



Università degli Studi di Padova

Facoltà di Ingegneria

Corso di Laurea Magistrale in Ingegneria delle Telecomunicazioni

TESI DI LAUREA

**Ottimizzazione della
trasmissione video su reti P2P
tramite Teoria dei Giochi e
Descrizioni Multiple**

Relatore: Prof. Calvagno Giancarlo

Correlatore: Dott. Milani Simone

Laureando: Gaggio Marco

14 Marzo 2011

Indice

Sommario	III
1 Introduzione	1
1.1 Media Server Farm	2
1.2 Content Delivery Network (CDN)	3
1.3 Tipologia P2P live streaming	4
1.3.1 Tree network	4
1.3.2 Mesh network	5
1.4 Codifica a Descrizioni Multiple (MDC)	6
1.4.1 Trasmissione video su reti P2P tramite MDC	8
2 Codificatore JSVM	9
2.1 Introduzione	9
2.2 Scalabilità e applicazioni	10
2.3 Storia SVC	13
2.4 Aspetti fondamentali H.264/AVC	14
2.4.1 Network Abstraction Layer (NAL)	14
2.4.2 Video Coding Layer (VCL)	14
2.5 Concetti base dell'estensione H.264/SVC	17
2.5.1 Scalabilità temporale	17
2.5.2 Scalabilità spaziale	20
2.5.3 Scalabilità di qualità	23
2.5.4 Controllo del drift nella scalabilità di qualità	24
2.5.5 Estrazione del bit stream	27
2.6 Scalabilità combinata	27
2.7 Conclusioni	28

3	Codifica a Descrizioni Multiple	31
3.1	Introduzione	31
3.1.1	Modello base di Descrizioni Multiple (channel splitting)	32
3.1.2	Channel splitting per sorgenti incorrelate	33
3.1.3	MD viste nella Teoria dell'Informazione	33
3.2	Applicazioni della codifica MD	35
3.3	Come generare MDs	36
3.3.1	Multiple Description mediante IFPD	37
3.3.2	Codifica Scalabile e Unequal Error Protection	37
3.3.3	Multiple Description Coding usando codici FEC	39
 4	 Teoria dei Giochi	 45
4.1	Introduzione	45
4.2	Rappresentazione di un gioco	47
4.2.1	Forma estesa	47
4.2.2	Forma strategica	48
4.2.3	Forma caratteristica	48
4.3	Razionalità e utilità	49
4.3.1	Lotterie	50
4.4	Concetto di “soluzione” del gioco	52
4.4.1	Equilibrio di Nash	52
4.5	Giochi a somma zero	53
4.5.1	Gioco a due giocatori in forma normale	53
4.5.2	Gioco a due giocatori a somma zero senza equilibri puri di Nash	54
4.6	Strategie miste	56
4.6.1	Calcolo equilibrio di Nash in strategie miste	57
 5	 Approccio adottato nella realizzazione	 59
5.1	Introduzione	59
5.2	Da bitstream H.264/SVC a MDC	60
5.3	Classificazione basata sulla Teoria dei Giochi	62
5.3.1	Funzione utilità adottata	65
5.3.2	Tipologia di “Gioco” adottato e equilibrio di Nash	68
5.4	Valutazione della qualità video	69
5.4.1	Peak Signal-to-Noise Ratio (PSNR)	69
5.4.2	Video Quality Metric (VQM)	70

INDICE

6	Risultati sperimentali	75
6.1	Introduzione	75
6.1.1	Tipo di canale utilizzato (modello di Gilbert)	76
6.2	Risultati codifica a descrizioni multiple	78
6.3	Risultati della classificazione basata sulla Teoria dei Giochi	81
6.3.1	Congestione sulla core network	82
6.3.2	Congestione su link non condiviso	88
6.3.3	Congestione sia su link non condiviso che sulla core network	93
7	Conclusioni e sviluppi futuri	99
	Ringraziamenti	101
	Bibliografia	101

Sommario

Lo scopo di questo lavoro di tesi è l'applicazione della codifica a descrizioni multiple ad un flusso video generato da un codificatore scalabile e la sua trasmissione attraverso una rete Peer-to-Peer. Solitamente, in tali reti, i singoli utenti trasmettono il flusso codificato "indipendentemente" dagli altri utenti, creando delle possibili congestioni con il conseguente riempimento delle code di attesa poste nei nodi di instradamento. Nel tentativo di migliorare questo problema si è scelto di modellare il problema tramite la Teoria di Giochi e differenziare le priorità dei pacchetti identificando gli equilibri di Nash. I risultati ottenuti sono stati confrontati con quelli già presenti in letteratura. Lo studio è stato eseguito (per la parte di codifica e decodifica video) sul codificatore scalabile H.264/SVC, mentre per la realizzazione della rete Peer-to-Peer è stato utilizzato il simulatore NS2.

Nel capitolo 1, verrà presentato lo stato dell'arte delle diverse tipologie di reti utilizzate sia per trasmissioni di contenuti statici (file sharing) che real time (streaming video). Su quest'ultimo servizio verranno fatte alcune considerazioni su quali siano le scelte di rete ottimali e quali miglioramenti si possono ottenere utilizzando opportune tecniche di codifica (codifica a descrizioni multiple).

Nel capitolo 2, viene descritto il codificatore H.264/SVC che è di fatto l'estensione scalabile del codificatore H.264/AVC realizzato da Joint Video Team (JVT) con ITU-T Video Coding Experts Group (VCEG) e ISO/IEC Moving Picture Experts Group (MPEG). Si ricorda che lo standard MPEG-4 è tutt'ora utilizzato per la compressione audio e video in applicazioni come la videotelefonata e la televisione digitale, per la trasmissione di filmati via Web e per la memorizzazione su supporti CD-ROM. Oltre che l'aspetto prettamente tecnico, verrà anche presentato l'utilizzo della codifica scalabile nelle diverse applicazioni che richiedono molteplici bit-rate a seconda dei diversi servizi forniti.

Nel capitolo 3, verrà esposto il concetto di codifica a descrizioni multiple (applicata alla codifica video) e come si presta alla trasmissione in reti di tipo Peer-to-Peer. Allo

scopo di rendere più robusta la trasmissione alle perdite dovute ad una congestione della rete, con un netto peggioramento della qualità video, si vedrà come sia possibile eseguire una codifica congiunta tra la codifica di sorgente (utilizzando il codificatore scalabile) e codifica di canale (con l'uso di codici a blocco di tipo FEC). Poichè nelle reti Peer-to-Peer i singoli peers possono essere visti come "giocatori razionali" di un "gioco" definito nella Teoria dei Giochi, verrà presentata l'idea di applicare la nozione di Equilibrio di Nash al fine di ottimizzare ulteriormente la qualità ricevuta.

Nel capitolo 4, verrà presentata la Teoria dei Giochi ed in particolare verranno trattati gli aspetti utilizzati in questo lavoro: la rappresentazione di gioco (definizione di funzione utilità e strategia), i giochi non cooperativi e equilibrio di Nash (per strategia sia pure che miste). A completamento della tesi, si introdurranno anche i concetti di giochi cooperativi (con utilità trasferibile e non trasferibile) per eventuali spunti futuri. Per rendere più chiara la comprensione verranno forniti degli esempi per esemplificare i concetti esposti.

Il capitolo 5, tratterà come operativamente si è ottenuto da uno stream video scalabile un flusso a descrizioni multiple e successivamente come è stata applicata la Teoria dei Giochi per controllare il traffico IP delle reti simulate. Si farà riferimento agli articoli dai quali si è tratto lo spunto iniziale e si mostrerà quali scelte e modifiche sono state apportate per adattare l'idea di partenza alle esigenze e ai limiti imposti in fase di progetto.

Nel capitolo 6, verranno esposti i risultati sperimentali ottenuti e verranno messi a confronto con lo stato dell'arte presente in letteratura.

Nel capitolo 7, infine verranno riportati i commenti finali e gli sviluppi futuri individuati in questa tesi di Laurea.

Capitolo 1

Introduzione

Oggi giorno, una parte significativa del traffico Internet è generata da applicazioni come file sharing, Video on Demand (VoD), comunicazioni in tempo reale e live media streaming. Tuttavia, queste applicazioni possono essere raggruppate in due classi distinte e con esigenze diverse dal punto di vista della Quality of Service (QoS). La prima classe comprende le applicazioni di file sharing caratterizzate dal fatto che il tempo necessario per scaricare il singolo contenuto non è una prerogativa del servizio. La seconda è composta da tutte le applicazioni real time (istant messaging, video conference, video live streaming ecc..) che pongono dei vincoli ovvi sulla disponibilità immediata (o entro certi limiti) del contenuto richiesto.

Con riferimento alle applicazioni real time, in particolare il video streaming, sul quale si sviluppa l'intero lavoro di tesi, è possibile identificare le seguenti peculiarità:

- *elevati rate di trasmissione.* Le applicazioni video richiedono un alto data rate;
- *limiti di ritardo.* Le applicazioni streaming sono particolarmente sensibili ai ritardi di ricezione dei singoli pacchetti e alle variazioni del ritardo stesso (jitter).

Per far fronte a queste necessità (specialmente per quanto riguarda il contenuto real time) le attuali strutture di rete presentano notevoli difficoltà nel far fronte ai vincoli richiesti dal servizio [1].

I principali punti critici sono:

1. *packet loss.* In [2] è stato osservato che la percentuale di pacchetti persi può essere elevata (fino al 15%) con effetti di tipo burst, ovvero la perdita di più pacchetti consecutivi. Inoltre, protocolli che prevedono meccanismi di gestione delle

perdite (come TCP/IP) risulterebbero proibitivi, a causa delle frequenti ritrasmissioni, sia per i limiti di ritardo imposti dall'applicazione sia per le conseguenti congestioni che causerebbe;

2. *differenziazione della QoS*. Internet non dispone di sistemi efficienti di assegnazione di priorità ai singoli pacchetti. Ad esempio un flusso con determinati vincoli sul ritardo (come un flusso video streaming) verrebbe trattato al pari di un flusso con contenuto statico (immagini o testo) comportando l'inevitabile perdita di qualità del video trasmesso;
3. *costo*. Poichè il costo di trasmissione è proporzionale alla banda utilizzata, un fornitore di servizi di video streaming che fornisce migliaia di flussi contemporaneamente (uno per utente) dovrebbe sostenere una spesa troppo elevata;
4. *scalabilità*. Inviare migliaia di flussi contemporaneamente causa inevitabili "colli di bottiglia" sulla rete. Questo causa l'instabilità del flusso trasmesso, se non il suo blocco totale.

Per far fronte a tali esigenze richieste da questo servizio, sono stati proposti diversi approcci: Media Server Farm che risponde all'idea più semplice di trasmissione Client Server (CS), l'approccio di tipo Content Delivery Network (CDN) basati sempre su CS ma ottimizzando la distribuzione delle risorse, ed infine le reti Peer-to-Peer (P2P).

1.1 Media Server Farm

Nei Media Server Farm uno o più server vengono accentrati in un unico punto affinché possano soddisfare le richieste dei singoli utenti (Fig. 1.1). Questa struttura consente sia di bilanciare il carico tra diversi server sia di reagire ad un improvviso guasto di uno dei server dal momento che verrebbe rimpiazzato da uno dei server associati.

In questa tipologia di rete si può inoltre stabilire la seguente relazione tra la banda necessaria all'uscita del server centrale in relazione al numero di utenti Eq. (1.1).

$$\text{BandaNecessaria} = \text{NumeroUtenti} \cdot \text{QualitàVideo} \quad (1.1)$$

Da questo si osserva che, dato un numero arbitrario di utenti, migliore è la qualità richiesta (bit-rate elevati) e maggiore sarà la banda necessaria. Di conseguenza il costo del servizio aumenta all'aumentare della qualità richiesta.

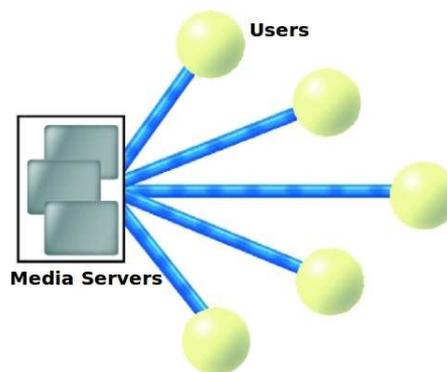


Figura 1.1. Tipologia di rete Media Server Farm

Un secondo aspetto riguarda la scarsa possibilità di scalabilità della struttura in esame; non avendo a disposizione una banda illimitata, il crescere del numero di utenti può portare alla saturazione della banda disponibile, creando un effetto “collo di bottiglia” che congestionerà l’intera rete.

1.2 Content Delivery Network (CDN)

La filosofia CDN [3] [4], basata su una struttura di rete di tipo CS, fu sviluppata per far fronte ai problemi di congestione della rete e di sovraccarico dei server dovute alle numerose richieste di accedere ai contenuti più popolari da parte degli utenti. Ad esempio, la struttura CDN (Fig. 1.2) cerca di ovviare al problema replicando i contenuti più richiesti su diversi server di appoggio chiamati edgserver dislocati in modo da essere il più possibile vicini a tutti gli utenti [5]. Questa scelta permette di risolvere in modo consoni i problemi di congestione della rete in quanto, riducendo il percorso tra la fonte di informazione e l’utente, diminuirà la probabilità di perdita del singolo pacchetto con una conseguente riduzione della congestione dovuta dalle ritrasmissioni dei pacchetti persi.

Inoltre, riducendo la lunghezza del percorso tra server e utente si avrà una conseguente diminuzione del tempo di trasmissione (Round Trip Time - RTT) adatta ad un servizio di tipo real time o streaming dove i tempi di latenza non possono essere trascurati.

