calcolate come segue [13]:

$c_1 = (k_1\, L)^2 \qquad c_2 = (k_2\, L)^2 \qquad c_3 = \frac{c_2}{2}$

- Funzione di confronto della luminanza - $l(x,y) = \frac{2\,\mu_x\,\mu_y + c_1}{\mu_x{}^2 + \mu_y{}^2 + c_1}$
- Funzione di confronto del contrasto - $c(x,y) = \frac{2\,\sigma_x\,\sigma_y + c_2}{\sigma_x{}^2 + \sigma_y{}^2 + c_2}$
- Funzione di confronto della struttura - $s(x,y) = \frac{\sigma_{xy} + c_3}{\sigma_x\,\sigma_y + c_3}$

  Dove $\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y)$

Infine il valore dell'SSIM fra le regioni x e y è dato dalla formula [13] :

$$SSIM(x,y) = [l(x,y)^{\alpha}] \cdot \left[c(x,y)^{\beta}\right] \cdot [s(x,y)^{\gamma}]$$

Dove $\alpha, \beta, \gamma > 0$ indicano l'importanza di ciascun indice di confronto. Per semplificare la formula si pone in genere $\alpha = \beta = \gamma = 1$.

3.4 Il codec VVC

Lo scopo di questa sezione è quello di fornire al lettore delle conoscenze di base riguardanti il codec VVC, necessarie per comprendere l'utilizzo della tecnica GDR e le conclusioni tratte dalle analisi sperimentali.

Nel VVC l'unità di base di un'immagine è la CTU (Coding Tree Unit), ovvero la rappresentazione di una regione quadrata dell'immagine, contenente i campioni di luminanza e crominanza che la costituiscono. Una CTU è rappresentabile tramite una struttura ad albero le cui foglie sono dette CUs (Coding Units), ovvero regioni di dimensione variabile contenute nella CTU.

VVC offre la possibilità di definire delle linee che rappresentano dei confini virtuali (virtual boundaries) all'interno di un'immagine. Le operazioni effettuate dai filtri in-loop sono disabilitate se queste ultime operano attraverso la linea di confine virtuale, il loro scopo è infatti di delineare delle discontinuità presenti nelle immagini. Per esempio in un'immagine che rappresenta nella metà a destra la visuale anteriore di un'auto e nella metà a sinistra la visuale posteriore, una linea di confine virtuale sarà posizionata in mezzo per evitare di confondere elementi che sono fisicamente vicini dell'immagine ma che non hanno nulla in comune.

Gli strumenti di codifica offerti da VVC sono moltissimi e non è lo scopo di questo documento analizzarli nel dettaglio, pertanto verranno trattate come esempio solo alcune delle tecniche offerte.

- Codifica intra.
  Nel caso di codifica in modalità intra, data una CU corrente su cui si sta operando, il relativo blocco di riferimento puntato dal MV è selezionato solamente fra i blocchi contenuti nell'immagine corrente. Una tecnica è per esempio la predizione spaziale, secondo la quale il blocco di riferimento è cercato fra i blocchi spazialmente vicini alla CU corrente.
- Codifica inter.
  Nel caso di codifica in modalità inter il blocco di predizione di una CU è costituito da uno o due blocchi relativi alle immagini di riferimento, puntati dai relativi MVs (nel caso di due blocchi, il blocco di predizione viene prodotto considerando le caratteristiche di entrambi i due blocchi di riferimento). A differenza della codifica inter, come detto in precedenza, i blocchi di riferimento possono essere contenuti in altre immagini.
  Fra le molte modalità inter offerte da VVC è presente la "spatial MVP", ovvero una predizione spaziale del vettore i movimento. Selezionata la CU corrente la spatial MVP consiste nel predirne il MV basandosi sui MVs delle CU vicine, ipotizzando che fra i due blocchi di riferimento ci sia una correlazione dovuta alla vicinanza spaziale delle due CU. In questa modalità di codifica viene quindi assegnato alla CU corrente una copia del MV di una CU vicina selezionata, il blocco di riferimento della CU corrente è quindi vicino al blocco di riferimento della CU scelta per la predizione (in quanto è ottenuto tramite una traslazione del MV). Se per esempio la CU corrente utilizza il MV della CU all'immediata sinistra, il blocco di riferimento della CU corrente è il blocco all'immediata destra del blocco puntato dalla CU usata per fare predizione.
  Un'altra tecnica di codifica è "l'affine merge", in cui la CU selezionata è divisa in sotto blocchi di dimensione 4 x 4 pixel. Ad ogni sotto blocco è assegnato il proprio MV, calcolato automaticamente basandosi su una media fra due o tre MV di CU vicine.

La fig. 5 [11] rappresenta graficamente la suddivisione in sotto blocchi della CU corrente e i relativi MV assegnati applicando la tecnica di affine merge.

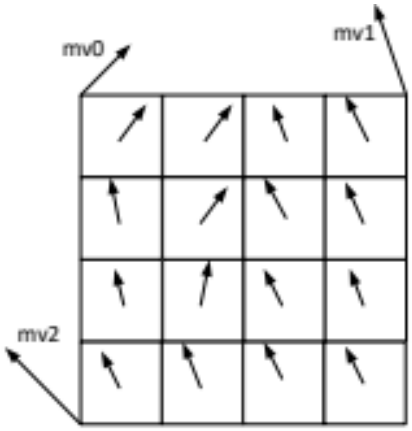


Fig. 5 [11]: rappresentazione grafica di un “affine merge”. Mv0, mv1 e mv2 sono i MV delle CU vicine utilizzati per calcolare i MV dei sotto blocchi.

Capitolo 4 - LO STREAMING VIDEO A BASSA LATENZA

4.1 Il concetto di latenza

Quando si guarda uno streaming dal vivo in broadcast la risorsa video arriva al dispositivo client con un ritardo di circa 30 secondi rispetto all'evento reale che sta rappresentando. Se per esempio si invia un messaggio alla persona che sta parlando in live streaming, chiedendo di salutarci per nome, solo dopo 30 secondi riceveremo tale saluto a video.

Questo ritardo è la latenza, ovvero il tempo che trascorre da quando un evento dal vivo ha luogo a quando il client lo visualizza. La latenza è composta da 5 componenti [10], ovvero: tempo di codifica, tempo di trasmissione (ovvero il tempo impiegato dai pacchetti contenenti segmenti di video ad essere consegnati al dispositivo client), ritardo iniziale di buffering, tempo di decodifica e tempo di ritardo iniziale dell'output (dovuto ad un'eventuale operazione di riordinamento delle immagini). Quest'ultimo è pari a zero se non è necessario il riordinamento, quindi se le immagini sono state trasmesse nello stesso ordine con cui devono essere visualizzate, come accade generalmente in applicazioni che necessitano di minimizzare la latenza.

La fig. 6 [10] rappresenta le varie componenti della latenza e come queste si sommano creando il ritardo fra client e server. I rettangoli rappresentano le immagini inviate e la loro altezza indica la dimensione in bit. Si osserva che dall'inizio dello streaming si ripete periodicamente un'immagine di dimensione maggiore delle altre, che quindi comporta ad un ritardo maggiore, colorata in rosso. Queste sono immagini codificate in maniera intra, e come verrà trattato in seguito esse giocano un ruolo fondamentale per quanto concerne la minimizzazione della latenza.

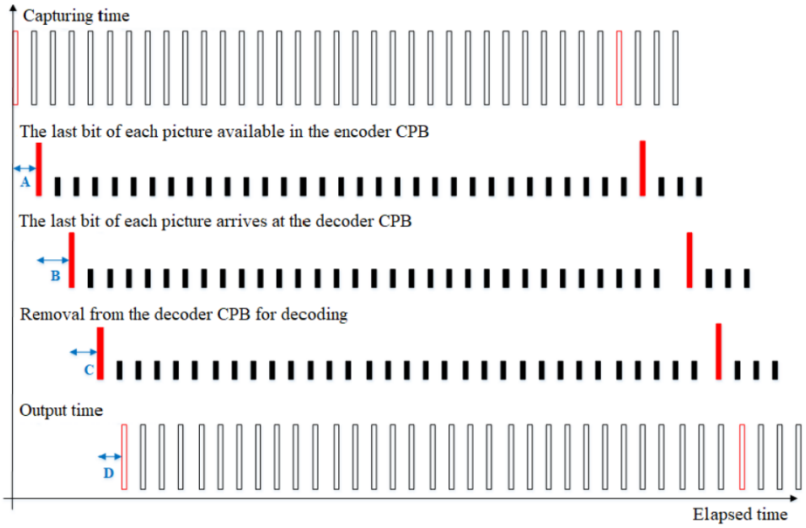


Fig. 6 [10]: le componenti che costituiscono la latenza.

A: tempo di codifica

B: tempo di trasmissione

C: ritardo iniziale di buffering

D: tempo di decodifica

4.2 QoE Applicazioni a (ultra) bassa latenza

Per determinati tipi di servizi streaming, per esempio lo streaming on-demand fornito dalle OTT, la latenza non gioca un ruolo importante sulla qualità del servizio, se non per il ritardo iniziale di visualizzazione.

Altri tipi di applicazioni invece hanno requisiti di bassa latenza (low latency) o ultra-bassa latenza (ultra-low latency). La differenza fra bassa latenza e ultra-bassa latenza sta nella dimensione di tale ritardo. Si parla di latenza bassa se essa è nell'ordine del secondo, mentre di latenza ultra-bassa se essa è nell'ordine di qualche centesimo di secondo [14].

Un esempio di applicazioni a bassa latenza sono quelle di video conversazione, come videochiamate o videoconferenze. Questo tipo di servizi necessitano di bassa latenza per garantire una QoE adeguata, infatti la loro funzione è di simulare una conversazione che avviene in tempo reale, quindi il ritardo con cui avviene tale comunicazione deve essere limitato nell'ordine di qualche decimo di secondo per non essere percepito.

Altri tipi di servizi hanno requisiti ancora più stretti in termini di latenza, come una sessione gioco sul cloud (cloud gaming), in cui la latenza ottima varia fra 20 e 40 millisecondi. Una latenza superiore a 100 millisecondi inciderebbe molto negativamente sulla QoE, in quanto il ritardo fra le azioni fatte dall'utente e l'output prodotto a schermo dal videogioco sarebbe facilmente percepibile e l'esperienza di gioco sarebbe di conseguenza pessima.

Per ridurre i tempi di codifica e decodifica una soluzione è usufruire di un hardware più performante, mentre per ridurre il tempo di trasmissione utilizzare una rete ad alta larghezza di banda (come la fibra ottica). Gli utenti tuttavia dispongo di risorse limitate, quindi è necessario adottare delle soluzioni che non fanno affidamento ai mezzi disponibili degli utenti, ma che strutturino lo streaming video con meccanismi che riducono la latenza a prescindere da essi.

4.3 Possibili soluzioni per ridurre la latenza

La componente della latenza su cui è più proficuo operare è il tempo iniziale di buffering. Le variazioni del bitrate causano delle variazioni del tempo di trasmissione, rendere più uniforme il bitrate nel tempo comporta quindi ad una riduzione del ritardo.

Senza adottare nessuna tecnica infatti il bitrate non è uniforme, in quanto ogni volta che è tramessa un'immagine codificata in modalità intra essa causa dei picchi nel bitrate, dal momento che tali immagini necessitano di molti più bit (un ordine di grandezza) delle immagini codificate in

modalità inter per essere rappresentate. Le immagini codificate inter invece sono generalmente uniformi in termini di quantità di bit per rappresentarle, è quindi necessario adottare delle misure per eliminare i picchi nel bitrate causati dalle immagini codificate intra.

Una possibile soluzione consiste nel regolare il fattore di quantizzazione (QP) con cui sono codificate le immagini intra, in modo che il numero di bit necessari a rappresentarle sia simile alle immagini inter. Quest'approccio causa però una notevole riduzione di qualità delle immagini intra, facilmente osservabile dall'utente, in quanto si crea un'enorme differenza di qualità fra immagini consecutive. Ripetendo questa operazione per ogni immagine intra l'osservatore noterebbe tali sbalzi di qualità periodicamente e frequentemente. Per questo motivo questa soluzione non risulta essere efficace dal punto di vista della QoE in quanto degrada eccessivamente la qualità del video.

Un'altra soluzione è quella di saltare il processo di codifica di poche immagini che seguono l'immagine codificata intra. Questo non comporta ad una riduzione del ritardo causato da quest'ultima, ma dopo che viene visualizzata a schermo l'immagine codificata intra, le successive immagini decodificate possono essere mostrate a schermo prima, senza rispettare il tempo reale di cattura fra l'immagine codificata intra e queste. Tuttavia questa tecnica causa un'alterazione forzata nel dominio del tempo, in quanto viene velocizzata la visualizzazione delle immagini successive ad un'immagine codificata intra.

Entrambe le tecniche citate introducono gravi problematiche per quanto riguarda la QoE: la prima causa una ripetizione periodica di immagini con qualità visibilmente minore, la seconda altera artificiosamente le tempistiche di rappresentazione a schermo delle immagini.

La tecnica GDR (Gradual Decoding Refresh) permette di rendere sufficientemente uniforme il bitrate nel tempo tramite appositi meccanismi di codifica, senza introdurre le problematiche citate in precedenza.

4.4 Gradual Decoding Refresh (GDR)

GDR è una funzionalità supportata dal codec VVC progettata per essere utilizzata da applicazioni con requisiti di (ultra) bassa latenza.

La tecnica GDR, invece di codificare determinate immagini come intra e trasmetterle al client, prevede di distribuire aree codificate intra in più immagini, rimuovendo quindi picchi nel bitrate causati dalla trasmissione di immagini interamente codificate in maniera intra.

La fig. 7 [11] schematizza il concetto di GDR. Un'immagine GDR con un'area codificata forzatamente in modalità intra viene trasmessa all'istante POC(n), ovvero l'immagine numero "n" della riproduzione del video, e altre aree codificate forzatamente in maniera intra sono distribuite nelle N immagini successive, da sinistra verso destra.

In POC(n+N-1) viene trasmessa l'immagine con l'ultima aerea rimanente codificata intra (a destra) e tale immagine è detta punto di ripristino. L'intervallo che va da POC(n) a POC(n+N-1) è chiamato periodo GDR, ed è costantemente ripetuto durante lo streaming video.

Le immagini GDR sono costituite da due aree la cui distinzione è estremamente importante: la "clean area" e la "dirty area". La clean aerea di un'immagine è la regione (rettangolare) a sinistra dell'area forzata alla codifica intra, mentre la dirty area è la regione (rettangolare) a destra di quest'ultima. Il confine fra le due aree è marcato da una retta verticale, detta "virtual boundary" nel codec VVC. La tecnica GDR prevede un progressivo aggiornamento delle immagini relative al periodo GDR corrente man mano che nuove immagini GDR con nuove aree codificate intra vengono trasmesse. La clean area di ogni immagine è aggiornata (detta infatti anche "refreshed area"), mentre la dirty area non lo è (detta anche "non-refreshed area"). Grazie al progressivo aggiornamento delle immagini durante la trasmissione l'area aggiornata delle immagini si espande fino a che nell'immagine a POC(n+N-1) l'intera immagine è aggiornata.

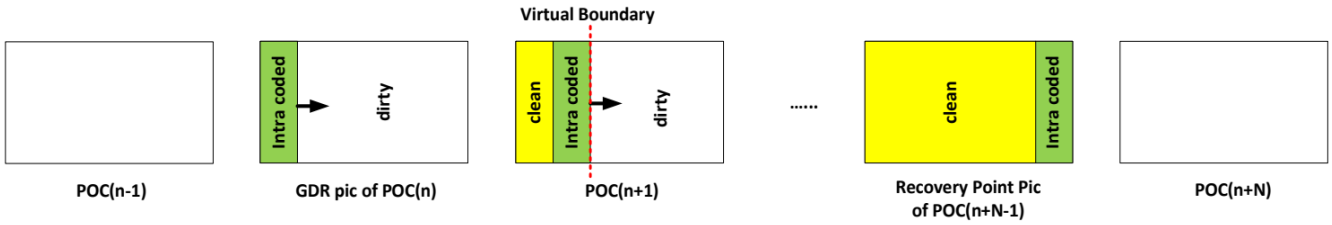


Fig. 7: rappresentazione schematica di un periodo GDR di N immagini. Le regioni codificate intra (in verde) espandono le clean area (in giallo) delle immagini progressivamente, grazie all'operazione di aggiornamento delle immagini. La linea di confine virtuale si sposta verso destra, riducendo la dirty area (bianco), fino a che nell'immagine che costituisce il punto di recupero l'intera immagine è aggiornata.

Affinché la tecnica GDR produca una corretta decodifica del video è fondamentale che si abbia una perfetta corrispondenza nei punti di ripristino, ovvero che tutte le immagini relative ad un periodo GDR ricostruite tramite gli aggiornamenti progressivi siano identiche fra codificatore e decodificatore nei punti di ripristino. Le mancata corrispondenza delle immagini nei punti di

ripristino è detta leak e comporta ad errori nella decodifica con conseguenti errori nella rappresentazione delle immagini.

4.5 Evitare i GDR leaks

I vari strumenti di codifica forniti dal codec VVC sono basati sull'utilizzare informazioni di CUs vicine spazialmente e/o temporalmente alla CU selezionata per fare operazioni di predizione e rendere efficiente la codifica. Si introducono leaks ogni qualvolta una CU contenuta nella clean area dell'immagine considerata utilizza qualsiasi tipo di informazione contenuta nella dirty area di un'immagine. Se quindi un blocco riferimento contiene dei pixel nella dirty area dell'immagine riferimento, si ha una contaminazione della clean area con la conseguente introduzione di leaks.

Sebbene il codec VVC supporti la tecnica GDR, è compito dello specifico codificatore utilizzare correttamente la tecnica GDR evitando la creazione di leaks. Gli accorgimenti da adottare al fine di evitare l'introduzione di leaks dipende dalla tecnica di codifica utilizzata, con la caratteristica comune che ogni immagine GDR possiede un virtual boundary per separare clean area da dirty area.

Verranno presentati in seguito alcuni degli strumenti di codifica forniti dal codec VVC e verranno analizzate le tecniche da utilizzare per prevenire che tali strumenti causino l'introduzione di leaks GDR (le tecniche citate in seguito sono illustrate nella sezione 3.4 "Il codec VVC").

- Predizione spaziale intra

  Al fine di evitare leaks, è necessario che i blocchi di riferimento puntati dal MV della CU corrente non contengano alcun pixel nella dirty area dell'immagine corrente.
- Spatial MVP.

  Per eseguire correttamente la tecnica di spatial MVP con GDR è necessario rimuovere dalla lista delle CUs candidate tutte le CUs contenute nella dirty area. Oltre a questo è necessario un ulteriore accorgimento, infatti questa tecnica prevede l'applicazione del MV di una CU vicina (chiamata CU *x*) alla CU corrente, con una conseguente traslazione del blocco di riferimento. A causa di questa traslazione, se il blocco di riferimento era sufficientemente vicino alla dirty area dell'immagine di riferimento, il blocco ottenuto per la CU corrente sarà nella dirty area dell'immagine di riferimento, causando una contaminazione della clean area dell'immagine corrente. Va esclusa pertanto dalla lista delle CU candidate per la spatial MVP anche la CU *x,* nonostante questa sia contenuta nella clean area.
- Affine merge.

  Se la CU corrente è codificata con la tecnica affine merge è necessario che i vettori mv0, mv1, e mv2, utilizzati per calcolare i MV dei sotto blocchi, siano fuori dalla dirty area.

Inoltre è possibile che uno dei MVs ricavati, relativo ad un sotto blocco, punti ad un sotto blocco di riferimento contenuto parzialmente nella dirty area. Questa caso va pertanto escluso e la tecnica di affine merge utilizzando mv0, mv1 e mv2 non può essere applicata alla CU corrente.

Utilizzando la tecnica GDR è fondamentale evitare l'introduzione di leaks, ma è importante anche scegliere lo strumento di codifica più efficiente per ogni CU. E' necessario quindi utilizzare un metodo opportuno per selezionare la modalità di codifica da utilizzare per ogni CU, detto "joint mode selection" o "joint mode decision method", con lo scopo di massimizzare l'efficienza di codifica oltre che evitare i leaks..

La joint mode selection scarta dalla lista di tutte le tecniche di predizione possibili quelle che introdurrebbero leaks, e dalle tecniche che utilizzano una lista di CU candidate da utilizzare nella predizione vengono scartate tutte le combinazioni di CU che introdurrebbero leaks. In seguito a questa selezione, ogni tecnica rimanente è una tecnica di codifica valida per l'utilizzo di GDR, e per ciascuna di esse viene calcolato un costo c secondo la formula $c = D + \alpha \times R$.

D rappresenta la distorsione introdotta dalla tecnica di codifica, R è il numero di bit utilizzati per rappresentare la CU corrente e α è un moltiplicatore di lagrange. Il costo c stabilisce il "punteggio" di una tecnica di codifica e per ogni CU viene selezionata la tecnica di codifica con costo associato minore.

La tecnica GDR permette quindi di rendere uniforme il bitrate codificando, secondo le tecniche descritte, regioni intra distribuite su molte immagini invece che codificare un'unica immagine interamente intra. Affinché GDR sia utilizzata correttamente è necessario non introdurre leaks durante la codifica, e come detto in precedenza al fine di evitare l'introduzione di leaks è necessario scartare delle possibilità di codifica per ogni CU.

E' possibile che le tecniche di codifica scartate comprendessero le tecniche di codifica ottime (ovvero le tecniche più efficienti) per una data CU, costringendo il codificatore a rinunciare all'utilizzo della tecnica di codifica migliore e a scegliere invece fra le tecniche che è concesso utilizzare. Ogni qualvolta l'utilizzo di GDR costringe a scartare la tecnica di codifica ottima per una qualsiasi CU, si ha una perdita di efficienza rispetto ad eseguire la codifica senza l'utilizzo di GDR. In conclusione l'utilizzo di GDR permette di rendere uniforme il bitrate dello streaming, eliminando picchi di ritardo, al prezzo di una minore efficienza di codifica.