Chiaramente la soluzione adottata è sicuramente più scalabile rispetto al Media Server Farm, fermo restando che gli utenti non si concentrino attorno ad un unico edgserver. Tuttavia la problematica del costo non viene risolta, in quanto c’è la necessità di dislocare centinaia di edgserver in posizioni ottimali.

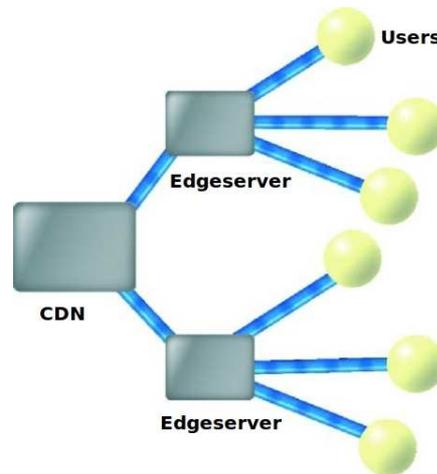


Figura 1.2. Tipologia di rete Content Delivery Network

1.3 Tipologia P2P live streaming

Nelle tipologie di reti descritte in precedenza il singolo utente riceve “passivamente” lo stream trasmesso dal fornitore senza contribuire alla ritrasmissione dell’informazione ricevuta. In tal modo non viene sfruttata la capacità trasmissiva disponibile (in termini di banda) del singolo utente gravando completamente sulle risorse del fornitore. L’idea alla base dell’approccio P2P live streaming consiste in un contributo attivo da parte del singolo utente (peer) che sfruttando la propria banda disponibile, ritrasmette il contenuto ad altri peers che a loro volta ripeteranno la stessa azione creando una sorta di “catena”. Questa interconnessione tra i vari peers permette di realizzare configurazioni di reti dinamiche (in quanto le interconnessioni possono variare nel tempo a seconda del numero e della posizione degli utenti connessi). Inoltre è possibile creare reti con differenti topologie, mutando le prestazioni globali dell’intero traffico.

A tale scopo l’approccio al P2P live streaming è essenzialmente diviso in due classi di reti [6]: uno basato su *Tree network* oppure su *Mesh network*.

1.3.1 Tree network

Il principio sul quale si basa la *Tree network* consiste nell’organizzare i peer in diversi “multiple trees” sui quali inviare l’informazione. In tal modo, prendendo ad esempio il caso di Fig. 1.3(a), il fornitore del servizio dovrà servire solamente due stream contemporaneamente, che si propagheranno (utilizzando le risorse dei nodi intermedi) fino agli utenti più esterni. Tuttavia, oltre alla ovvia efficienza di tale struttura questo

meccanismo presenta delle limitazioni dettate sia dalla profondità degli “alberi” (in quanto oltre un limite massimo di utenti l’albero saturerebbe), sia dalla condizione di “deadlock event”. Quest’ultima condizione è causata quando una parte dei nodi interni lascia “bruscamente” il servizio causando una rottura dell’albero penalizzando i nodi più esterni. Per risolvere tale condizione viene effettuato un aggiornamento periodico della struttura dell’albero comportando però, un ulteriore sovraccarico della rete con informazioni puramente di servizio.

1.3.2 Mesh network

L’approccio sul quale si basa la topologia Mesh network è ispirato al meccanismo “file swarming” ideato per servizi di file sharing (una delle tante applicazioni è BitTorrent [7], [8]) dove i partecipanti formano delle connessioni prive di gerarchia (Fig. 1.3(b)) mantenendo comunque un limite massimo sia di utenti *leechers* che possono essere serviti sia di utenti *seed* da cui attingere le risorse. In linea di principio quando un utente desidera entrare nella rete, contatta un nodo *tracker* (che gestisce la dislocazione delle risorse all’interno della rete) richiedendo i seed che potenzialmente potrebbero fornire il servizio desiderato. Il nodo tracker fornisce (utilizzando algoritmi per l’ottimizzazione della banda e soprattutto del ritardo introdotto) un sottoinsieme di seed che soddisfino le richieste del nuovo entrante permettendo la trasmissione del servizio. Il nuovo nodo appena entrato nella rete diventerà di tipo leecher visto che il suo contributo è totalmente passivo. Non appena avrà scaricato (anche se non in modo completo) parte dello stream, lo metterà a disposizione di altri utenti aumentando l’efficienza della rete. Il fatto di non avere gerarchie e di poter attingere le risorse necessarie da più utenti, rende la topologia Mesh network estremamente flessibile alle eventuali uscite di servizio dei singoli utenti, e una minimizzazione del ritardo di trasmissione che la rendono adeguata a servizi di video streaming.

Oggi, molte sono le applicazioni real time che utilizzano l’approccio P2P live streaming, sempre più orientato verso la topologia di tipo Mesh network. Esempi di applicazioni video streaming sono: BBC iPlayer [9] [10], Swarmplayer [11], Peercast [12] che oltre a rendere il servizio affidabile ne aumentano l’accessibilità.

Anche se i risultati sono molto efficienti, la ricerca sta facendo grandi progressi sull’ottimizzazione dell’efficienza delle reti P2P, in quanto l’assegnazione dei seed non è casuale ma tiene conto dell’intera topologia della rete e delle congestioni al suo interno al fine di ottimizzare le prestazioni [13]. Questo viene gestito dal lavoro congiunto tra l’ISP (Internet Service Provider) che fornisce la topologia fisica della rete al tracker

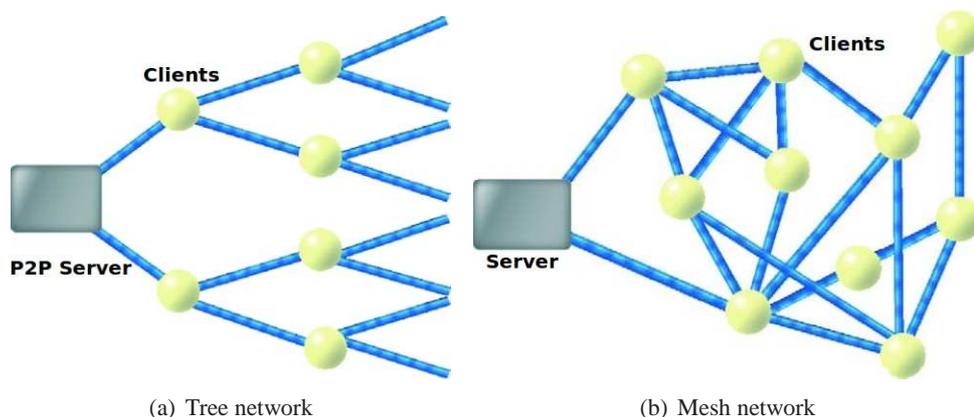


Figura 1.3. Tipologie P2P live streaming

che utilizza queste informazioni ottimizzando l'assegnazione dei seed al nuovo utente. Ovviamente ciò comporta un notevole traffico di dati di servizio, sovraccaricando la rete. Al fine di ridurre questo traffico, molti sono i lavori in corso d'opera quali P4P [14] oppure il protocollo ALTO (Application Layer Traffic Optimization) [15] [16]. Inoltre, altre soluzioni sono già disponibili in rete ad es. Pando Networks [17].

1.4 Codifica a Descrizioni Multiple (MDC)

Con riferimento alle reti P2P live streaming, proprio per la loro flessibilità nell'adattarsi ad una rete fisica già esistente ad un costo contenuto, si stanno cercando nuove applicazioni al fine di incrementare la qualità del servizio real time. Tra le più usate si farà riferimento alla "codifica scalabile" [18] e alla "codifica a descrizioni multiple" (multiple description coding - MDC) [19]. Un'analisi particolare sarà fatta su quest'ultima tecnica di codifica, trattata dettagliatamente nel corso di questo lavoro di tesi. La struttura generale della codifica MDC può essere rappresentata in Fig. 1.4.

L'idea di base consiste nel creare, partendo da una sorgente singola, dei data stream indipendenti, ognuno dei quali può essere decodificato riproducendo il video a bassa qualità. La peculiarità di questo sistema sta nella possibilità di aumentare la qualità del risultato finale in relazione a quanti data stream sono stati ricevuti, dal momento che ogni stream aggiuntivo ne migliora la qualità. Con riferimento alla Fig. 1.4, dei dodici data stream creati solo quattro vengono recapitati correttamente al destinatario, ottenendo una qualità video inferiore rispetto a quella della sorgente. In tal modo è possibile non solo far fronte alle possibili congestioni che causano perdite di data stream

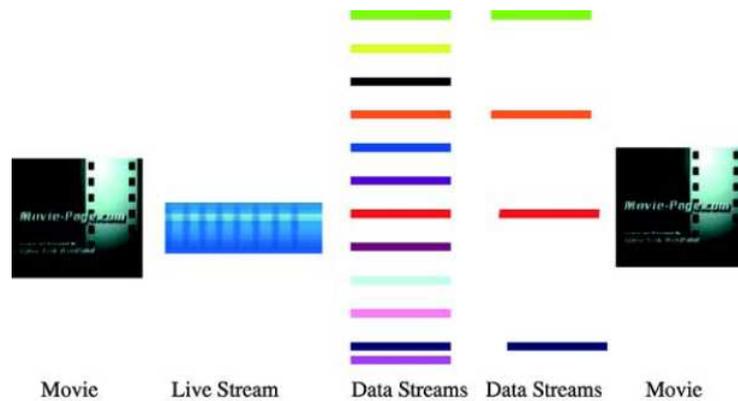


Figura 1.4. Applicazione della codifica a descrizioni multiple

(ricostruendo comunque il video), ma anche di rendere il servizio scalabile in termini di data rate disponibile, in quanto è possibile inviare un sottoinsieme di data stream sfruttando solamente la banda disponibile. Con riferimento alla Fig. 1.5 supponiamo che i data stream (ognuno con rate di 128 kbps) siano distribuiti dal peer 1 al peer 6. Inoltre si supponga che la banda disponibile (in downlink) dal peer destinatario (peer 7) sia di 768 kbps (in modo da poter ricevere contemporaneamente tutti i data stream). In tal caso il peer 7 riceverà i vari data stream da tutti i seed attraverso cammini differenti, riducendo la probabilità di perdita. Nel caso in cui (a causa di una possibile congestione), una parte di data stream fosse persa, sfruttando la loro indipendenza si riuscirebbe comunque a ricostruire la sorgente video con una certa qualità.

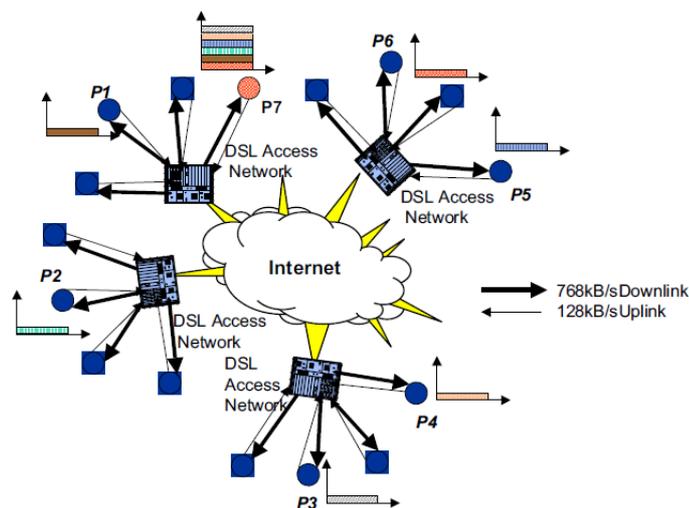


Figura 1.5. Applicazione MDC su reti P2P

Lo stesso ragionamento può essere fatto con una codifica di tipo scalabile, solamente che (come si vedrà descrivendo il codificatore H264/SVC) i vari data stream prodotti (o più comunemente chiamati layer) hanno una gerarchia di decodifica che non consente l'indipendenza dagli altri layer come in MDC, ed inoltre i vari layer necessitano di QoS differenti in base all'importanza. In letteratura questa QoS differenziata può essere ottenuta con tecniche di tipo Unequal Error Protection (UEP) mediante codici di tipo Forward Error Correction (FEC) applicate opportunamente [20] [21].

1.4.1 Trasmissione video su reti P2P tramite MDC

È stato visto come la codifica a descrizioni multiple si presta ad essere utilizzata in una rete P2P. Questo è dovuto al fatto che tale codifica permette sia una buona scalabilità, qualora la rete cambiasse le proprie dimensioni, sia una maggiore affidabilità in quanto la qualità di ricostruzione del video sorgente dipende dal numero di flussi *distinti* ricevuti. La ricerca in questo periodo sta compiendo degli studi al fine di ottimizzare la ricostruzione (il più possibile fedele) del video sorgente. A tale scopo è possibile trarre spunto da concetti della Teoria dei Giochi adattandoli al conteso della trasmissione video su reti P2P [22] [23]. Il principio su cui è basata questa idea sta nel considerare i singoli peer come dei "giocatori razionali" (ognuno con i propri data stream da trasmettere). Al singolo peer verrà data l'opportunità di assegnare ad ogni data stream una priorità (relativa alle code interne ai nodi di reindirizzamento), che si rifletterà sul ritardo e sulla probabilità di perdita del data stream e quindi sulla qualità finale del video ricostruito. Ovviamente tali priorità verranno assegnate allo scopo di ottenere una condizione che "soddisfi" tutti i peer e che massimizzi la qualità finale (equilibrio di Nash). Altri lavori invece, adottano un approccio di assegnazione "fissa" di tali priorità. In tal modo i singoli peer sono tra loro indipendenti non considerando le possibili scelte effettuate dagli altri [24].

Tuttavia in letteratura si trovano algoritmi (Rainbow Network Problems [25]) che cercano di ottimizzare la qualità finale percepita minimizzando il numero di data stream *replicati* e ricevuti dal singolo cliente, in quanto ridondanti quindi privi di informazione aggiunta.

Capitolo 2

Codificatore JSVM

2.1 Introduzione

Gli standard internazionali come H.261, MPEG-1, MPEG-2 Video, H.263, MPEG-4 Visual, and H.264/AVC hanno avuto un ruolo fondamentale nell'affermazione delle applicazioni video digitali. Tali standard hanno permesso un'alta interoperabilità tra prodotti differenti, favorendo un'alta flessibilità per l'implementazione e l'ottimizzazione di diverse applicazioni. In questo campo lo standard H.264/AVC rappresenta lo stato dell'arte nel campo della codifica video. Inoltre, rispetto agli standard precedenti, permette una significativa riduzione del rate necessario a pari livello di qualità, con conseguente aumento dell'efficienza di codifica.

Tuttavia, per far fronte alla richiesta di adattamento “on-the-fly” (ad esempio, nella velocità di visualizzazione o di elaborazione video in alcuni dispositivi) oppure alla necessità di variare le condizioni di trasmissione (dettate dalla qualità richiesta da alcune applicazioni), è stato necessario adottare un approccio di codifica scalabile.

Al giorno d'oggi la codifica video viene usata in molte applicazioni ad esempio: multimedia messaging, telefonia video, video conferenza su reti wireless, Internet video streaming, diffusione di contenuti video ad alta definizione, ecc. che hanno trovato enorme impiego specialmente su tipologie di reti Internet e wireless. In tali reti la scalabilità è stata necessaria da un lato a causa delle continue variazioni delle condizioni di trasmissione, dall'altro per la disomogenea varietà dei dispositivi di decodifica (vedi Fig. 2.1). In tali situazioni risulta opportuno fornire un flusso dati che sia flessibile alle condizioni che la rete presenta e allo stesso tempo permetta l'interoperabilità tra dispositivi differenti.

Per far fronte a tutto ciò lo standard H.264/SVC codifica il bit stream video in un

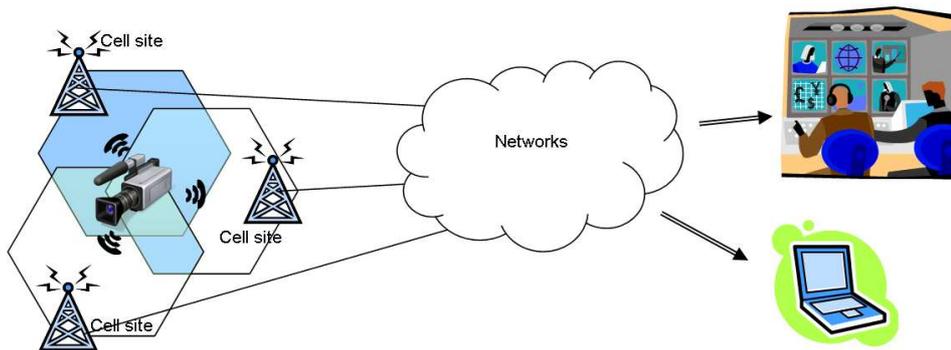


Figura 2.1. Esempio di video streaming su rete Internet con dispositivi eterogenei

flusso scalabile con un moderato incremento della complessità di decodifica, la quale prevede di adattare il formato e il bit rate alle esigenze del contesto di trasmissione fornendo nel contempo una buona robustezza nelle trasmissioni con perdite. Tutto questo al fine di incrementare l'efficienza di trasmissione e memorizzazione da parte delle molteplici applicazioni. In Fig. 2.2 è rappresentato il principio di scalabilità in H.264/SVC come estensione dello standard precedente H.264/AVC.

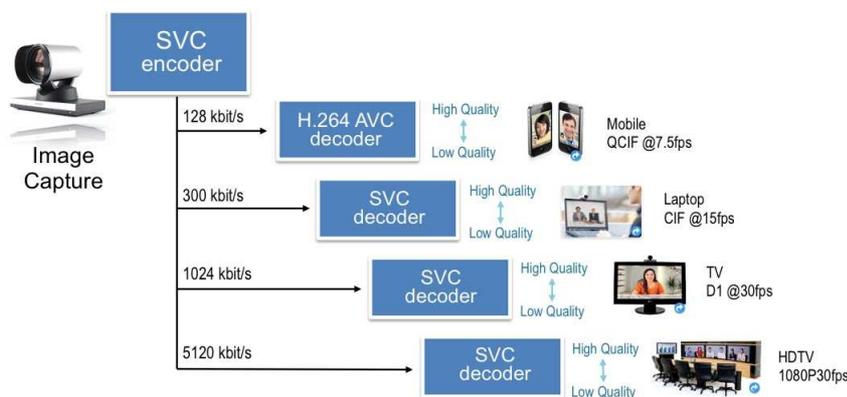


Figura 2.2. Esempio di Codifica Video Scalabile (SVC)

2.2 Scalabilità e applicazioni

In generale un flusso video è detto scalabile quando una parte dello stream (rispettando opportune gerarchie) può essere rimossa, ottenendo un sub-stream a sua volta decodificabile con qualità inferiore rispetto al video sorgente, ma con qualità maggiore rispetto alla minimo flusso decodificabile. Eventuali bitstream che non godono di questa carat-

teristica vengono detti single-layer bitstream. Tuttavia si possono distinguere diversi tipi di scalabilità: scalabilità temporale, spaziale e di qualità. La scalabilità temporale e spaziale rappresentano il contenuto sorgente con un frame rate inferiore o con una risoluzione inferiore a partire da una partizione dello stream codificato. Nella scalabilità di qualità il sub-stream mantiene la stessa risoluzione spazio-temporale ma con minore qualità visiva (riferita ad un diverso valore di Peak Signal to Noise Ratio - PSNR). Tale scalabilità, molte volte viene denominata anche fidelity scalability o SNR scalability. I diversi tipi di scalabilità possono per giunta essere combinati, creando un unico flusso con caratteristiche spazio-temporali diverse (vedi Fig. 2.3).

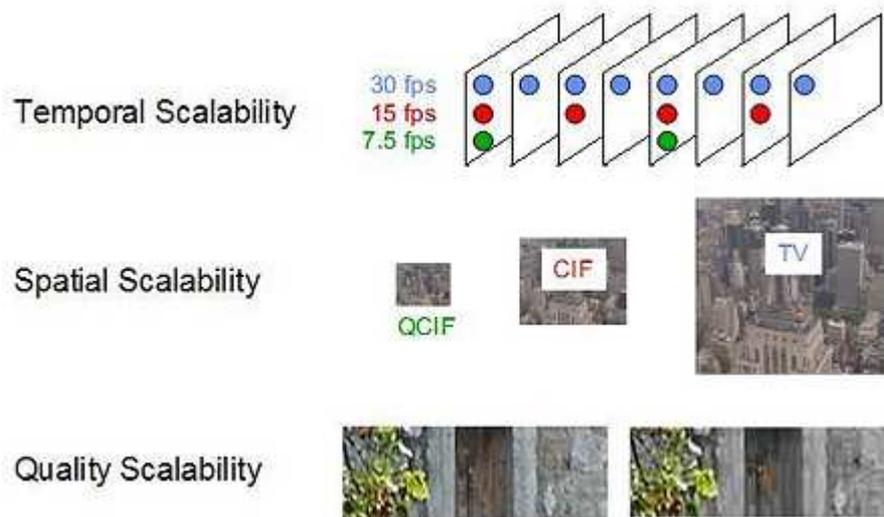


Figura 2.3. Diversi tipi di scalabilità forniti dal codificatore H264/SVC

Si vedrà ora come SVC e di conseguenza la codifica scalabile possano portare alle diverse applicazioni degli effettivi benefici. Si consideri ad esempio un servizio di video streaming su di una rete eterogenea (dove allo stesso tempo vengono richiesti più flussi video con caratteristiche di qualità e di risoluzioni spazio-temporale diverse). Con l'uso di un flusso SVC sarà necessario codificare il video sorgente una sola volta (con la massima qualità e risoluzione) ottenendo i flussi con minor risoluzione scartando parte del flusso codificato. Ad esempio un utente con risorse limitate (risoluzione del display, capacità di calcolo o ridotta carica di batteria) necessiterà di decodificare solamente una parte del flusso trasmesso. Similmente, in una trasmissione multicast (ad esempio P2P) dove i terminali presentano diversi bit rate diviene necessario fornire solamente una parte dello stream sorgente. Infine, potrebbe venir richiesto un flusso VGA e nel contempo lo stesso file sorgente ma con risoluzione QVGA, ecco che con

la codifica SVC ambedue le richieste verrebbero soddisfatte.

Un ulteriore beneficio apportato da SVC sta nel fatto che in un flusso scalabile siano presenti parti del bitstream con differente importanza. Questo viene sfruttato mediante l'uso di tecniche Unequal Error Protection (UEP) specialmente nei casi dove la banda disponibile varia nel tempo con un relativo aumento del packet loss. Nelle reti Media-Aware Network Elements (MANEs) il terminale fornirà dei messaggi di feedback (con le proprie condizioni di banda e dello stato del canale) a seconda dei quali sarà inviato parte dell'intero flusso codificato (vedi Fig. 2.4). Quindi la perdita di informazione (dovuta alle congestioni) può essere parzialmente evitata con un conseguente aumento della robustezza del servizio di trasmissione video.

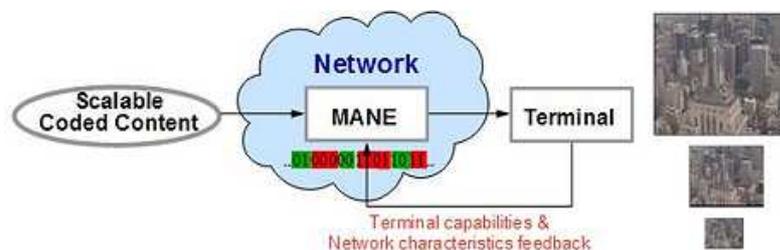


Figura 2.4. Adattamento del flusso video scalabile in una rete Media-Aware Network Elements

SVC può inoltre essere utilizzato in applicazioni di controllo o sorveglianza, nelle quali la sorgente non deve solamente essere visualizzata su diversi monitor o Personal Digita Assistant (PDA), ma deve prestarsi per essere archiviata. Con SVC le parti ad alta risoluzione possono essere memorizzate fino alla scadenza di un tempo limite, dopo il quale solamente la parte a bassa risoluzione viene archiviata. Anche se gli schemi usati da SVC per la scalabilità offrono una vasta disponibilità di funzionalità, il profilo scalabile di altri standard non viene pienamente sfruttato, questo è dovuto principalmente all'elevato costo computazionale e alla bassa efficienza di codifica fornita dalla loro scalabilità spaziale e di qualità. Contrariamente la scalabilità temporale viene ampiamente utilizzata (vedi H.264/AVC) principalmente per la sostanziale efficienza di codifica che si riesce a trarne.

H.264/AVC è lo standard più recente di codifica video non scalabile ed è oggi diffuso in molte applicazioni. Per questo motivo tale standard è stato ulteriormente esteso con una versione scalabile (per l'appunto H.264/SVC). Il successo di questa versione scalabile dipende essenzialmente da alcuni requisiti:

- possiede un'efficienza di codifica (per ogni flusso scalabile) molto vicina a quella

del codificatore H.264/AVC;

- la complessità di decodifica è sugli stessi livelli (anche se di poco maggiori) rispetto a H.264/AVC;
- supporta scalabilità di tipo spaziale, temporale e di qualità;
- è compatibile con lo standard precedente H.264/AVC;
- dopo la codifica il bitstream prodotto è facile da adattare alle esigenze dei servizi richiesti.

2.3 Storia SVC

Tecniche di codifica ibrida (come H.264/AVC) ampiamente utilizzate, presentano una compensazione del moto basata su Temporal Differential Pulse Code Modulation (DPCM). DPCM è caratterizzato dall'uso di un loop sincronizzato sia al codificatore che al decodificatore, tuttavia effetti di “drift” causano fastidiosi artefatti. Tale effetto affligge comunque la codifica scalabile qualora parte dello stream venisse rimosso (a causa di adattamenti alle condizioni della rete).

Contrariamente alla codifica DPCM, una codifica a trasformata non presenta effetti di “drift”, proprio per questo fin dall'inizio si è pensato di applicare a SVC una soluzione di moto compensazione basata su trasformata 3D Wavelet. Il progresso di tale tecnologia diede inizio nell'Ottobre 2003 ad una ricerca da parte di MPEG di soluzioni innovative per un possibile sviluppo di SVC. Ad Aprile 2004 ci furono quattordici proposte, di cui dodici basate su 3D Wavelet, mentre solo due prevedevano di estendere la struttura già esistente dello standard in vigore H264/AVC (che utilizza una compensazione basata su un sistema simile al DPCM). Nell'Ottobre 2004 la proposta dell'Image Communication Group of the Heinrich-Hertz-Institute (HHI) venne presa in considerazione da MPEG, ed ebbe inizio il progetto SVC. Nel Gennaio 2005 MPEG e Video Coding Experts Group (VCEG) definirono il progetto come “Amendment of H.264/MPEG-4 AVC”. Infine nel Luglio 2007 il progetto ebbe le approvazioni finali.

Sebbene il progetto iniziale prevedeva una compensazione del moto basata su 3D Wavelet, l'idea fu rimossa dalle specifiche per ridurre la complessità del codificatore e decodificatore, aumentando l'efficienza di codifica. A tale scopo si ritornò sull'idea del DPCM con alcuni accorgimenti per evitare l'effetto “drift” da cui era affetto.

2.4 Aspetti fondamentali H.264/AVC

Concettualmente, lo standard H.264/AVC si divide in due sottoaree funzionali principali: *Video coding Layer* (VCL) e *Network Abstraction Layer* (NAL). Mentre il VCL fornisce la rappresentazione codificata della sorgente, il NAL contiene tutti gli header al fine di manipolare in modo semplice lo stream VCL.