Capitolo 5 - ANALISI SPERIMENTALE DI CODIFICA

5.1 Descrizione delle misurazioni

L'obiettivo di questo capitolo è di mettere a confronto una codifica/decodifica VVC tradizionale e una con l'utilizzo di GDR per analizzare le conseguenze che l'utilizzo di tale tecnica comporta tramite misurazioni sperimentali.

La risorsa video selezionata per le sperimentazioni è nominata "foreman_cif.yuv", sequenza molto spesso usata come esempio in questo ambito. Si tratta di un video rappresentato nello spazio YUV, in formato CIF di dimensione 352 x 288 pixel composto da 300 immagini.

Fig. 8: un'immagine fornita come esempio appartenente alla sequenza video utilizzata, in particolare si tratta del frame 41 di 300.

Tale video verrà codificato utilizzando una tecnica di codifica tradizionale e una codifica utilizzando GDR, per poi analizzare i risultati ottenuti. I dati ottenuti saranno riportati in appositi grafici e una tabella riassuntiva.

Per misurare la differenza di qualità fra ogni immagine originale e ogni immagine codificata verranno utilizzati due indici: il PSNR e l'SSIM (vedi sezione 3.3). Per confrontare matematicamente le due codifiche verranno utilizzate le metriche di Bjontegaard, che permettono tramite funzioni matematiche di confrontare due diverse codifiche in termini di efficienza. Per avere a disposizione un campione di dati sufficientemente ampio il video verrà codificato 4 volte per ognuna delle due modalità facendo variare il parametro di quantizzazione QP. I valori di QP selezionati sono QP = 22, QP = 27, QP = 32, QP = 37, per un totale di 8 procedure di codifica.

L'obiettivo è di verificare sperimentalmente le conseguenze che l'utilizzo di GDR comporta (nella sezione 4.5 è data la spiegazione teorica) e di quantificare la differenza media percentuale di bitrate e qualità delle immagini prodotte.

5.2 Software utilizzato

Per effettuare le varie codifiche della risorsa video campione nelle due modalità interessate verrà utilizzato il software open riferimento dello standard VVC, ovvero VVC TEST MODEL (VTM), il cui repository github è reperibile tramite l'indirizzo URL [15].

VTM è un software scritto nel linguaggio c++ il cui scopo è di fornire all'utente un codec VVC per eseguire delle simulazioni di codifica di qualsiasi natura. Per eseguire una codifica è necessario comunicare al software il valore dei parametri di interesse, che possono essere liberamente impostati per analizzare le diverse configurazioni di codifica.

Il valore dei parametri può essere espresso tramite linea di comando o tramite un file di configurazione con estensione ".cfg" scritto con un'apposita sintassi. Dal momento che i parametri da specificare sono molteplici è preferibile la seconda opzione in modo da tenere traccia delle analisi effettuate e avere a disposizione un file di testo facile da visualizzare e modificare.

VTM permette di simulare un codec VVC in tutta la sua completezza ed è usato in ambito professionale per testare le molteplici funzionalità di codifica che il codec VVC offre. Al fine di eseguire le simulazioni di interesse che mettono a confronto una codifica standard e una codifica GDR verrà utilizzato un file di configurazione predefinito fornito dagli sviluppatori, nominato "encoder_lowdelay_vtm.cfg". Si è scelto di utilizzare questo file di configurazione perché si tratta di una configurazione del codec più adatta se si vuole minimizzare la latenza, in cui non viene fatta predizione dal futuro verso il passato in modo da eliminare la latenza strutturale associata alla predizione relativa al passato (backward).

Tale file di configurazione verrà utilizzato come base e modificato per eseguire le simulazioni di codifica interessate.

Per quanto riguarda invece le misurazioni dell'SSIM e le metriche di Bjontegaard verranno utilizzate delle apposite funzioni Matlab, in quanto VTM restituisce in output solamente il valore di PSNR medio, bitrate medio e dimensione di ogni frame.

La funzione per il calcolo dell'SSIM riceve in input due immagini e restituisce il valore dell'SSIM fra esse. Dal momento che l'immagine del video originale è l'immagine di riferimento, tale valore

indica la qualità dell'immagine codificata. Il valore restituito dalla funzione è compreso fra 0 e 1 e il valore 1 indica che le due immagini coincidono. La funzione Matlab fornita verrà utilizzata con i valori dei parametri di default, ovvero le costanti assumono i seguenti valori:

K1 = 0.1, K2 = 0.3, L = 255.

Per calcolare l'SSIM per ogni immagine di un video è necessario estrarre i singoli frame del video originale e quello codificato e passarli come parametro alla funzione di calcolo, per farlo verranno utilizzate apposite funzioni Matlab che permettono di estrarre il k-esimo frame di un video, e verranno messi a confronto frame in posizioni corrispondenti.

La funzione per il calcolo delle metriche di Bjontegaard riceve in input due matrici, denominate matrice 1 e matrice 2. Le due matrici sono costituite da quattro coppie di valori, ovvero coppie di bitrate medio e indice di qualità medio. Ogni coppia di valori rappresenta quindi un risultato medio che riassume una simulazione di codifica, e come indice di qualità è possibile utilizzare a piacimento sia il PSNR che l'SSIM. Nelle sperimentazioni verranno utilizzati entrambi gli indici, producendo un risultato delle metriche di Bjontegaard relativo al PSNR e un altro relativo all'SSIM.

Se per esempio si decide di utilizzare come indice il PSNR, la prima riga della matrice 1 è la coppia di valori bitrate medio e PSNR medio della codifica senza GDR ponendo QP = 37, la seconda riga sarà analoga ma con valore di QP = 32, la terza con QP = 27 e infine la quarta riga con QP = 22. La matrice 2 invece conterrà i valori analoghi della codifica con GDR, e l'output prodotto dalla funzione di calcolo produrrà una coppia di valori: la differenza media di PSNR e la differenza media di bitrate fra le due codifiche. Analogamente verrà ripetuta questa procedura utilizzando come indice di qualità l'SSIM, che produrrà in output la differenza media percentuale di bitrate e la differenza media di SSIM fra le due codifiche.

5.3 risultati delle simulazioni

Le figure seguenti rappresentano degli istogrammi raffiguranti l'andamento del bitrate delle varie codifiche eseguite sullo stesso video campione. Come detto in precedenza, le codifiche sono state effettuate su quattro valori diversi di quantizzazione QP, per un totale di quattro codifiche usando GDR e quattro codifiche non usando GDR.

Si è scelto di configurare il codec su un totale di 64 immagini per video, considerando i tempi di elaborazione necessari per effettuare tale operazione. Per quanto riguarda la codifica senza l'utilizzo di GDR si è deciso di codificare un'immagine intra ogni 16 immagini, quindi con un GOP di 16

immagini. Per la codifica usando GDR si è scelto che tutte le immagini sono codificate come immagini GDR e non inserire quindi immagini tradizionali fra un'immagine GDR e l'altra.

Nei grafici riportati, fig. da 9 a 16, l'asse orizzontale rappresenta la numerazione delle immagini, quindi la prima immagine è POC(1), la seconda POC(2), e così via, mentre l'asse verticale rappresenta il numero di bit utilizzati per rappresentare le immagini. Le asticelle rappresentano quindi le immagini sequenzialmente, e la relativa altezza ne indica la dimensione.