2.4.1 Network Abstraction Layer (NAL)

L'intero flusso video codificato è organizzato in NAL units, cioè in pacchetti, ognuno con un numero intero di byte. Ogni pacchetto inizia con un header di un byte che segnala il tipo di dati contenuti. I byte rimanenti sono adibiti al payload. I NAL units possono essere classificati in VCL NAL units, che contengono slice codificate, e non-VCL NAL units, che contengono informazioni aggiuntive. Le più importanti non-VCL NAL units sono: parameter sets e Supplemental Enhancement Information (SEI). I parameter sets contengono informazioni sulla sequenza e sui singoli frame, mentre il SEI fornisce informazioni che possono rendere più efficiente la decodifica e la visualizzazione dello stream (non sono indispensabili per la decodifica del flusso codificato). Un insieme di NAL units con specifiche proprietà viene riferito come un *access unit*. La decodifica di un access unit equivale a decodificare un singolo frame. Il concetto può essere esteso riferendoci ad un insieme di access units consecutive che rappresenta una porzione del video codificato. Quindi, una sequenza video codificata rappresenta una parte indipendente e decodificabile dell'intero bit stream di NAL units. Tale bit stream comincia con access unit definita come Instantaneous Decoding Refresh (IDR), la quale indica che questa access unit e le successive access units possono essere decodificate indipendentemente dai frame precedenti.

2.4.2 Video Coding Layer (VCL)

Il VCL dello standard H.264/AVC segue l'idea di una codifica video a blocchi. Sebbene la stessa idea sia simile a quella degli standard precedenti come H.261, MPEG-1 Video, MPEG-2 Video, H.263, o MPEG-4 Visual, H.264/AVC presenta nuove caratteristiche che forniscono dei miglioramenti significativi nell'efficienza di compressione rispetto agli standard precedenti. I singoli frame sono partizionati in blocchi più piccoli seguendo la suddivisione in *macroblocchi* i quali possono essere raggruppati in *slices*. Ogni macroblocco comprende una regione rettangolare pari a 16×16 campioni di luminanza e, nel caso di video con formato 4:2:0 anche di 8×8 campioni di cromaticità.

Si ricorda che nel formato 4:2:0 i campioni di cromaticità (C_b e C_r) vengono sottocampionati di un fattore due sia verticalmente che orizzontalmente rispetto ai campioni di luminanza (Y). I campioni di ogni macroblocco vengono predetti, sia dal punto di vista temporale che spaziale, e i rispettivi residui vengono codificati mediante una codifica a trasformata. Infine, i macroblocchi vengono raccolti in una o più slice (indipendenti tra i vari frame).

A seconda del tipo di predizione utilizzata si distinguono diversi tipi di slice:

1. I-slice: *Intra-picture predictive*, cioè viene utilizzata una predizione spaziale tra i blocchi vicini;
2. P-slice: *Intra-picture predictive* e *inter-picture Predictive*, cioè viene utilizzata una predizione sia spaziale tra i blocchi vicini, sia una predizione temporale con frame precedenti;
3. B-slice: *Intra-picture predictive*, *inter-picture Predictive*, *inter-picture Bipredictive*, cioè viene utilizzata una predizione spaziale tra i blocchi vicini e una predizione temporale tra i frame precedenti e successivi. Il relativo risultato è ottenuto come media pesata dei due.

Per le I-slice H.264/AVC definisce diverse direzioni di predizione al fine di approssimare i blocchi. Per le componenti di luminanza la predizione può essere eseguita su blocchi di diverse dimensioni, 4×4 , 8×8 oppure 16×16 , mentre per le componenti di cromaticità la dimensione è sempre la stessa, partendo da 16×16 e scalato in base al formato di sottocampionamento della cromaticità adottato.

Per le P-slice e B-slice H.264/AVC definisce una predizione temporale utilizzando più frame di riferimento e blocchi di dimensione variabile. Tali dimensioni possono essere 16×16 , 16×8 , 8×16 o 8×8 . Se la dimensione dei blocchi risulta 8×8 è ulteriormente possibile suddividere il blocco in altri blocchi di dimensione 8×4 , 4×8 , 4×4 . Per le P-slice viene trasmesso un solo motion vector per ogni blocco. Inoltre, le reference pictures (possono essere più di una) vengono segnalate al decodificatore attraverso un parametro *reference index*.

Per le B-slice sono disponibili due liste di reference pictures e per ognuna è possibile scegliere blocchi di dimensione 16×16 , 16×8 , 8×16 o nel caso di blocchi da 8×8 varrà il principio di partizionamento introdotto per le P-slice. Inoltre il metodo di predizione può essere scelto tra *list0*, *list1*, o *biprediction*. Mentre per i modi di predizione *list0* e *list1* la predizione viene eseguita scegliendo una sola tra le due liste, nel modo *biprediction* viene eseguita una somma pesata tra i due valori predetti.

Inoltre sono presenti altri due modi di predizione chiamati *direct mode* per le B-slice e *skip mode* per le P-slice e B-slice nel quale i dati relativi ai motion vector e ai riferimenti alle liste di pictures sono derivati da quelli già trasmessi (riducendo la ridondanza di informazione).

Successivamente i residui vengono codificati mediante una codifica a trasformata. H.264/AVC specifica un insieme di trasformazioni eseguite su blocchi di dimensione differente. Mentre per i blocchi intra la dimensione della trasformata è uguale a quella del blocco utilizzato, per i blocchi moto compensati (per la luminanza) la dimensione del blocco non scende sotto 8×8 , ed è possibile scegliere una dimensione tra 4×4 o 8×8 . Inoltre, per tali blocchi viene prima eseguita una codifica a trasformata su blocchi 4×4 e successivamente si applica una trasformata di Hadamard sui coefficienti DC. Una trasformazione simile avviene anche per macroblocchi intra di dimensione 16×16 . Inoltre, per evitare possibili errori in fase di decodifica, vengono eseguite le esatte trasformazioni inverse. H.264/AVC quantizza tutti i coefficienti con *quantizzatore uniforme*. È possibile scegliere uno tra i 52 passi di quantizzazione disponibili tramite il parametro QP. Le operazioni di scalabilità (ad esempio nel caso di scalabilità di qualità) viene eseguita con incrementi logaritmici del passo di quantizzazione.

Per ridurre l'effetto di deblocking (tipico della codifica a trasformate) H.264/AVC applica un "adaptive deblocking filter" presente nel loop di predizione della moto compensazione.

H.264/AVC supporta due tipi di codifica entropica, ambedue sfruttando il principio di *context based adaptivity* al fine di migliorare le performance. Mentre la codifica *context-based adaptive variable-length coding* (CAVLC) usa codici a lunghezza variabile applicata ai soli coefficienti della trasformata, la codifica *context-based binary arithmetic coding* (CABAC) utilizza un codice aritmetico per incrementare l'efficienza di codifica. Non a caso quest'ultima codifica incrementa l'efficienza di codifica del $10 \div 15\%$ rispetto l'CAVLC.

In H.264/AVC l'ordine di codifica e di visualizzazione sono completamente disaccoppiati. Tuttavia, qualsiasi frame può essere marcato come *reference picture* e servirà per la moto compensazione dei frame successivi. Si necessita quindi di un *decoded picture buffer* (DPB), che può memorizzare fino a 16 frames ed è controllato mediante un comando chiamato *management control operation* (MMCO). Inoltre la lista di frame utilizzati per predire P- o B-slice può essere ricostruita dai frame disponibili nel DPB mediante il comando *reference picture list reordering* (RPLR).

Per rendere più flessibile il partizionamento dei frame tra le slice, in H.264/AVC viene introdotto il concetto di *slice group*. I macroblocchi possono essere

organizzati in gruppi di slice tramite un *slice group map*. Questo slice group map, che viene specificato nel contenuto del picture parameter set e in alcune slice di header, assegna un unico *slice group identifier* per ogni macroblocco del frame. Per ottenere la singola slice sarà necessario scandire in modo *raster scan* i macroblocchi con lo stesso slice group identifier.

Come negli standard precedenti, una *picture* comprende un insieme di slice che rappresentano l'intero frame o un *field* dello stesso (ad esempio un video interlacciato dove un frame può essere trasmesso come un'unica immagine oppure vengono trasmessi due field separati). Lo standard H.264/AVC supporta un macroblock-adaptive swiching tra frame e field codificati. A riguardo, una coppia di macroblocchi verticali viene considerata come una *single coding unit*, che può essere trasmessa o come due macroblocchi vicini oppure interlacciata come bottom field e top field.

2.5 Concetti base dell'estensione H.264/SVC

Una delle caratteristiche principali che ha reso H.264/SVC uno degli standard più apprezzati nel campo della codifica video, sta nell'efficienza della codifica e della complessità con cui è stato realizzato. Questo è dovuto al fatto che essendo l'estensione scalabile dello standard (già esistente) H.264/AVC, eredita tutti i core già sviluppati e ottimizzati. Oltretutto è possibile, partendo da un flusso H.264/SVC ottenere un flusso a single layer H.264/AVC, restando compatibile con quest'ultimo standard. Inoltre, i nuovi tools sono stati aggiunti per rendere efficiente la scalabilità che viene richiesta.

2.5.1 Scalabilità temporale

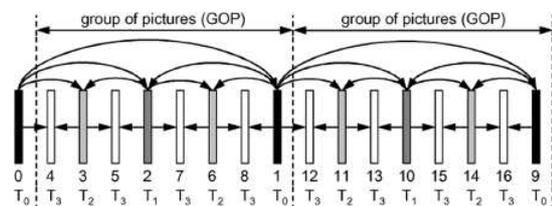
Un bitstream fornisce una scalabilità temporale quando un insieme di access units può essere diviso in un base layer e uno o più enhancement layer, secondo quanto segue. Si supponga che un temporal layer sia identificato da T , che va da 0 (base layer) ed è incrementato di 1 ad ogni altro temporal layer successivo. Quindi, per ogni intero k è possibile scartare ogni temporal layer con T maggiore di k ottenendo un flusso decodificabile.

Generalmente, i codificatori ibridi eseguono la scalabilità temporale solamente predicendo i frame da reference picture con temporal layer identifier minore o uguale a quello del frame da codificare. H.264/AVC fornisce una grande flessibilità della scalabilità temporale in quanto presenta un reference picture memory control, che permette di codificare i picture con diverse dipendenze, limitate solamente dalla capacità

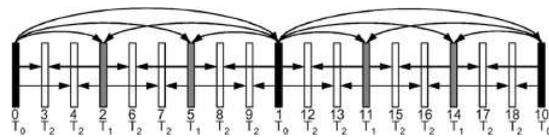
del DPB. Di conseguenza per effettuare tale scalabilità temporale non sono stati fatti cambiamenti da H.264/AVC.

Predizione gerarchica

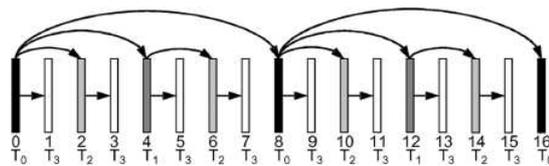
La scalabilità temporale può essere efficientemente ottenuta con il concetto di B-picture gerarchiche (vedi Fig. 2.5(a)). I frame dell'enhancement layer sono tipicamente codificati come B-picture dove le reference list 0 e 1 comprendono i frame temporalmente precedenti e successivi con temporal layer minore a quello del picture predetto. Con ciò ogni insieme di temporal layer T_0, \dots, T_k può essere decodificato indipendentemente dai temporal layer T_i con $i > k$. Inoltre, l'insieme di picture tra due pictures con temporal base layer viene definito come *Group Of Pictures* (GOP).



(a) Predizione con B-picture gerarchiche (struttura diadica)



(b) Predizione gerarchica non diadica



(c) Predizione gerarchica di tipo encoding/decoding

Figura 2.5. Predizione gerarchica nella scalabilità temporale

Sebbene la struttura di predizione appena descritta fornisce, oltre alla scalabilità temporale anche una buona efficienza di codifica (come sarà mostrato nel seguito) essa rappresenta solo un caso speciale. Questo perchè in generale la struttura di predizione gerarchica per la predizioni temporale, può essere combinata con un concetto di *multiple reference picture* introdotto in H.264/AVC. Questo consiste nel creare una reference picture list con una o più picture anche dello stesso layer del picture da predire. Quindi

le strutture di predizione gerarchica non sono ristrette solamente al caso *diadico* (Fig. 2.5(a)).

La Fig. 2.5(b) rappresenta una struttura di predizione *non diadica*, la quale fornisce due sottosequenze con $1/9$ e $1/3$ del frame rate disponibile. È possibile notare che la struttura del base layer temporale può variare in modo arbitrario, al fine di migliorare l'efficienze di codifica. Inoltre, la scelta di questa struttura può non essere costante nel tempo, adattandola ad ogni situazione. Da notare com'è possibile regolare il ritardo introdotto dalla codifica e decodifica, restringendo la predizione tra la picture che dev'essere visualizzata e quella che la succede.

In Fig. 2.5(c) viene mostrata una predizione che non tiene conto delle picture future. Tale struttura permette lo stesso grado di scalabilità della Fig. 2.5(a) ma introducendo un ritardo trascurabile paragonato a 7 della struttura 2.5(a). Tuttavia questo miglioramento comporta una perdita di efficienza della codifica.

L'ordine di codifica utilizzato nelle strutture di predizione temporale deve prevedere che le reference pictures debbano essere codificate (in ordine temporale) prima che vengano utilizzate per la moto compensazione. Questo può essere assicurato da molte strategie che differiscono per ritardo introdotto e per quantità di memoria utilizzata.

L'efficienza di codifica introdotta dalla predizione gerarchica dipende fortemente dal passo di quantizzazione scelto nei differenti layer. Intuitivamente, i frames del layer temporale di riferimento dovrebbero essere codificati con una maggiore fedeltà in quanto servono come riferimento per la moto compensazione degli altri layer. Per quest'ultimi, il passo di quantizzazione potrebbe essere incrementato visto che esso influenzerà un numero minore di picture predette. Di conseguenza il passo di quantizzazione aumenterà direttamente con il numero di layer.

Un modo avanzato per il calcolo del passo di quantizzazione ottimale, consiste nell'analisi della funzione Rate Distortion, causando un eccessivo dispendio computazionale. Nello standard H.264/SVC viene utilizzato il seguente algoritmo che presenta delle buone performace in molti dei test eseguiti. Si parte da un passo di quantizzazione QP_0 applicato al base layer, per i layer con $T > 0$ il passo di quantizzazione sarà calcolato come $QP_T = QP_0 + 3 + T$. Sebbene questa tecnica presenti delle fluttuazioni del valore del PSNR in un group of pictures, il video ricostruito non presenterebbe eccessivi artefatti.

2.5.2 Scalabilità spaziale

Per supportare la scalabilità spaziale, SVC segue l'approccio del *multilayer coding* adottato anche in MPEG-2 Video, H.263, e MPEG-4 Visual. Ogni layer corrisponde ad una risoluzione spaziale che il codificatore può fornire, ed è definita da uno spatial layer o *dependency identifier D*. Per il base layer, tale parametro è uguale a $D = 0$ ed è incrementato di 1 per lo spatial layer successivo. In ogni spatial layer la compensazione del moto e la predizione intra vengono eseguite come nel codificatore a singolo layer. Per aumentare l'efficienza viene utilizzato un nuovo meccanismo chiamato *inter-layer prediction* (Fig. 2.6).

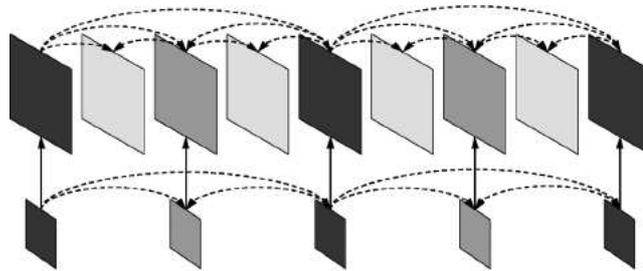


Figura 2.6. Struttura multilayer con inter-layer prediction nella scalabilità spaziale

Per ridurre lo spazio di memoria necessario e la complessità del decoder, SVC utilizza per tutti i layer lo stesso ordine temporale di codifica dei frame. Inoltre (in un determinato istante) l'insieme delle differenti risoluzioni spaziali prende il nome access unit e deve essere trasmessa in ordine crescente dei corrispondenti spatial identifier D . Tuttavia, come si vede dalla Fig. 2.6 il base layer non è presente ad ogni istante, rendendo possibile una scalabilità combinata tra temporale e spaziale.

Inter-Layer prediction

Lo scopo principale di tale predizione consiste nell'utilizzare quanta più informazione possibile del base layer al fine di massimizzare l'efficienza del rate-distortion relativo agli enhancement layer. In MPEG-2 Video, H.263 e MPEG-4 Visual, l'unica predizione inter-layer consiste nel predire i campioni dell'enhancement layer sovracampionando il base layer. Sebbene i campioni (opportunamente interpolati) rappresentino tutta l'informazione contenuta nel base layer, questa non è la soluzione migliore per una predizione inter-layer. Questo è dovuto dal fatto che l'inter-layer prediction deve competere con la temporal-prediction, specialmente per sequenze lente ad alta risoluzione, in quanto una predizione temporale darà un risultato decisamente migliore ri-

spetto ad un sovracampionamento del base layer. Per aumentare l'efficienza di codifica (in una codifica scalabile spaziale), in SVC vengono definiti due tipi di inter-layer prediction: *predizione dei macroblocchi e dei relativi vettori di compensazione del moto e predizione dei residui*.

SVC permette di scegliere liberamente tra intra e inter-layer prediction a seconda della sorgente a disposizione. L'inter-layer prediction può avvenire solamente nella relativa access unit, utilizzando per la predizione del layer con indice D , un layer con indice $D' < D$. Il layer utilizzato come riferimento viene detto *reference layer* ed è indicato nell'header delle slices dell'enhancement layer. Poichè SVC permette un inter-layer prediction sia per il moto che per i residui, il codificatore dovrà allineare le strutture temporali dei vari layer.

Sebbene SVC supporti una scalabilità spaziale con risoluzioni arbitrarie, verrà presentata l'inter-layer prediction nel semplice caso che la risoluzione raddoppi sia in altezza che in larghezza.

Inter-Layer motion prediction In tale funzione SVC mette a disposizione un nuovo tipo di macroblocco segnalato con un *base mode flag*. Per tale macroblocco solo i residui sono trasmessi evitando di spedire informazioni aggiuntive sul tipo di predizione intra o di moto compensazione. Quando il parametro *base mode flag* è 1 e il corrispondente blocco 8×8 nel reference layer è codificato intra, allora tale macroblocco viene predetto mediante *inter-layer intra-prediction* (che verrà trattato nel seguito). Quando il blocco di riferimento è inter-coded, anche il macroblocco dell'enhancement layer viene inter-coded. In questo caso la predizione avviene considerando il vettore di compensazione di moto del reference layer relativo al blocco 8×8 corrispondente e si parlerà di *inter-layer motion prediction*.

Il partizionamento dei macroblocchi nell'enhancement layer viene eseguito sovracampionando il corrispondente blocco 8×8 del reference layer. Se il blocco 8×8 non è partizionato in ulteriori sottoblocchi, nemmeno quello relativo all'enhancement layer lo sarà, altrimenti ogni sottoblocco $M \times N$ corrisponderà ad un partizione $2M \times 2N$ del blocco dell'enhancement layer. Inoltre, il vettore di compensazione di moto relativo all'enhancement layer sarà ottenuto scalando le due componenti del vettore di un fattore 2. L'utilizzo della inter-layer motion prediction viene indicato attraverso il parametro *motion prediction flag*. Se tale parametro è posto a 1 i reference index non sono codificati nell'enhancement layer ma vengono utilizzati i reference index dei blocchi corrispondenti del reference layer e i relativi motion vector vengono scalati come indicato precedentemente. Se il *motion prediction flag* è 0 allora la reference picture list

viene codificata nell'enhancement layer e si utilizza la procedura di compensazione del moto specificato in H.264/AVC.

Inter-Layer residual prediction Questo tipo di predizione può essere applicata a tutti i macroblocchi inter-coded, che possono essere sia i macroblocchi convenzionali oppure quelli segnalati dal *base mode flag*. La inter-layer residual prediction viene individuata mediante il parametro *residual prediction flag*. Quando il flag è 1, il blocco 8×8 corrispondente del reference layer viene interpolato mediante un filtro bilineare e utilizzato per predire i residui dell'enhancement layer (così solamente le differenze vengono codificate nell'enhancement layer). È da notare che il sovracampionamento viene eseguito sui coefficienti trasformati evitando il filtraggio del deblocking filter che causerebbe delle alterazioni del segnale (vedi Fig. 2.7(a) e 2.7(b)).



(a) Residual prediction con filtraggio sui bordi dei blocchi della trasformata (b) Residual prediction senza filtraggio sui bordi dei blocchi della trasformata

Figura 2.7. Visione dell'enhancement layer con e senza filtraggio sui bordi dei blocchi trasformati

Inter-Layer Intra-Prediction Quando un macroblocco dell'enhancement layer viene codificato con il *base mode flag* a 1 e il blocco 8×8 corrispondente al reference layer viene codificato intra, allora l'enhancement layer è ottenuto mediante *inter-layer intra-prediction*, sovracampionando il reference layer. Per l'interpolazione dei campioni di luminanza viene utilizzato un filtro FIR a 4 coefficienti monodimensionale, eseguendo il filtraggio sia in orizzontale che in verticale. Inoltre il filtraggio viene eseguito tenendo conto dei blocchi vicini codificati intra eliminando eventuali effetti di bordo. Nel caso in cui tali blocchi non fossero intra-coded si applicano particolari algoritmi di estensione dei bordi. Questo permette di evitare la ricostruzione dei blocchi predetti

temporalmente nel reference layer (single-loop decoding). Per evitare possibili disturbi, prima del sovracampionamento viene applicato il filtro di deblocking descritto in H.264/AVC (Fig. 2.8).

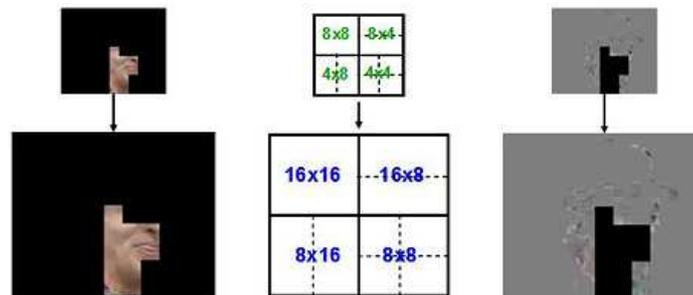


Figura 2.8. Inter-Layer Intra e Residual Prediction

2.5.3 Scalabilità di qualità

La scalabilità di qualità può essere considerata come caso particolare di scalabilità spaziale, dove tutti i layer hanno uguale risoluzione. Questo caso è anche detto *coarse-grain quality scalability* (CGS). Inoltre viene utilizzato lo stesso meccanismo di inter-layer prediction ma senza utilizzare il sovracampionamento e il deblocking per i macroblocchi codificati intra. In questo tipo di scalabilità l'inter-layer intra- e residual-prediction vengono eseguiti sui coefficienti della trasformata. Nel CGS la scalabilità avviene quantizzando i coefficienti con passo di quantizzazione più fine al crescere del reference layer. Tuttavia questo sistema permette di ottenere un limitato numero di bit rate disponibili (tipicamente il numero di rate sarà identico al numero di layer). Al fine di rendere più flessibile il bitstream è stata adottata una variante del CGS, chiamata *Medium-Grain Scalability* (MGS). Tale tecnica prevede che ogni NAL unit possa essere scartata fornendo in questo modo una scalabilità di qualità organizzata in pacchetti rendendo l'intero bit stream più gestibile. Inoltre SVC permette di organizzare i coefficienti della trasformazione in più slice (Fig. 2.9).

Per tenere traccia dei coefficienti, il primo e l'ultimo sono segnalati nell'header della relativa slice. Questo permette di organizzare l'informazione relativa ad una certa qualità in diverse NAL units ognuna delle quali contenente i relativi coefficienti.

Nelle Figure 2.10(a) e 2.10(b) vengono rappresentate le curve di Distortion Rate relative alla codifica della sequenza "Foreman" mediante una codifica scalabile di qualità costituita da 3 layer, utilizzando il metodo MGS e CGS.