Fig. 9: Bitrate di codifica non GDR con QP = 37

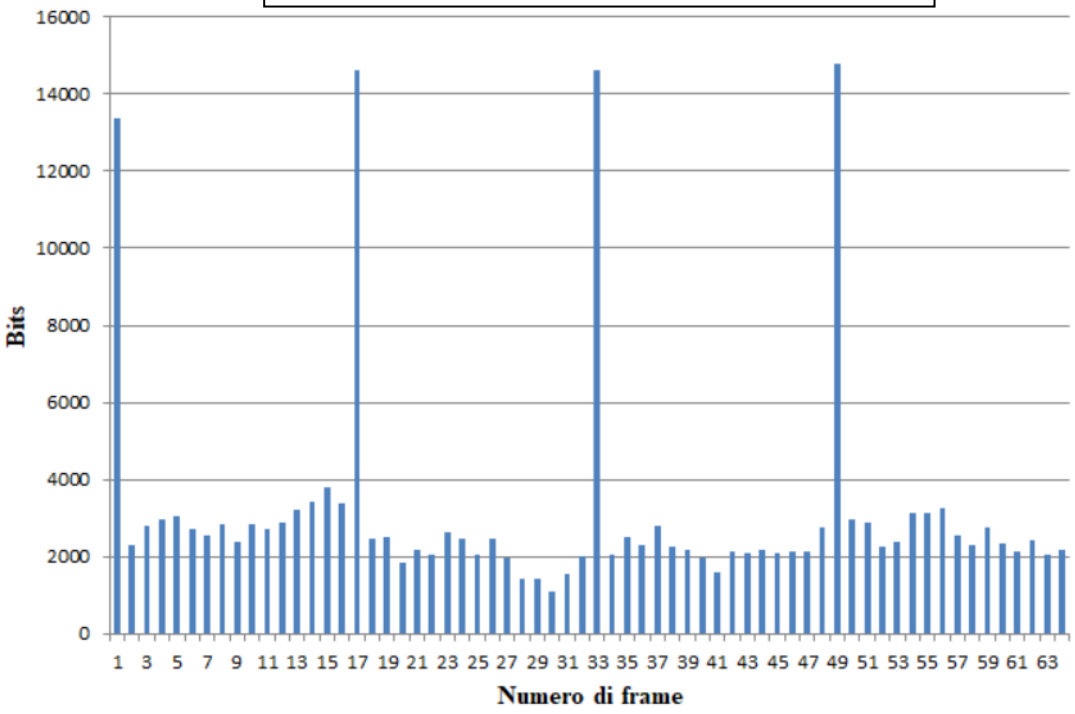


Fig. 10: Bitrate di codifica GDR con QP = 37

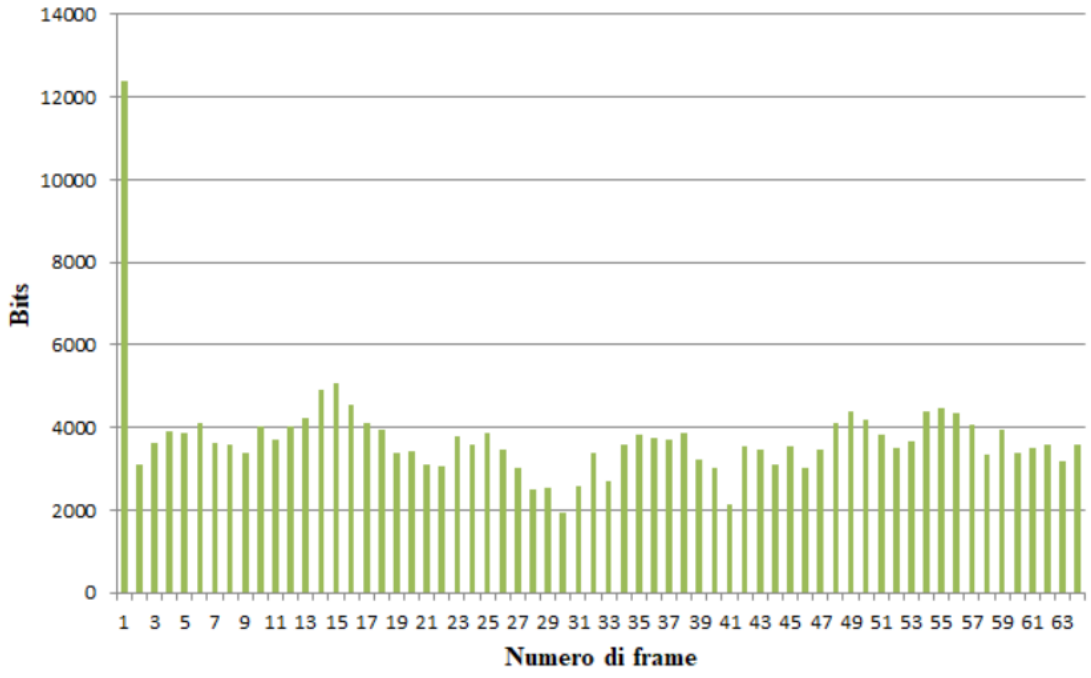


Fig. 11: Bitrate di codifica non GDR con QP = 32

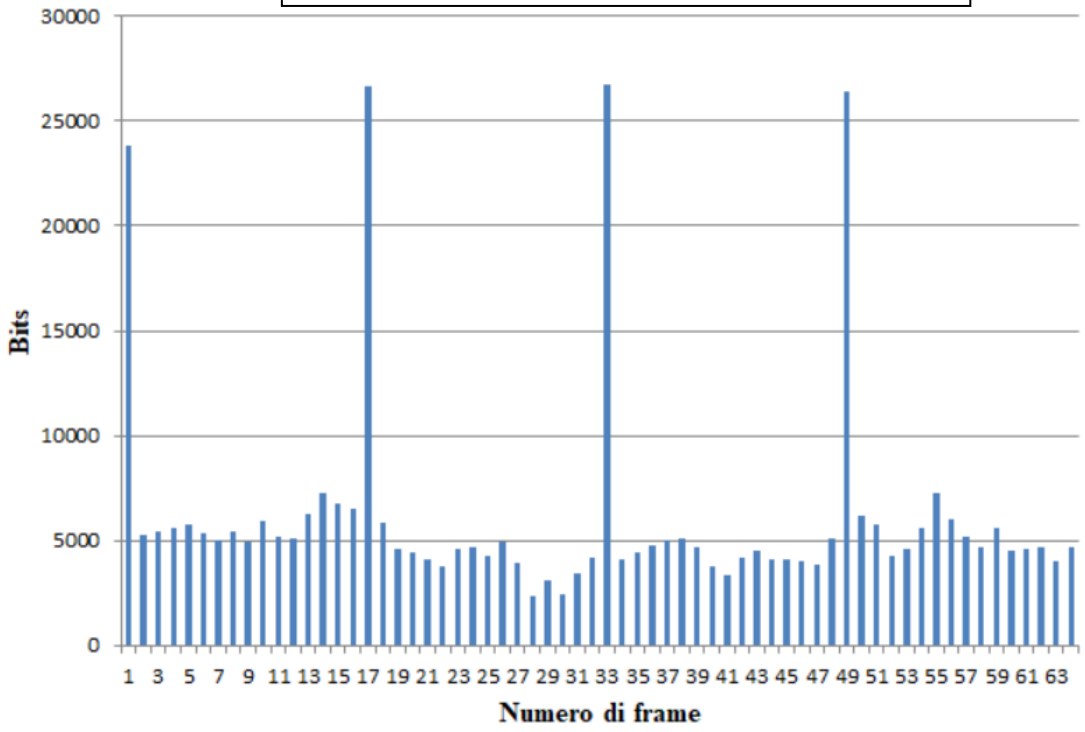


Fig.12: Bitrate di codifica GDR con QP = 32

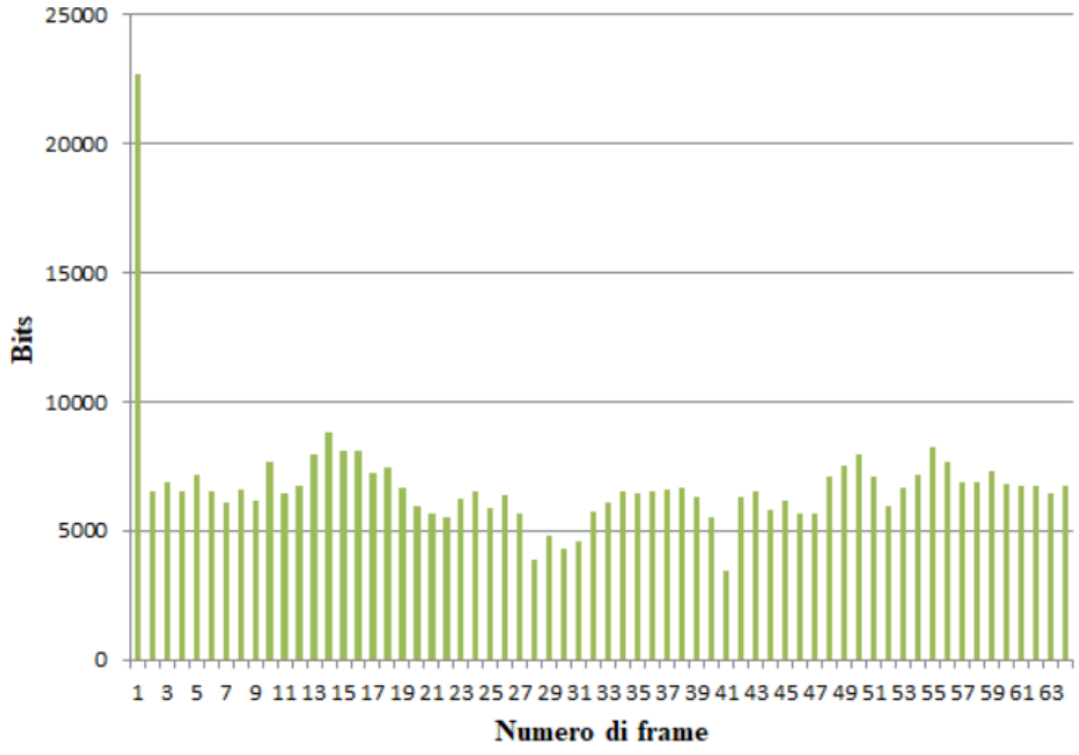


Fig. 13: Bitrate di codifica non GDR con QP = 27

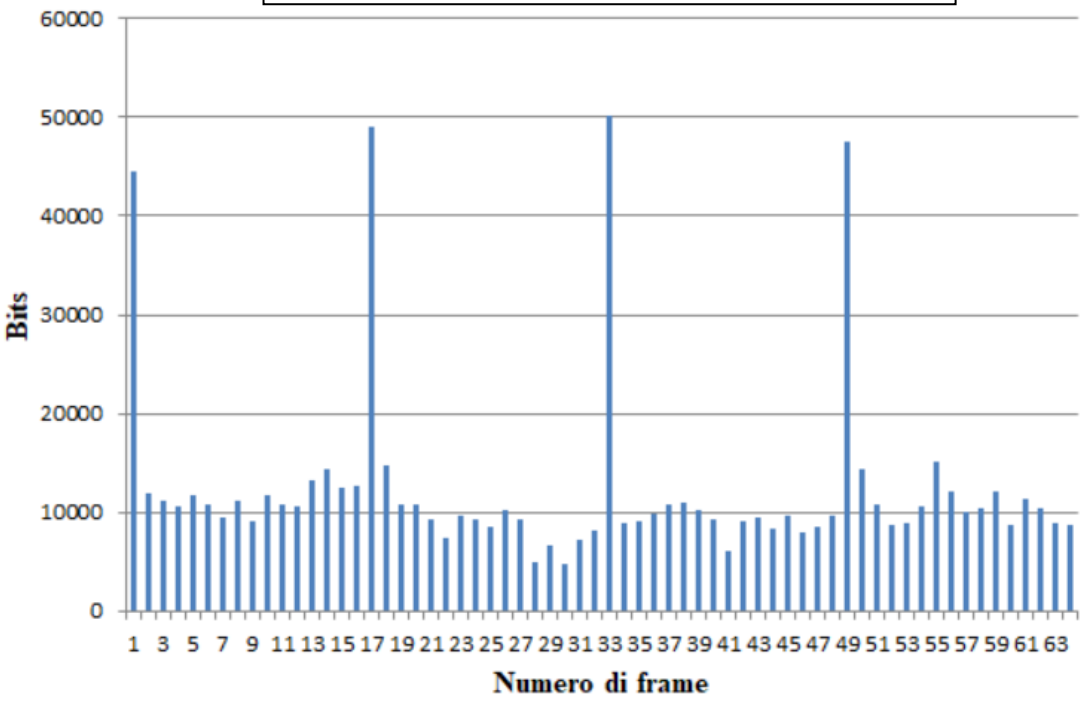


Fig. 14: Bitrate di codifica GDR con QP = 27

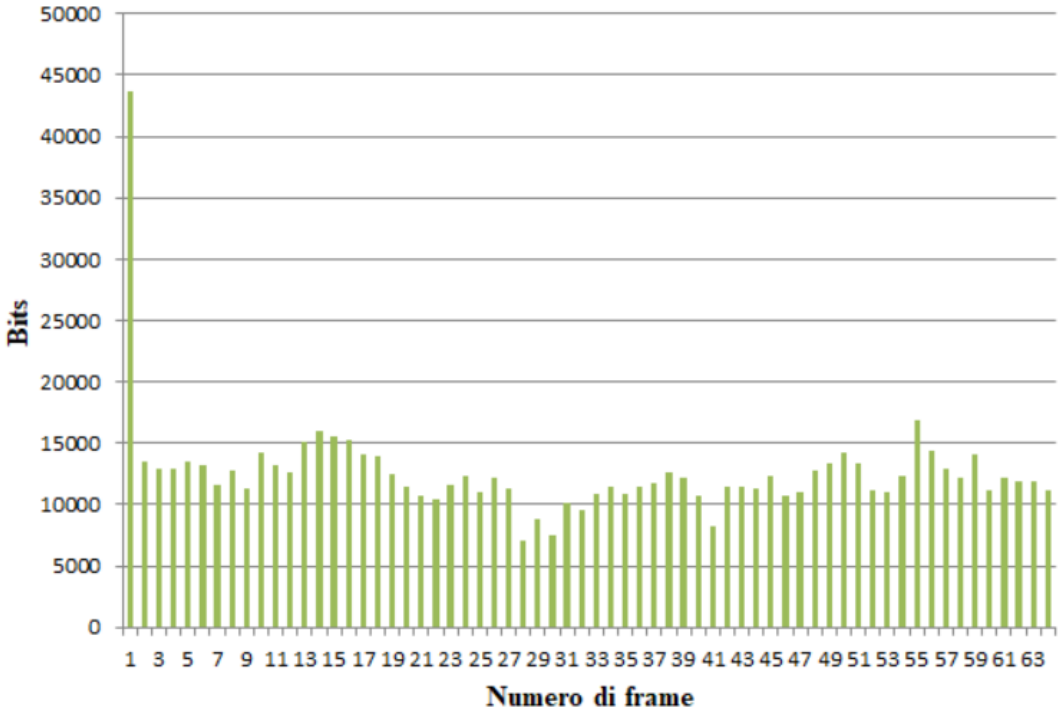


Fig. 15: Bitrate di codifica non GDR con QP = 22

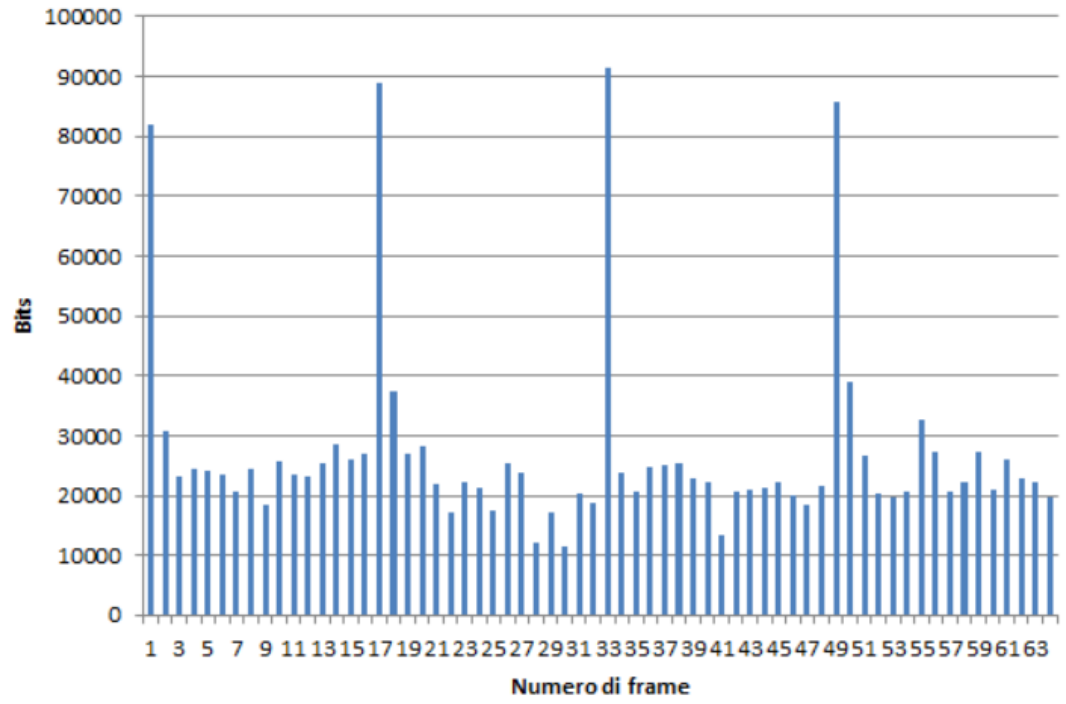
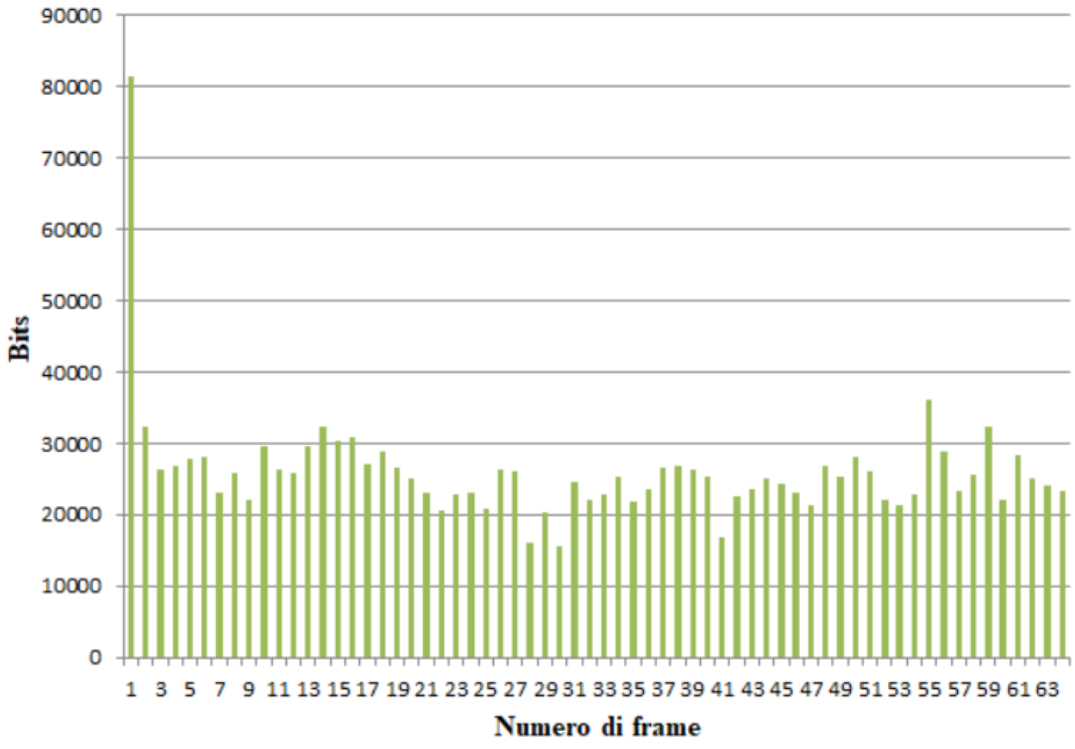


Fig. 16: Bitrate di codifica GDR con QP = 22



5.4 Interpretazione dei risultati

Osservando i grafici riportati si può notare che codificando un video senza utilizzare GDR si ripetono periodicamente delle immagini che necessitano di molti più bit delle altre per essere rappresentate, ovvero le immagini intra. Nelle codifiche GDR invece tali picchi non sono presenti, ma confrontando le due codifiche a parità di QP si nota che mediamente le immagini necessitano di più bit per essere rappresentate rispetto ad una codifica tradizionale. Nelle codifiche GDR la prima immagine è codificata in maniera intra a causa del file di configurazione utilizzato, che comunque non si ripete periodicamente.

Il software utilizzato fornisce di default i valori di bitrate medio e PSNR medio per ogni codifica, volendo inoltre calcolare l'indice di qualità SSIM medio è necessario utilizzare del software specifico a tale scopo, in questo caso sono state usate delle funzioni Matlab.