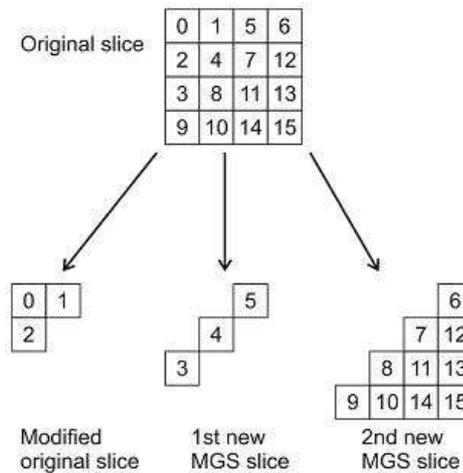


Figura 2.9. Disposizione dei coefficienti nelle NAL units

2.5.4 Controllo del drift nella scalabilità di qualità

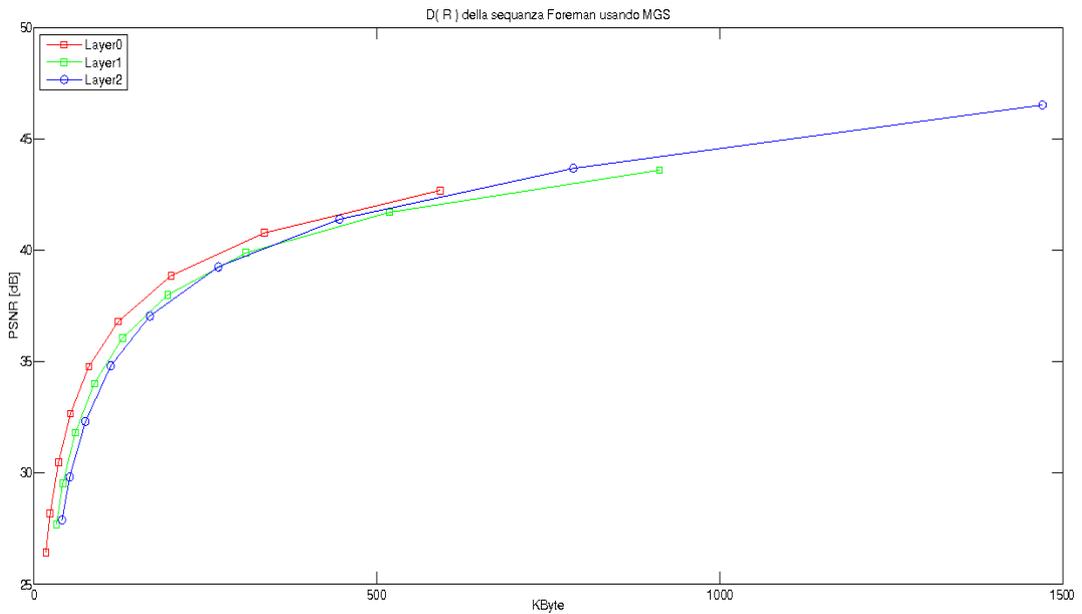
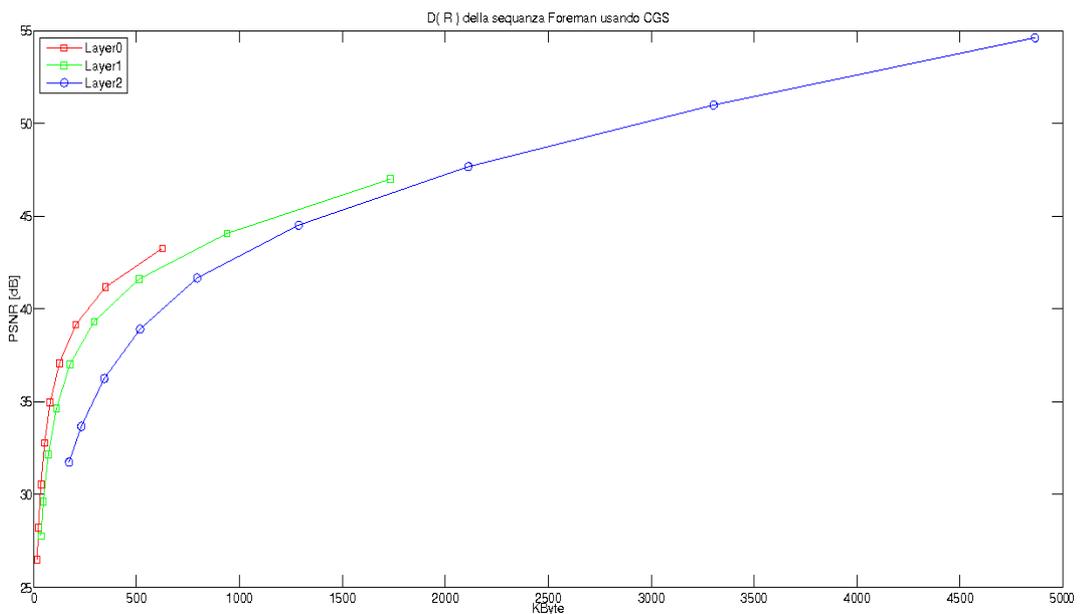
Il processo con il quale avviene la moto compensazione dev'essere studiato al fine di ottenere un compromesso tra efficienza di codifica dell'enhancement layer e l'effetto *drift*. Quest'ultimo si manifesta quando nel loop di moto compensazione viene perso il sincronismo tra codificatore e decodificatore, ad esempio a causa della perdita di un pacchetto dal bitstream. In Fig. 2.11 si possono osservare diverse strutture che trovano un compromesso tra la gestione dell'effetto drift e l'efficienza di codifica.

Controllo del base layer

Viene utilizzato nella codifica *fine-grain quality scalability* (FGS) utilizzata in MPEG-4 Visual presentando una struttura di predizione come in Fig. 2.11(a). Da come si può osservare la moto compensazione viene eseguita solamente sul base layer quindi un'eventuale perdita di pacchetti (dell'enhancement layer) non ha alcun impatto sul loop di moto compensazione. Tuttavia, questo va a discapito di una scarsa efficienza di codifica, addirittura minore del caso a singolo layer.

Controllo dell'enhancement layer

Questa struttura (vedi Fig. 2.11(b)) viene utilizzata nello standard MPEG-2 Video, dove il layer con qualità maggiore viene impiegato per la moto compensazione. Questo permette un'alta efficienza di codifica e una bassa complessità di realizzazione in quanto solamente una sola reference picture dev'essere immagazzinata in ogni istante.

(a) $D(R)$ della sequenza "Foreman" utilizzando MGS(b) $D(R)$ della sequenza "Foreman" utilizzando CGS**Figura 2.10.** Confronto tra MGS e CGS per la sequenza "Foreman"

Tuttavia, l'effetto drift può essere controllato solamente con l'aggiunta di frame di tipo intra.

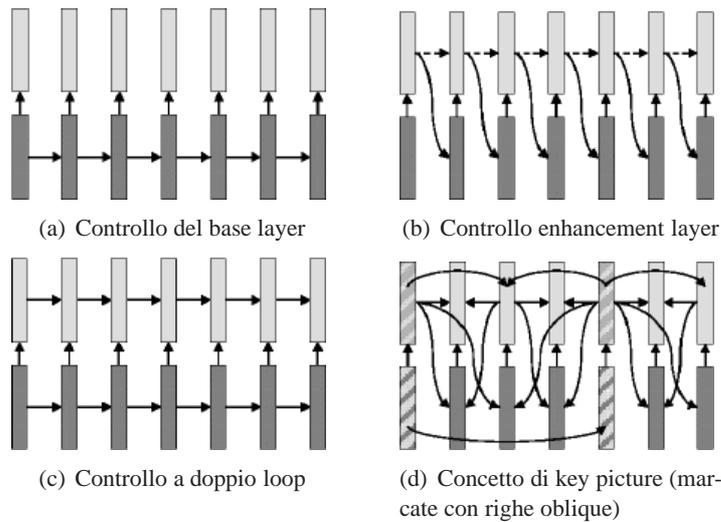


Figura 2.11. Esempi di tradeoff tra efficienza di codifica per l'enhancement layer e robustezza all'effetto drift

Controllo a doppio loop

Questa struttura (Fig. 2.11(c)) è simile a quella adottata per la scalabilità spaziale fornita nei codificatori MPEG-2 Video, H.263 e MPEG-4 Visual. Sebbene il base layer non è influenzato dalla perdita di qualche pacchetto dell'enhancement layer, la qualità di quest'ultimo layer è molto sensibile all'effetto drift.

Controllo mediante key picture

La scalabilità MGS presente in SVC utilizza l'approccio di tipo *key picture*. Per ogni picture viene trasmesso un flag che segnala se il base layer o l'enhancement layer della reference picture sono utilizzati per la predizione nella compensazione del moto. Per limitare la memoria utilizzata, viene anche trasmesso un secondo parametro che segnala se il base layer ricostruito viene o meno salvato nel DPB. Inoltre, per ridurre l'overhead delle key pictures, viene stabilito se i parametri relativi alla compensazione del moto rimangono gli stessi sia per il base layer che per l'enhancement. La Fig. 2.11(d) mostra come il concetto di key picture sia adatto alla struttura di predizione gerarchica.

Inoltre si nota che tutte le pictures del layer temporale (più lento) sono trasmesse come key picture e solo per queste il base layer viene inserito nel DPB. Di conseguenza (per questo layer temporale) gli effetti di drift vengono evitati. Per aumentare l'efficienza di codifica tutti gli altri layer temporali utilizzano il layer a più alta qua-

lità per la predizione di moto compensazione. Poichè le key picture servono come sincronizzazione tra coder e decoder, il drift è limitato all'interno del GOP.

2.5.5 Estrazione del bit stream

Per estrarre un substream con un particolare bit rate partendo da un flusso con scalabilità di qualità (ad esempio MGS) esistono molti modi. Ad esempio il bit rate desiderato può essere raggiunto scartando diverse NAL unit. Un metodo molto semplice consiste nello scartare randomicamente pacchetti MGS fino ad ottenere il bit rate voluto. In alternativa, si prevede di assegnare delle priorità alle NAL unit scartando quelle con priorità più bassa. Se il bit rate raggiunto non dovesse soddisfare le richieste allora si passerà alla priorità immediatamente superiore.

2.6 Scalabilità combinata

Il concetto generale di combinazione tra scalabilità temporale, spaziale e di qualità è rappresentato in Fig. 2.12.

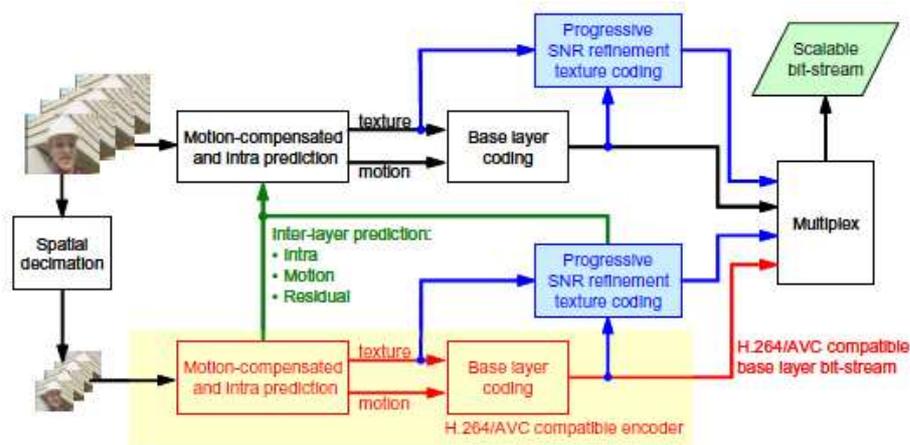


Figura 2.12. Struttura generale del codificatore SVC

Principalmente la struttura di codifica utilizzata da SVC è organizzata in *dependency layers*. La singola dependency layer rappresenta una specifica risoluzione spaziale. Può succedere che la risoluzione di ambedue le dependency layer sia identica. In tal caso i layers forniscono una coarse-grain scalability (CGS) che li differenzia in qualità. I dependency layer sono definiti da un *dependency identifier D*. Inoltre, la risoluzione spaziale non può diminuire da un dependency identifier a quello successivo. Per ogni

dependency layer sono definiti i concetti di stima del moto, intra prediction come nel caso di single layer. La ridondanza tra più layer viene sfruttata mediante i concetti di inter-layer prediction.

I quality layer all'interno di un dependency layer sono definiti tramite un quality identifier Q . Per questi quality layer verrà applicata una predizione di tipo inter-prediction.

Per un quality identifier $Q > 0$ viene sempre utilizzata una inter-layer prediction mediante il quality identifier $Q - 1$, vedi Fig. 2.13.

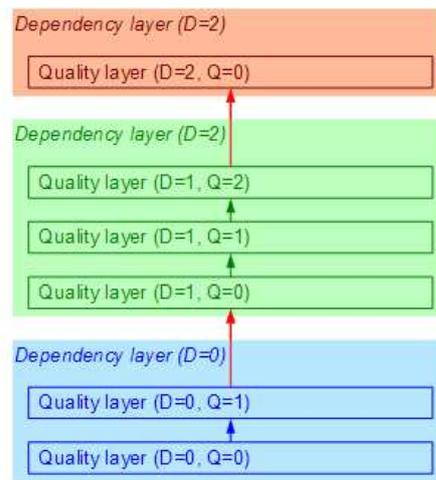


Figura 2.13. Rappresentazione delle dipendenze in SVC

Un'importante differenza tra i concetti di dependency layer e quality refinement sta nel fatto che è possibile cambiare tra differenti dependency layer solamente in precisi "switching point". Tuttavia il passaggio tra differenti quality refinement layer è possibile in ogni access unit.

2.7 Conclusioni

Mettendo a confronto lo standard H.264/SVC con i suoi predecessori si possono notare importanti differenze:

- la possibilità di utilizzare una struttura di predizione gerarchica per ottenere una scalabilità temporale composta da più layer incrementando in tal modo l'efficienza di codifica e fornire una maggiore flessibilità delle scalabilità sia spaziale che di qualità;

- il codificatore SVC presenta nuovi metodi di inter-layer prediction del moto e dei residui, incrementando l'efficienza di codifica nella scalabilità sia spaziale che di qualità;
- introduce il concetto di key picture per un controllo efficiente dell'effetto drift nella scalabilità di qualità con predizione gerarchica;
- presenta un solo loop di moto compensazione nella scalabilità spaziale o di qualità riducendo di molto la complessità del decodificatore (nell'ordine di quello a single layer).

Capitolo 3

Codifica a Descrizioni Multiple

3.1 Introduzione

Molti dei sistemi di trasmissione attualmente utilizzati generano contenuti attraverso sorgenti scalabili, spedendo lo stream mediante opportuni protocolli (ad es. TCP/IP) che permettono il controllo e la ritrasmissione degli eventuali pacchetti persi. Sebbene questa tecnica si presti ad adattarsi a molte applicazioni, in alcuni casi produce dei ritardi eccessivi per il tipo di servizio richiesto (ad es. il video streaming e altri servizi real time).

Supponiamo che L pacchetti di uno stream video scalabile (codificati in modo che il pacchetto i -esimo dipenda dal pacchetto $i - 1$ -esimo) vengano spediti per essere decodificati in ricezione secondo l'ordine dato dall'indice $i = \{1, 2, \dots, L\}$. Nell'ipotesi in cui ogni pacchetto corrisponda ad un layer diverso, la qualità del flusso ricevuto aumenta in relazione al numero di pacchetti consecutivi ricevuti. In questo caso l'ordine di arrivo dei pacchetti diventa un fattore critico; ad esempio, se vengono ricevuti i pacchetti $i = \{1, 2, 4, 5, \dots, L\}$, la qualità sarà proporzionale a quella fornita dai soli primi due pacchetti.

Per tale motivo, la codifica scalabile presenta delle ottime prestazioni se i pacchetti sono spediti e ricevuti corrispondentemente secondo un determinato ordine. In caso contrario la decodifica entra in uno stato di “stallo” e in attesa del pacchetto o dei pacchetti mancanti ricostruisce la sequenza con qualità più bassa. Sfortunatamente il protocollo TCP/IP comporterà questo inconveniente in quanto il ritardo di ritrasmissione è tipicamente superiore a quello di inter-arrivo di due pacchetti consecutivi. Tuttavia molti dei sistemi di codifica tutt'oggi usati, si basano proprio sul fatto che i pacchetti inviati vengano ricevuti in un ordine prestabilito.