La seguente tabella riporta i risultati ottenuti dalle simulazioni di codifica, riportando bitrate medio [Kbit/s], PSNR medio e SSIM medio per ogni codifica effettuata.

QP	GDR si/no	Bitrate [Kbit/s]	PSNR	SSIM
37	no	50.7929	34.8945	0.89130
37	si	57.7960	34.9322	0.89136
32	no	98.0280	37.5603	0.92468
32	si	107.9300	37.5520	0.92462
27	no	198.4120	40.3098	0.95305
27	si	202.0380	40.2818	0.95284
22	no	452.5820	43.3756	0.97387
22	si	416.3760	43.3424	0.97370

Tali dati sono stati utilizzati per il calcolo delle metriche di Bjontegaard, effettuate tramite l'utilizzo di una funzione Matlab e di uno script riportato in seguito.

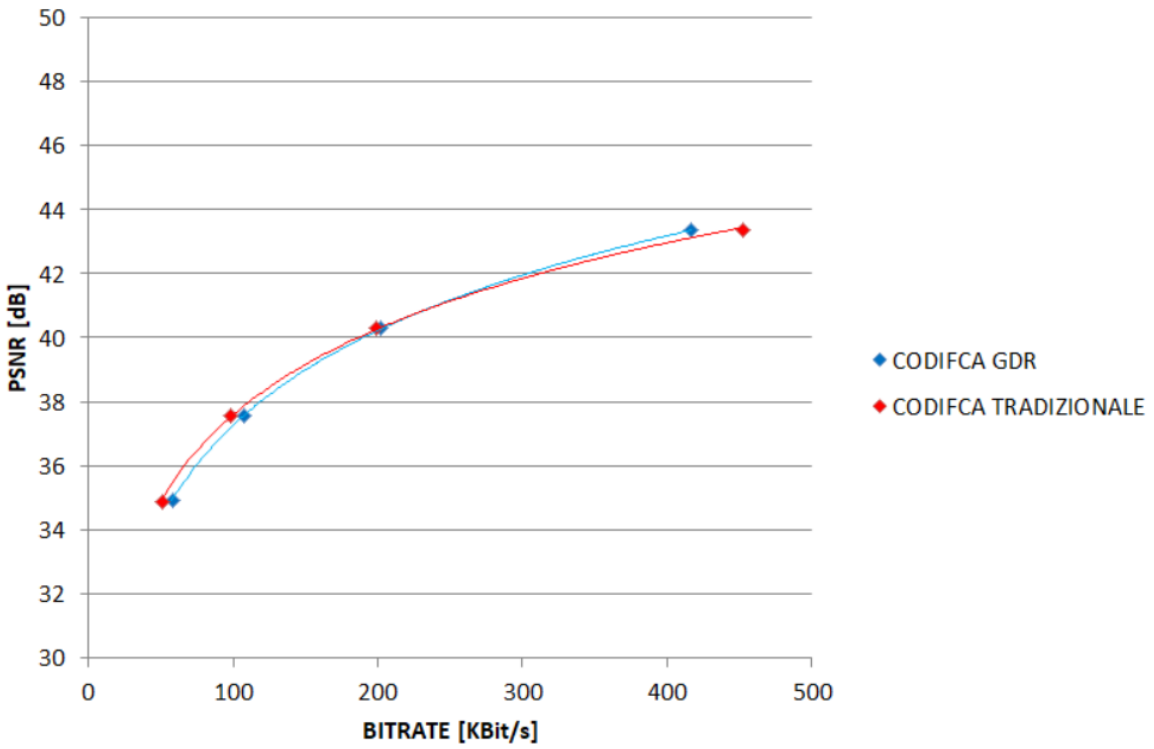
```
 data1 = [  %  CODIFICA NORMALE
%BITRATE -  PSNR,
 50.7929    34.8945     % QP = 37
 98.0280    37.5603     % QP = 32
 198.4120   40.3098     % QP = 27
 452.5820   43.3756     % QP = 22
 ];

 data2 = [  % CODIFICA GDR
%BITRATE    PSNR
 57.7960    34.9322     % QP = 37
 107.9300   37.5520     % QP = 32
 202.0380   40.2818%    % QP = 27
 416.3760   43.3424%    % QP = 22
 ];

 [deltapsnr , percrate] = bjm_e(data1, data2);
 disp(['delta PSNR : ',num2str(deltapsnr)]);
 disp(['risparmio % bitrate : ',num2str(percrate)]);
```

Le metriche di Bjontegaard permettono di ricavare la differenza di PSNR e il risparmio percentuale di bitrate fra la curva 1 e la curva 2. Tali metriche necessitano di alcuni punti nello spazio bitrate/indice di qualità ottenuti tramite delle misurazioni, in questo caso si tratta di 4 punti per entrambe le codifiche, ottenuti facendo variare il valore di QP nel file di configurazione del software.