Se le perdite diventano inevitabili, diviene necessario creare delle rappresentazioni del video originale che utilizzino qualsiasi pacchetto ricevuto, indipendentemente dall'ordine di arrivo. La codifica a descrizioni multiple (Multiple Description - MD) soddisfa proprio questo vincolo, incrementando la robustezza della trasmissione alle eventuali perdite, anche se a discapito di una minor efficienza di codifica.

La codifica a descrizioni multiple nasce negli anni '70 presso i Laboratori Bell, nell'ambito di ricerca sulla trasmissione vocale su rete telefonica. Una caratteristica richiesta dal canale telefonico è l'affidabilità. A tal proposito per garantire il funzionamento continuo in caso di guasto alla linea, le comunicazioni venivano deviate su linee secondarie. Per ovviare a questo dispendioso procedimento Boyle e Miller svilupparono il concetto di *channel splitting* (divisione di canale). Il segnale (ad esempio segnale vocale) viene campionato alla frequenza di Nyquist e successivamente sottocampionato di un fattore due, creando due flussi non affetti eccessivamente da aliasing. In questo modo se un flusso venisse perso, sarebbe in ogni caso possibile riprodurre quello ricevuto, ottenendo una qualità discreta.

3.1.1 Modello base di Descrizioni Multiple (channel splitting)

Jayant, ricercatore di Bell Laboratories propose uno schema innovativo per la trasmissione di un segnale vocale mediante la tecnica di channel splitting (vedi Fig. 3.1).

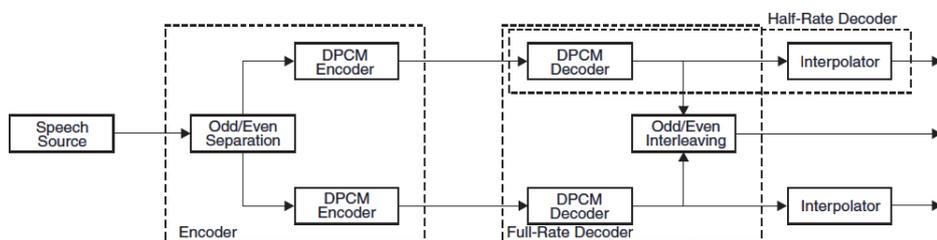


Figura 3.1. Codifica vocale utilizzata nel channel splitting proposto da Jayant

La differenza di tale sistema con gli standard dell'epoca, consisteva nell'utilizzare una frequenza di campionamento del segnale vocale di 12 kHz anziché di 8 kHz, in modo da ridurre l'aliasing in ognuno dei due canali creati. I campioni pari e dispari venivano successivamente codificati mediante due codificatori DPCM diversi e infine quantizzati.

Dal lato ricevitore erano presenti due decodificatori DPCM che ricostruivano i rispettivi canali ed inoltre era presente un codificatore che interlacciava i due flussi ri-

costruendo (a meno di un eventuale rumore di quantizzazione) il segnale di partenza. Inoltre, quest'ultimo decodificatore ricostruiva il flusso mancante (pari o dispari), qualora il canale corrispondente presentasse delle perdite, mediante una semplice interpolazione lineare. Un'analisi matematica del sistema viene descritta in [26], dalla quale è possibile stabilire che se la correlazione tra i campioni pari e dispari venisse rimossa prima della codifica (ad es. attraverso una predizione lineare) le performance del sistema risulterebbero inaccettabili.

3.1.2 Channel splitting per sorgenti incorrelate

Nel caso la sorgente non avesse ridondanza (ad esempio applicando una predizione prima di separare i campioni nel caso di channel splitting) e si usasse un semplicissimo quantizzatore uniforme a 16 livelli (rate totale di 4 bit/simbolo), si osserverebbe che lo spezzare i quattro bit in 2 coppie (una per canale) in modo da ricostruire l'eventuale canale mancante, risulterebbe inefficace (Fig. 3.2(a)).

Un altro modo di usare un quantizzatore uniforme nella tecnica channel splitting è rappresentato in figura 3.2(b), dove sono utilizzati due quantizzatori uniformi a 3 bit/simbolo con le relative soglie sovrapposte (ottenendo 16 livelli di quantizzazione come nel caso precedente). Questo permette di ricostruire il campione con più precisione proporzionalmente al numero di canali ricevuti. Ad esempio, ricevendo $q_1(x) = 110$ e $q_2(x) = 101$, allora x dev'essere nell'intervallo $[7/16, 9/16]$ e quindi sarà ricostruito come $1/2$.

Sebbene la ricostruzione del campione anche in mancanza di un canale è buona, quando ambedue i canali sono ricevuti correttamente tale quantizzatore presenta un rate totale di 6 bit/simbolo. In tale condizione funzionerebbe al pari del quantizzatore di Fig. 3.2(a) con un'occupazione di banda maggiore.

Per ovviare a questo problema si ricorre al quantizzatore di Fig. 3.2(c) (ispirato dall'idea dei codificatori di Reudink) dove le soglie relative ad ogni livello sono evidentemente disgiunte, ma il funzionamento è analogo a quello di Fig. 3.2(b). Tuttavia ognuno dei due quantizzatori presenta 6 livelli (circa 2.6 bit/simbolo) e complessivamente si ottiene una risoluzione di 16 livelli con rate indubbiamente minore rispetto al caso precedente.

3.1.3 MD viste nella Teoria dell'Informazione

La tecnica del channel splitting ispirò alla seguente domanda:

Se la sorgente viene descritta tramite due descrizioni separate, quali sono gli effetti

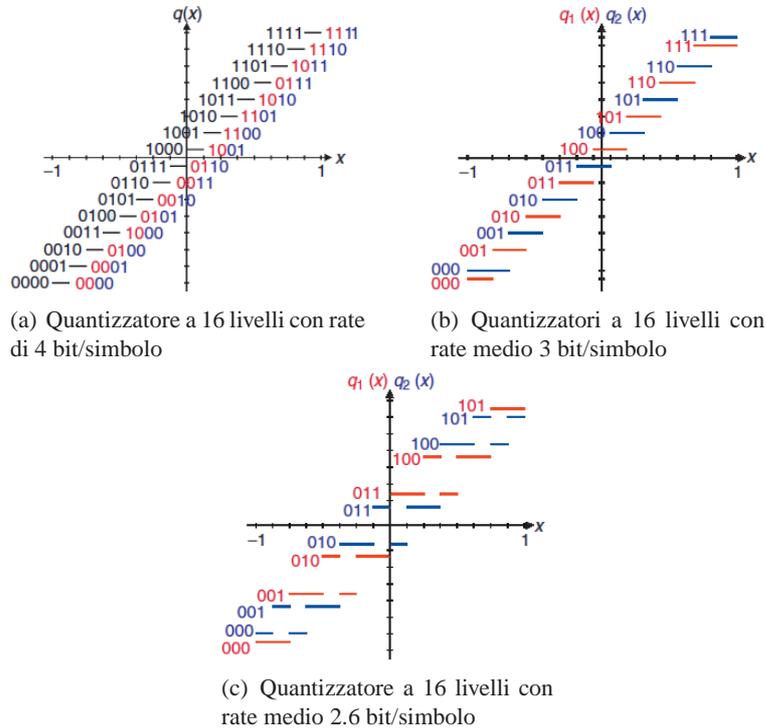


Figura 3.2. Quantizzatori nel caso di sorgente incorrelata nel channel splitting

sulla qualità finale se le descrizioni vengono prese singolarmente o congiuntamente?

Il quesito appena esposto viene anche detto *multiple description problem* (MD problem).

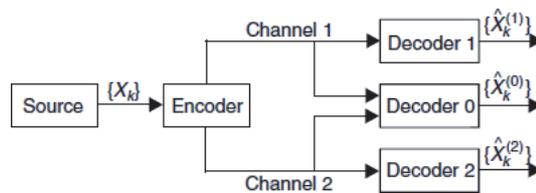


Figura 3.3. Modello su cui si basa l'MD problem

Per dare una risposta si consideri la situazione di Fig. 3.3. Ad un codificatore viene fornita una sorgente di simboli $\{X_k\}_{k=1}^N$ che verrà trasmessa a tre ricevitori su due canali privi di errori. Un decodificatore (*decoder centrale*) riceve l'informazione da ambedue i canali, mentre i decodificatori laterali ricevono i segnali dai relativi canali. Il rate di trasmissione lungo il canale i viene indicato con R_i bit/simbolo, mentre la distorsione di ricostruzione verrà indicata con D_i .

A questo punto, si realizza una descrizione con rate R_1 tale da minimizzare D_1 e allo stesso modo si realizza una seconda descrizione con rate R_2 che minimizza D_2 da spedire sul Canale 2. Fermo restando queste ipotesi non si può affermare che la descrizione con rate $R_1 + R_2$ sia quella che minimizza D_0 . Allo stesso modo non si può dire che una descrizione tale che $R_1 + R_2$ minimizza D_0 implichi che $D_1(R_1)$ e $D_2(R_2)$ siano minimi.

Tale modello si presenta perciò come un problema teorico il cui scopo sarà quello, data una sorgente e una misura di distorsione, di trovare i possibili valori della quintupla $(R_1, R_2, D_0, D_1, D_2)$ vedi [26].

3.2 Applicazioni della codifica MD

Perdite dovute a errori o congestioni della rete causano fastidiosi artefatti e distorsioni nella visualizzazione del video decodificato. Sebbene tecniche di *error concealment* (Fig. 3.4) possono essere d'aiuto, a causa dell'elevato costo computazionale, vengono usate molto raramente. Questo spiega la necessità di utilizzare delle tecniche di codifica che siano in grado di recuperare o ricostruire gli eventuali dati persi.



(a) Video decodificato senza error concealment



(b) Video decodificato con error concealment

Figura 3.4. Esempio di error concealment

Una tecnica di tipo Automatic Retransmission reQuest (ARQ) potrebbe essere adatta in una rete di tipo P2P ma non presenta una predisposizione alla trasmissione real time a causa degli elevati ritardi dovuti dalla ritrasmissione. Un'altra tecnica potrebbe essere quella di codificare il video proteggendolo solamente con codici Forward Error Correction (FEC), comportando però costi troppo elevati di elaborazione e ritardo introdotto. Inoltre presenta un effetto *cliff* determinato dal fatto che se la quantità di errori

supera la capacità correttiva del codice l'informazione ricevuta diviene completamente inutilizzabile.

Tuttavia una codifica di tipo scalabile è inappropriata per trasmissioni su canali con perdite in quanto la struttura gerarchica con la quale avviene la decodifica non la rende per niente adatta a tale situazione. Contrariamente, una codifica a descrizioni multiple non richiede alcun tipo di feedback e non presenta effetti di tipo cliff. Inoltre presenta un miglioramento graduale della qualità (più descrizioni vengono ricevute e migliore sarà la qualità) offrendo allo stesso tempo una trasmissione scalabile (il numero di descrizioni che vengono trasmesse è proporzionale alla qualità desiderata).

Per queste ragioni l'uso della codifica a descrizioni multiple risulta vantaggiosa ogni qualvolta una o entrambe le seguenti situazioni si verificano:

- **gli utenti sono affetti da perdite di almeno una descrizione.** Nella reti P2P questo indica che le descrizioni, a causa di congestioni potrebbero essere perse. Talvolta può anche succedere che gli utenti inviino di proposito un sottoinsieme di descrizioni (ad esempio per rispettare i limiti di banda imposti). Contrariamente, se la trasmissione fosse priva di perdite non si necessiterebbe di alcun tipo di codifica;
- **sono accettabili per gli utenti uno o più sotto-livelli di qualità dell'informazione.** Dal momento che il segnale ricostruito mediante MDC presenta una qualità proporzionale al numero di descrizioni ricevute, gli utenti devono essere in grado di supportare tale scalabilità.

3.3 Come generare MDs

Non è facile progettare ed implementare uno schema di codifica video tramite descrizioni multiple. Inoltre, essendoci già molti standard come MPEG-2, H.263 e H.264/SVC (che implementano strutture di tipo scalabile), la creazione di un nuovo standard più complesso non risulterebbe cosa da poco.

Esistono molte tecniche disponibili per la generazione di descrizioni multiple, ma la scelta migliore può essere fatta seguendo diversi criteri:

- **compatibilità.** Possibilità di usare un codificatore standard per ogni descrizione, possibilmente restando compatibili con gli standard precedenti;
- **semplicità.** Minima richiesta di memoria e di calcolo computazionale;

- **efficienza.** In condizioni ideali (assenza di perdite) la distorsione del segnale decodificato, dev'essere minima rispetto agli attuali standard di codifica scalabile.

3.3.1 Multiple Description mediante IFPD

Uno dei metodi più semplici per la creazione delle singole descrizioni, consiste nel partizionare i simboli della sorgente in più sottoinsiemi correlati (tanti quanti sono il numero di descrizioni desiderate) e successivamente eseguire la compressione di tali insiemi per ottenere la descrizione, da cui l'acronimo Independent Flux Poliphase Downsampling (IFPD). In fase di decodifica, verrà utilizzato un sistema di interpolazione per ottenere le descrizioni perse da quelle ricevute. Una separazione possibile consiste nel dividere i campioni pari da quelli dispari (come nel caso del channel splitting), oppure sono possibili separazione multidimensionali (Fig. 3.5) utilizzate essenzialmente per trasmissione video.

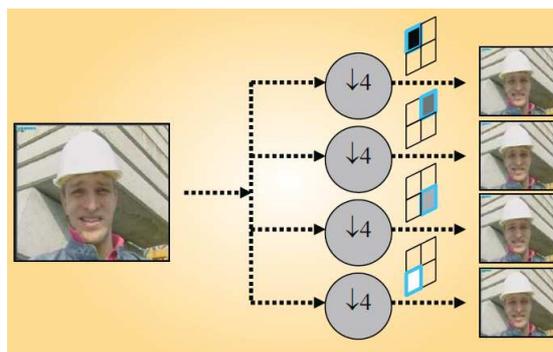


Figura 3.5. Generare descrizioni multiple mediante IFPD

La sequenza da codificare viene divisa in sottosequenze che vengono codificate indipendentemente. Questo può essere fatto in uno stadio di pre-processing (Fig. 3.5). In fase di decodifica è presente uno stadio di post-processing dove le sequenze vengono riordinate per ricreare l'originale (Fig. 3.6). Tale sistema è conosciuto come "Independent Flux Pliphase Downsampling Multiple Description" (IF-PDMD) [27].