Fig. 17: la seguente figura rappresenta i 4 punti ottenuti per ciascuna codifica nel piano bitrate-PSNR del video “foreman_cif.yuv”. Grazie a questo grafico è possibile ricavare le curve delle due codifiche tramite un’interpolazione dei dati, operazione effettuata dalle metriche di Bjontegaard.



Sono state calcolate le metriche di Bjontegaard sia con PSNR che con SSIM, ottenendo per il video "foreman_cif.yuv" i seguenti risultati.

Metriche di Bjontegaard bitrate-PSNR:

- La codifica non GDR ha un PSNR mediamente maggiore di 0.19 dB.
- La codifica non GDR ha mediamente un risparmio di bitrate del 4.88%.

Metriche di Bjontegaard bitrate-SSIM:

- La codifica non GDR ha un SSIM mediamente maggiore di 0.00249.
- La codifica non GDR ha mediamente un risparmio di bitrate del 6.61%.

Si osserva che il valore di risparmio percentuale del bitrate presenta due valori diversi usando PSNR e SSIM dal momento che le curve sono definite nel piano bitrate / indice di qualità, tale valore è conforme quindi alle metriche della qualità utilizzate.

Si può quindi concludere affermando che l'utilizzo di GDR diminuisce notevolmente i valori massimi del bitrate rispetto ad una codifica tradizionale, eliminando quindi picchi nel ritardo causati dalla trasmissione delle immagini intra. Grazie all'utilizzo delle metriche di Bjontegaard è stato possibile inoltre quantificare la perdita di efficienza che GDR comporta rispetto ad una codifica tradizionale, perdita che è dal punto di vista teorico giustificata dal momento che per evitare i GDR leaks è necessario escludere determinate tecniche e possibilità di codifica. Tale perdita di efficienza dipende dalle caratteristiche del video utilizzato come campione, e nelle simulazioni effettuate ammonta al 4.8814% utilizzando come indice di valutazione il PSNR, e al 6.6115% utilizzando invece l'SSIM. Questi valori rispecchiano le sperimentazioni effettuate nell'articolo [11], dove si afferma che la perdita che si ottiene usando GDR varia generalmente dal 6 al 10 percento.

5.5 Utilizzo di un altro video riferimento

Dal momento che la natura dei risultati ottenuti dipende in buona parte dalla natura del video utilizzato come riferimento, si è deciso di ripetere le stesse procedure effettuate in precedenza su un altro video di natura diversa.

A differenza del video utilizzato in precedenza, che raffigura scene piuttosto statiche con una persona inquadrata in primo piano che parla alla telecamera, il seguente video ("BUS_cif.yuv") rappresenta una scena più complessa che riprende un veicolo in movimento, con l'inquadratura che di conseguenza è stavolta dinamica. Il video è come il precedente rappresentato nello spazio YUV, in formato CIF 352 x 288 pixel con un totale di 150 frame.

Fig. 18: un'immagine fornita come esempio del video utilizzato, in particolare il frame 44 di 150. Si può notare che la scena inquadrata è notevolmente più complessa rispetto al video "foreman_cif.yuv", di conseguenza è ragionevole aspettarsi che le immagini necessitino di più bit per essere rappresentate, a parità di qualità.

Dal momento che le procedure sono analoghe allo studio fatto in precedenza, verranno sinteticamente espressi i risultati ottenuti. In seguito sono riportati i grafici raffiguranti i bitrate di codifica, fig. da 19 a 26, come in precedenza sono state effettuate 4 codifiche usando GDR e 4 codifiche tradizionali, con gli stessi valori di quantizzazione pari a 37,32,27 e 22. A differenza delle codifiche effettuate sul video "foreman_cif.yuv", in questo video si è scelto, per quanto riguarda la codifica senza GDR, di codificare più realisticamente un'immagine intra ogni 32 immagini, in quanto ogni 16 immagini è una scelta per visualizzare meglio tale aspetto nei grafici.

Fig. 19: Bitrate di codifica non GDR con QP = 37

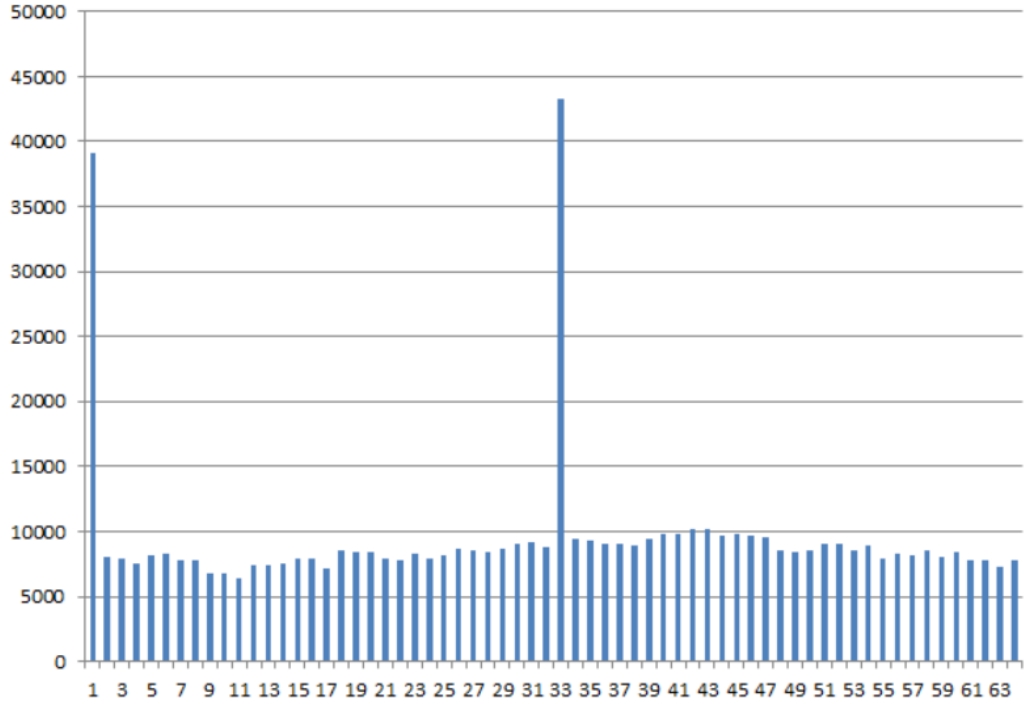


Fig. 20: Bitrate di codifica GDR con QP = 37

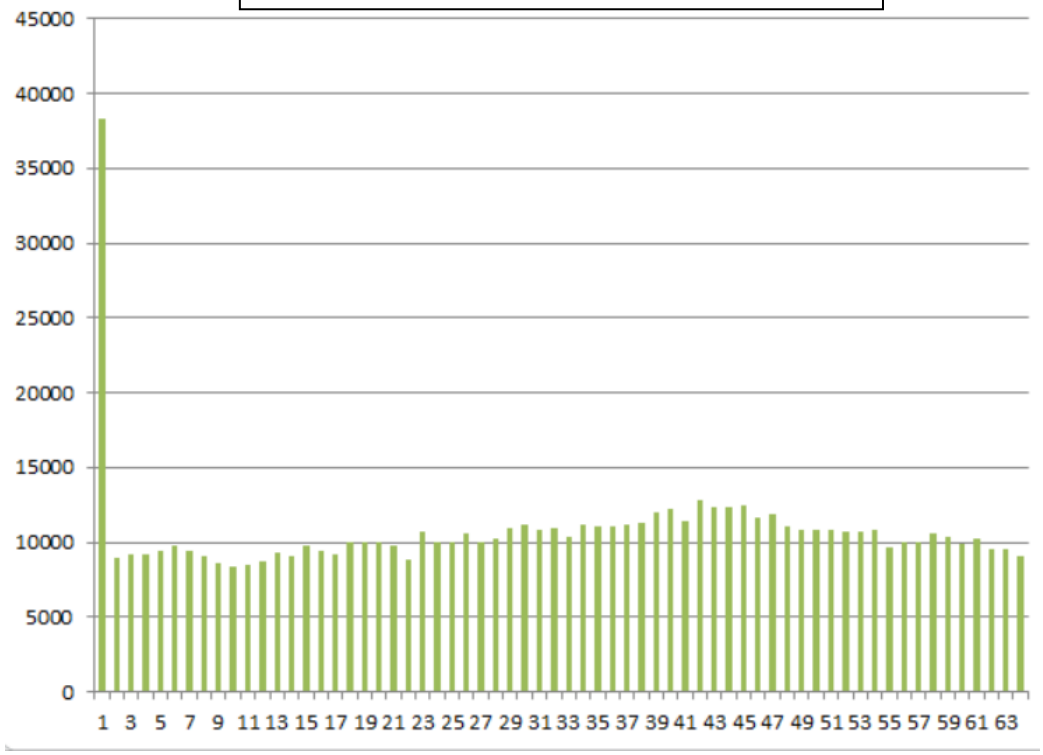


Fig. 21: Bitrate di codifica non GDR con QP = 32

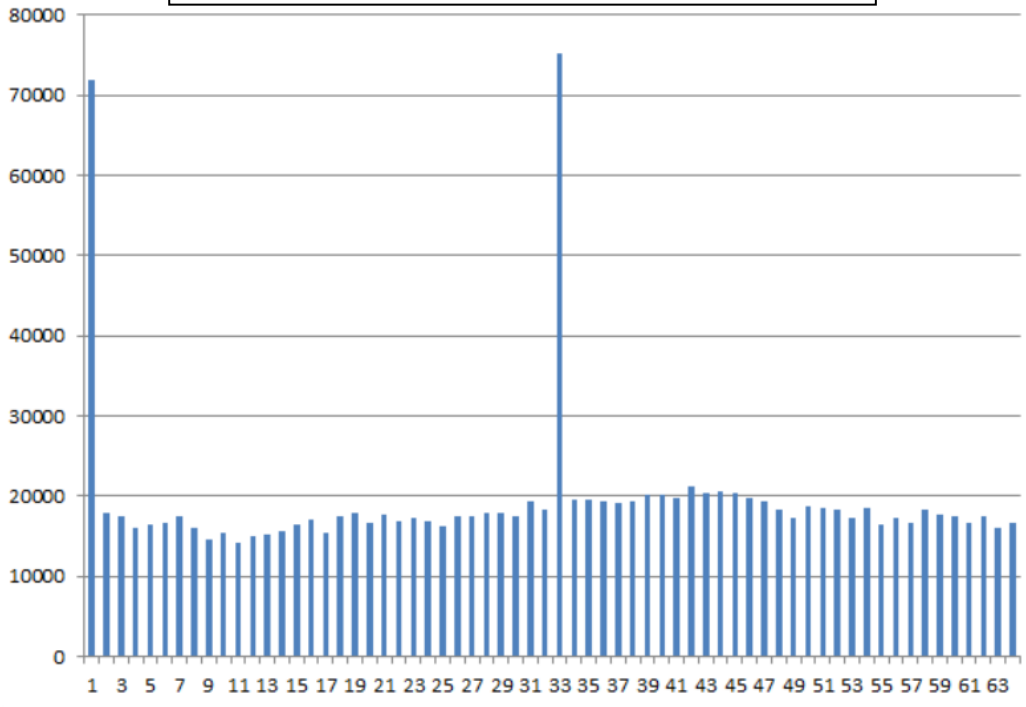


Fig. 22: Bitrate di codifica GDR con QP = 32

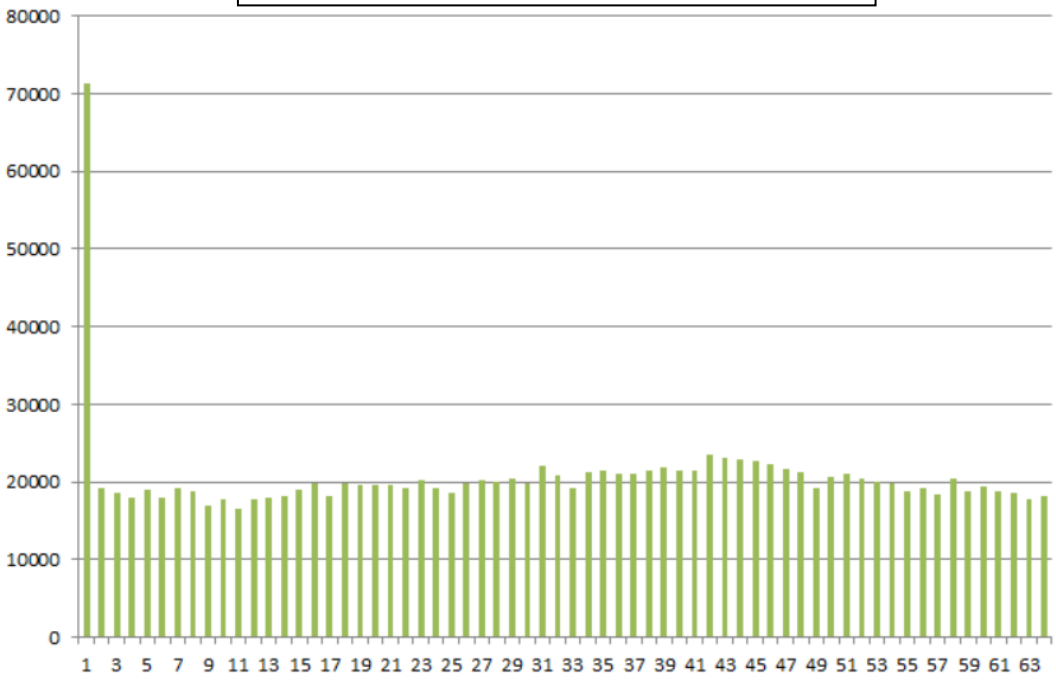


Fig. 23: Bitrate di codifica non GDR con QP = 27

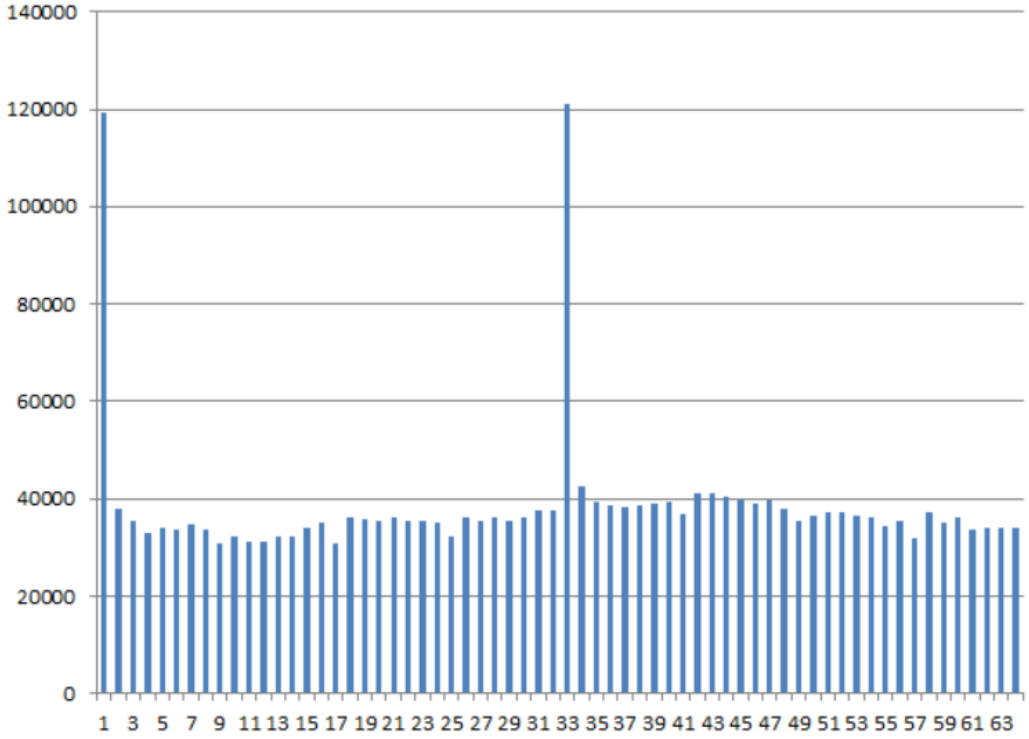


Fig. 24: Bitrate di codifica GDR con QP = 27

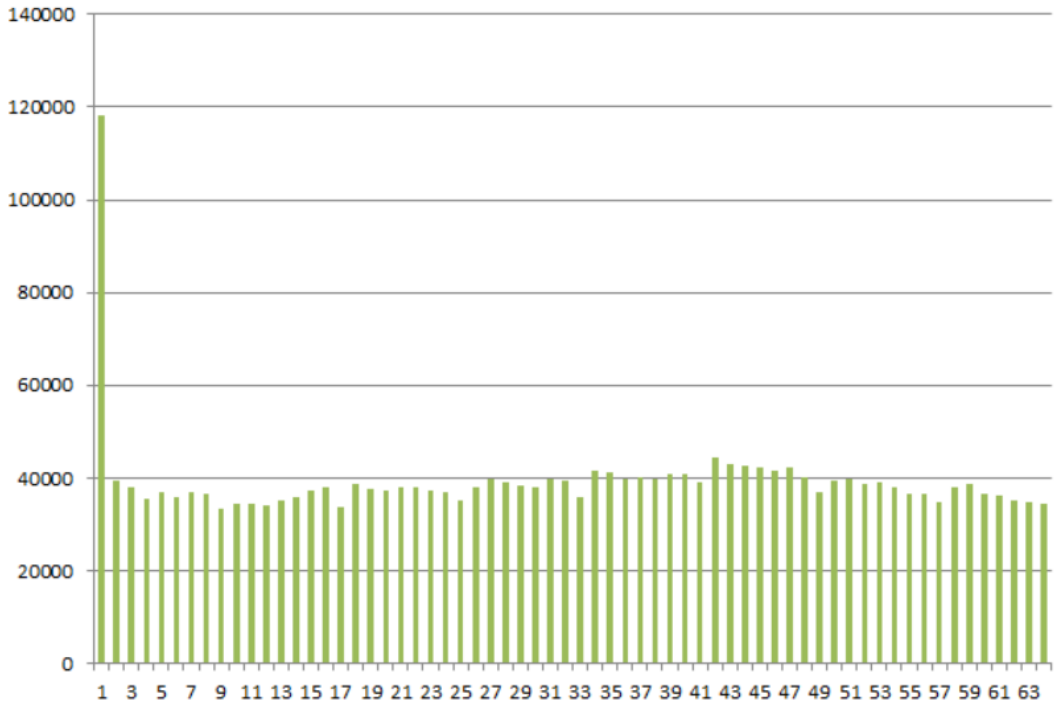


Fig. 25: Bitrate di codifica non GDR con QP = 22

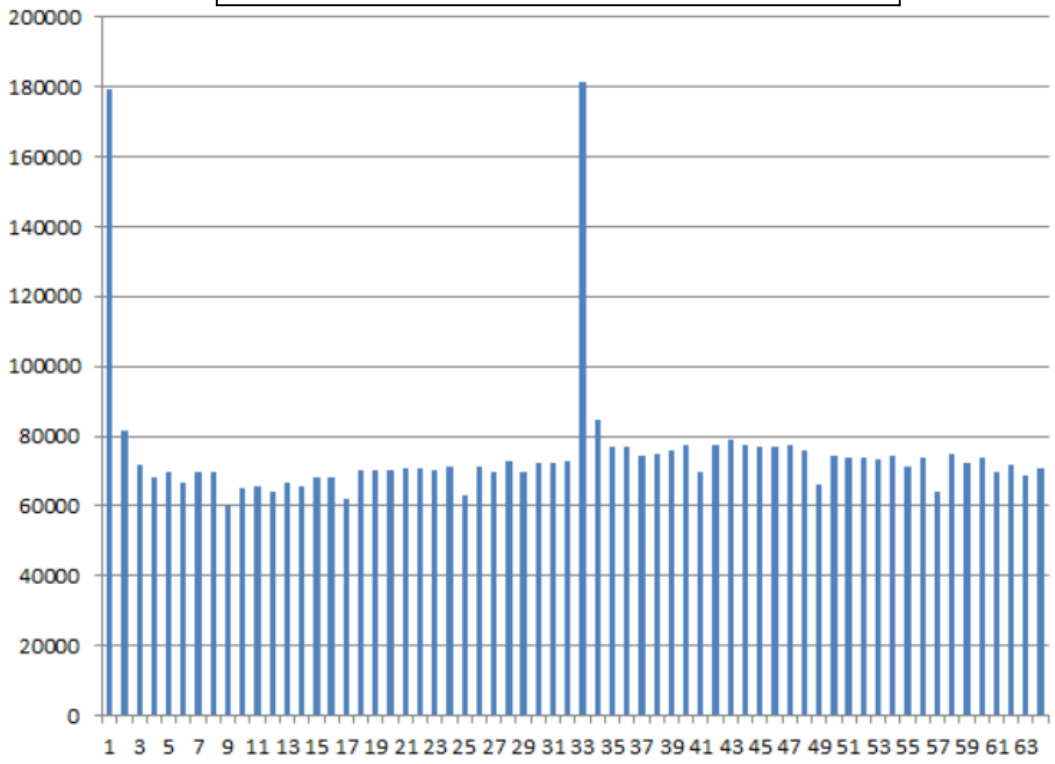
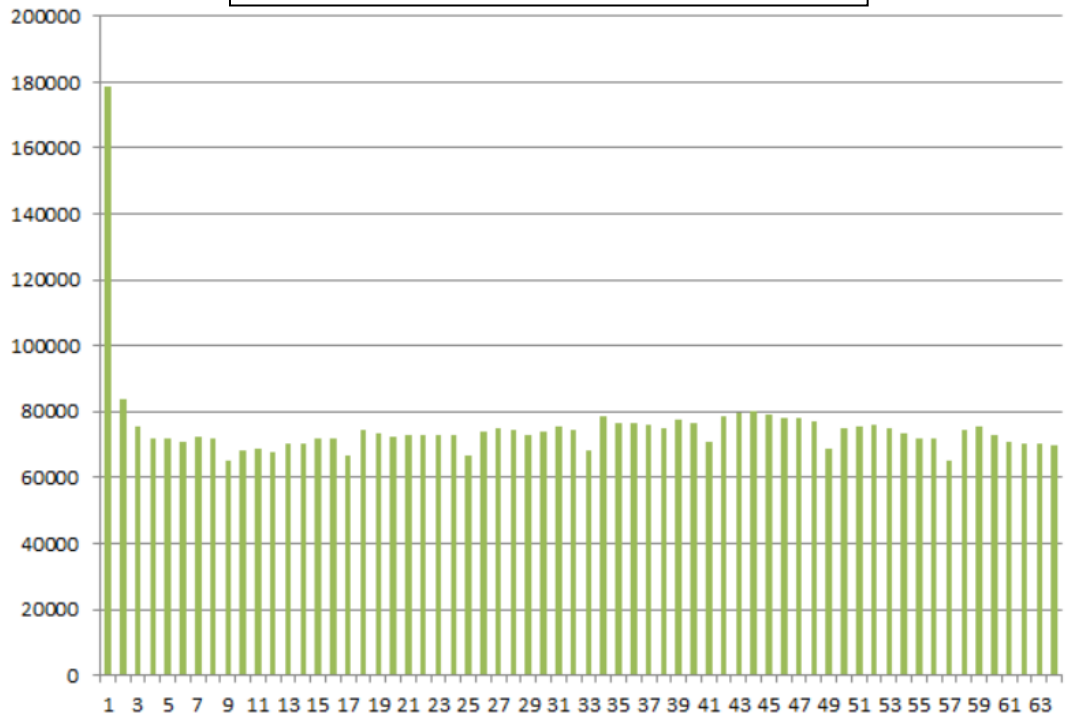


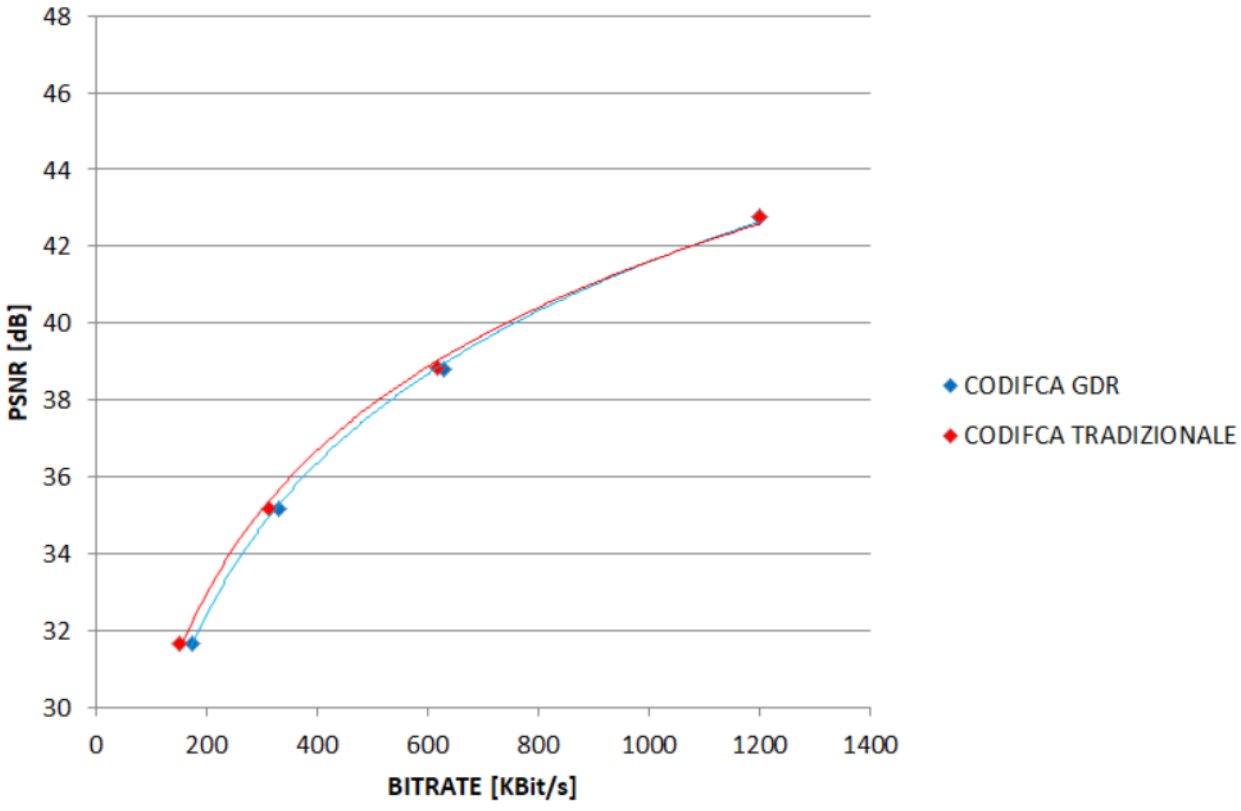
Fig. 26: Bitrate di codifica GDR con QP = 22



La tabella riassuntiva dei dati ottenuti è la seguente:

QP	GDR si/no	Bitrate [Kbit/s]	PSNR	SSIM
37	no	151.1980	31.6647	0.86830
37	si	172.2380	31.6833	0.86989
32	no	311.1180	35.1584	0.93456
32	si	330.4100	35.1649	0.93604
27	no	615.7420	38.8319	0.96883
27	si	628.3540	38.8245	0.96940
22	no	1200.5380	42.7756	0.98611
22	si	1199.7140	42.7539	0.98616

Fig. 27: analogamente a quanto svolto per il video “foreman_cif.yuv”, il seguente grafico riporta i 4 punti nel piano bitrate – PSNR delle due codifiche. Si può notare che la codifica GDR ha una qualità molto simile alla codifica tradizionale, ma un bitrate maggiore, ad esclusione del valore di QP = 22 in cui i due punti sono approssimativamente sovrapposti.



Utilizzando questi dati come input per le metriche di Bjontegaard si ottengono i seguenti risultati per il video “BUS_cif.yuv”.

Metriche di Bjontegaard bitrate-PSNR:

- La codifica non GDR ha un PSNR mediamente maggiore di 0.23 dB.
- La codifica non GDR ha mediamente un risparmio di bitrate del 4.56%.

Metriche di Bjontegaard bitrate-SSIM:

- La codifica non GDR ha un SSIM mediamente maggiore di 0.00264.
- La codifica non GDR ha mediamente un risparmio di bitrate del 5.37%.

Si può quindi dire che i risultati ottenuti nello specifico dipendono evidentemente dal video utilizzato come riferimento, tuttavia rimangono in linea con i risultati teorici espressi nell’articolo [11], secondo cui l’utilizzo di GDR comporta ad una perdita di efficienza generalmente minore del 10%.

CONCLUSIONI

Il fenomeno dello streaming video è in continua crescita ed evoluzione. Le tecniche e gli strumenti utilizzati hanno visto significativi cambiamenti nel corso degli anni, e in particolar modo il codec VVC, rilasciato nel 2020, ha migliorato di oltre il 40% l'efficienza del precedente codec HEVC. Oltre alle innovative ed efficienti tecniche di codifica introdotte il VVC offre la possibilità di codificare e decodificare nuovi formati di video in continua crescita emersi negli ultimi anni, come video a più riprese, video immersivi per emulare una realtà virtuale, video a 360 gradi e altri.

In particolar modo negli ultimi anni si è diffuso notevolmente l'utilizzo di applicazioni di streaming a bassa latenza, che per fornire un servizio valido necessitano di operare con valori di latenza sufficientemente bassi, permettendo quindi che lo scambio di contenuto video fra client e server sia percepito come fosse in tempo reale. La tecnica GDR è una recente tecnica di codifica supportata dal codec VVC che permette di rendere uniforme il bitrate di uno streaming video evitando di codificare periodicamente intere immagini in modalità intra, ma distribuendo aree codificate intra in più immagini, con una conseguente riduzione della latenza. Per utilizzare correttamente GDR il codificatore è però costretto a rinunciare molto spesso alla tecnica di codifica ottima per una data coding unit, con una conseguente perdita di efficienza complessiva nella codifica. Tale perdita di efficienza è tuttavia accettabile in quanto GDR è utilizzato per applicazioni a (ultra) bassa latenza e di conseguenza la minimizzazione di essa è una priorità.

Infine è stata eseguita un'analisi sperimentale utilizzando VTM (VVC Test Model), ovvero il software riferimento del codec VVC. Lo scopo dell'analisi è quello di mettere a confronto l'andamento del bitrate di un dato video codificato con GDR e senza GDR, verificando che l'utilizzo di GDR permette di rimuovere i picchi causati dalle immagini intra rendendo il bitrate visibilmente più uniforme. Inoltre sono state analizzate le coppie di valori bitrate [bit/s] e indice di qualità (sono state fatte misurazioni sia con PSNR che con SSIM), dimostrando e quantificando la perdita di efficienza che l'utilizzo di GDR comporta.

BIBLIOGRAFIA

[3] High Efficiency Video Coding, Standard ISO/IEC 23008-2,ISO/IEC JTC 1, Apr. 2013.

[4] Versatile Video Coding, Standard ISO/IEC 23090-3, ISO/IEC JTC 1, Jul. 2020.

[5] Kevin Spiteri, Ramesh Sitaraman, Daniel Sparacio, “From Theory to Practice: Improving Bitrate Adaptation in the DASH Reference Player”, MMSys’18, June 12–15, 2018, Amsterdam, Netherlands.

[7] Anthony Vetro, Iraj Sodagar, “The MPEG-DASH Standard for Multimedia Streaming Over the Internet”, October-December 2011.

[6] Jonathan Kua, Grenville Armitage, Philip Branch, “A Survey of Rate Adaptation Techniques for Dynamic Adaptive Streaming Over HTTP”, IEEE communications surveys & tutorials, VOL. 19, NO. 3, third quarter 2017.

[9] Michael Seufert, Sebastian Egger, Martin Slanina, Thomas Zinner, Tobias Hoßfeld, Phuoc Tran-Gia, “A Survey on Quality of Experience of HTTP Adaptive Streaming”, IEEE communications surveys & tutorials, VOL. 17, NO. 1, first quarter 2015.

[11] Limin Wang, Seungwook Hong, Krit Panusopone, “gradual decoding refresh for versatile video coding”, IEEE International Conference on Image Processing (ICIP), 2021

[10] Benjamin Bross , Ye-Kui Wang , Yan Ye , Shan Liu , Jianle Chen, Gary J. Sullivan, Jens-Rainer Ohm, “Overview of the Versatile Video Coding (VVC) Standard and Its Applications”, IEEE transactions on circuits and systems for video technology, VOL. 31, NO. 10, october 2021

SITOGRAFIA

[1] https://blog.gitnux.com

[2] https://www.broadbandsearch.net/blog/internet-statistics#post-navigation-3

[12] https://www.01smartlife.it/codec-vvc-cose-e-come-funziona-lerede-dellhevc/

[8] https://it.wikipedia.org/wiki/Peak_signal-to-noise_ratio

[14] https://steemit.com/webrtc/@airensoft/what-is-ultra-low-latency-video-streaming

[13] https://medium.com/srm-mic/all-about-structural-similarity-index-ssim-theory-code-in-pytorch-6551b455541e

[15] https://jvet.hhi.fraunhofer.de/

[16] https://streaminglearningcenter.com/blogs/open-and-closed-gops-all-you-need-to-know.html