3.3.2 Codifica Scalabile e Unequal Error Protection

Un metodo elementare per creare due descrizioni potrebbe essere quello di inviare due volte lo stesso flusso. Ad ognuno potrebbe venir applicata la più sofisticata tecnica di compressione e ogni qualvolta si ricevesse anche solo una descrizione il risultato

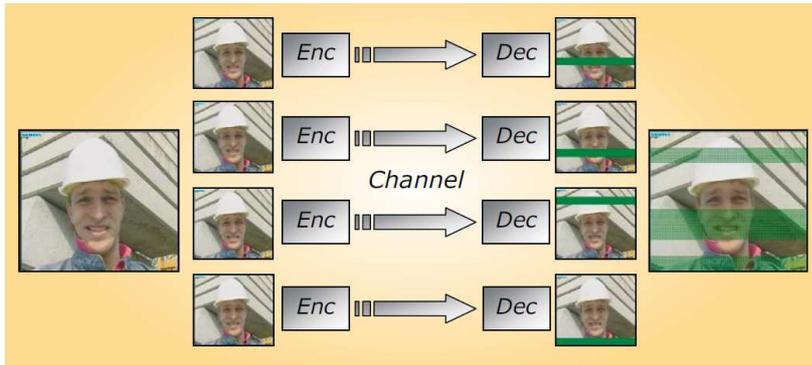


Figura 3.6. Pre-processing, codifica, trasmissione, decodifica e post-processing di IF-PDMD

sarebbe quanto meglio si potesse fare. Tuttavia nel caso di ricezione di ambedue le descrizioni non si avrebbe alcun guadagno.

Un metodo più flessibile consiste nel ripetere solo una frazione dell'informazione disponibile, ad esempio quella più importante, adattandosi bene al caso di codifica scalabile. Per produrre due descrizioni da R bit, codifichiamo il flusso originale (ad es. mediante un codificatore scalabile) in modo da ottenere uno stream con rate $(1 - \zeta)R$ con $\zeta \in [0, 1]$. I primi ζR bit che sono i più importanti vengono ripetuti in ambedue le descrizioni. I rimanenti $(1 - \zeta)R$ bit vengono invece divisi tra le due descrizioni. In Tab. 3.1 i bit di ridondanza sono stati rappresentati con $\boxed{\zeta R}$.

Descrizione 1	ζR	$(1 - \zeta)R$
Descrizione 2	$\boxed{\zeta R}$	$(1 - \zeta)R$

Tabella 3.1. Come creare due descrizioni da un flusso scalabile

Questo sistema fa in modo che i bit più importanti vengano protetti con un codice a “ripetizione” con rate $1/2$, mentre gli altri rimangono privi di protezione. Questo tipo di tecnica viene anche detta Unequal Error Protection (UEP).

Mediante UEP è possibile creare più di due descrizioni (in teoria un numero arbitrario). L'idea consente di creare L descrizioni utilizzando L codici a ripetizione con rate $1/L, 2/L, 3/L, \dots, 1$. In Tab. 3.2 si rappresenta il caso di 4 descrizioni, dove ogni colonna rappresenta la parola di codice associata ad ogni singolo codice a ripetizione.

Al fine di minimizzare la distorsione (causata dalle perdite di descrizioni introdotte dal canale) pur mantenendo una ridondanza tale da soddisfare i limiti di banda occupa-

Descrizione 1	$\zeta_1 R$	$\zeta_2 R$	$\zeta_3 R$	$\zeta_4 R$
Descrizione 2	$\zeta_1 R$	$\zeta_2 R$	$\zeta_3 R$	$\zeta_4 R$
Descrizione 3	$\zeta_1 R$	$\zeta_2 R$	$\zeta_3 R$	$\zeta_4 R$
Descrizione 4	$\zeta_1 R$	$\zeta_2 R$	$\zeta_3 R$	$\zeta_4 R$

Tabella 3.2. Creare 4 descrizioni con codici a ripetizione

ta, si vedrà in seguito come sia possibile ottenere un numero “arbitrario” di descrizioni partendo da un flusso scalabile.

3.3.3 Multiple Description Coding usando codici FEC

Si supponga di voler ottenere n descrizioni indipendenti, partendo da un determinato flusso scalabile. In tal caso sarà necessario dividere l'intero flusso in n layer $\{1, 2, 3, \dots, n\}$ (Fig. 3.7), dove $R_{i-2} - R_{i-1}$ rappresenta la dimensione del layer i -esimo. Come si vedrà, questo partizionamento seguirà una precisa metodologia al fine di ridurre la distorsione causata da eventuali perdite di qualche descrizione.

Avendo a disposizione n layer, verrà applicata la tecnica UEP, dividendo il layer i -esimo in i parti, disponendole nella matrice FEC come in Fig. 3.7

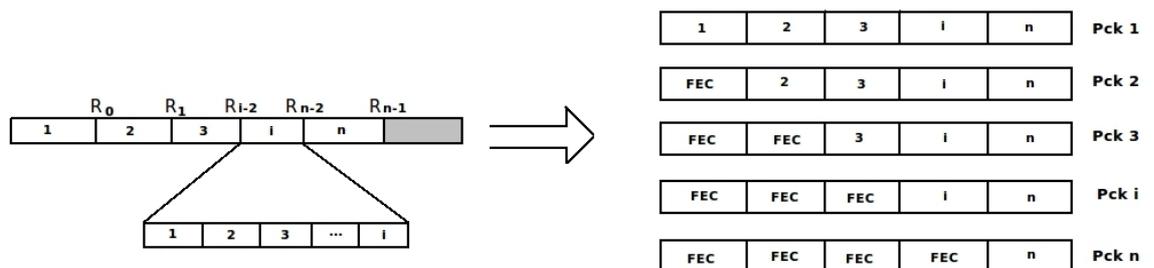


Figura 3.7. Creare n descrizioni indipendenti da un flusso scalabile

Ogni colonna della matrice avrà i pacchetti di informazione e $n - i$ pacchetti di protezione. Quindi, il layer i -esimo verrà ricostruito in presenza di al più $n - i$ descrizioni perse. Questo permette di ottenere una protezione del layer proporzionale alla

sua importanza (in quanto in fase di decodifica è necessario rispettare la gerarchia dei layer).

Lo scopo dell'algoritmo consiste nel trovare l'allocazione ottima della n -upla $\bar{R} = (R_0, R_1, R_2, R_3, \dots, R_{n-1})$ in modo da minimizzare la distorsione media "Average Distortion" ($E[D]$) Eq. (3.1).

$$E[D] = \sigma_x^2 q_0 + \sum_{j=0}^{n-1} q_{j+1} D(R_j) \quad (3.1)$$

Con q_j si indica la probabilità che su n descrizioni ne vengano ricevute j , mentre con σ_x^2 si indica la distorsione massima cioè la varianza dell'immagine di partenza.

Disponendo i vari layer nella matrice FEC di Fig. 3.7 si otterrà una dimensione complessiva (di tutte le descrizioni) pari a quella di Eq. (3.2):

$$R_t = R_0 \cdot n + \frac{R_1 - R_0}{2} \cdot n + \dots + \frac{R_{n-1} - R_{n-2}}{n} \cdot n \quad (3.2)$$

che può essere semplificata come:

$$R_t = \sum_{j=0}^{n-1} \alpha_j \cdot R_j \quad (3.3)$$

dove:

$$\alpha_j = \frac{n}{(j+1)(j+2)} \quad \text{con } j = 0, 1, \dots, n-2 \quad (3.4)$$

e $\alpha_{n-1} = 1$.

Quindi, date le condizioni iniziali del problema:

1. numero di descrizioni volute n ,
2. dimensione della singola descrizione $L \Rightarrow R^* = n \cdot L$,

il problema di ottimizzazione sarà quello rappresentato dal sistema di Eq. (3.5), dove oltre alla minimizzazione della distorsione si dovrà rispettare le limitazioni imposte

anche dai vincoli di dimensione totale, monotonicità e divisibilità.

$$\left\{ \begin{array}{l} \min E[D] = \sigma_x^2 q_0 + \sum_{j=0}^{n-1} q_{j+1} D(R_j) \\ R_t = \sum_{j=0}^{n-1} \alpha_j R_j \leq R^* \text{ (vincolo dim. totale)} \\ R_0 \leq R_1 \leq R_2 \leq \dots \leq R_{n-2} \leq R_{n-1} \text{ (vincolo monotonicità)} \\ R_j - R_{j-1} = k_j \cdot (j+1), k_j \geq 0, j = 1, 2, \dots, n-1 \text{ (vincolo divisibilità)} \end{array} \right. \quad (3.5)$$

Per questo problema verrà trovata una soluzione tenendo in considerazione solamente il vincolo di dimensione totale e di monotonicità. Infine, per soddisfare il vincolo di divisibilità verrà usato un metodo euristico che non comporterà un' eccessiva perdita di qualità [21].

Il problema 3.5 può essere affrontato utilizzando la tecnica dei Moltiplicatori di Lagrange considerando la funzione Lagrangiana di Eq. (3.6).

$$L(R_0, R_1, \dots, R_{n-1}, \lambda) = q_0 \cdot \sigma_x^2 + \sum_{j=0}^{n-1} q_{j+1} \cdot D(R_j) + \lambda \left(\sum_{j=0}^{n-1} \alpha_j R_j - R_t \right) \quad (3.6)$$

Derivando tale funzione sia rispetto a R_j con $j = 0, 1, 2, \dots, n-1$, che rispetto a λ (Moltiplicatore Lagrangiano), si ottiene il sistema completo di Eq. (3.7).

$$\left\{ \begin{array}{l} \frac{q_i}{\alpha_i} \cdot \frac{\partial D(R_i)}{\partial R_i} + \lambda = 0 \text{ con } i = 0, 1, 2, \dots, n-1 \\ \sum_{j=0}^{n-1} \alpha_j R_j = R_t \end{array} \right. \quad (3.7)$$

A questo punto si può già intuire una possibile soluzione tale che R_j soddisfi la condizione di Eq. (3.8).

$$\frac{\partial D(R_j)}{\partial R_j} = -\lambda \cdot \frac{\alpha_j}{q_j} \quad (3.8)$$

Ricordando inoltre che la funzione Distortion Rate $D(R)$ è una funzione convessa e monotona decrescente, se $\frac{\alpha_j}{q_j}$ fosse monotona decrescente, il vincolo di monotonicità verrebbe di conseguenza rispettato. Questo non può essere assunto vero a priori, in

quanto q_j dipendendo dal canale è del tutto aleatorio. Per ovviare a questo problema viene utilizzato il seguente Teorema:

Teorema 3.3.1. Se $\frac{\alpha_j}{q_j} \leq \frac{\alpha_{j+1}}{q_{j+1}}$ allora nella soluzione ottima $R_j = R_{j+1}$.

Dimostrazione. La dimostrazione verrà affrontata per assurdo. Si assume per assurdo che esista una soluzione tale che $R_n < R_{n+1}$.

Possiamo togliere ΔR bit da R_{n+1} e aggiungere $\Delta R \cdot \frac{\alpha_{n+1}}{\alpha_n}$ bit a R_n , senza che l'ipotesi fatta in precedenza venga negata. In tal modo è possibile calcolare la differenza di distorsione tra la distorsione di partenza e quella con in nuovi R_n e R_{n+1} appena calcolati, dove con $\lambda(R_n)$ indichiamo il modulo della pendenza di D nel punto R_n (Eq. (3.9)).

$$\begin{aligned} \Delta D &= q_n D(R_n) + q_{n+1} D(R_{n+1}) - \\ & [q_n (D(R_n) - \Delta R \frac{\alpha_{n+1}}{\alpha_n} \lambda(R_n)) + q_{n+1} (D(R_{n+1}) + \Delta R \lambda(R_{n+1}))] = \\ & \Delta R_{n+1} (\frac{q_n}{\alpha_n} \lambda(R_n) - \frac{q_{n+1}}{\alpha_{n+1}} \lambda(R_{n+1})) \end{aligned} \quad (3.9)$$

Poichè dalle ipotesi fatte $\frac{\alpha_n}{q_n} \leq \frac{\alpha_{n+1}}{q_{n+1}}$ e per le proprietà di $D(R)$ (monotonicità e convessità) $\lambda(R_n) > \lambda(R_{n+1})$, avremo $\Delta D > 0$. Non solo, ma potremmo togliere ΔR bit a R_{n+1} e aggiungerne $\Delta R \cdot \frac{\alpha_{n+1}}{\alpha_n}$ a R_n ottenendo una miglior soluzione, in contrasto con quanto affermato dall'ipotesi per assurdo. Di conseguenza la soluzione ottima sarà data da $R_n = R_{n+1}$ c.v.d. □

Osservazione 3.3.1.

- Nel caso in cui $\frac{\alpha_n}{q_n} \leq \frac{\alpha_{n+1}}{q_{n+1}}$, secondo il Teorema appena visto, il problema non cambia se si considera $R'_n = R_n = R_{n+1}$, tale per cui $\alpha'_n = \alpha_n + \alpha_{n+1}$ e $q'_n = q_n + q_{n+1}$.
- Si osserva che se $\frac{\alpha_n}{q_n} \leq \frac{\alpha_{n+1}}{q_{n+1}}$ allora $\frac{\alpha_n}{q_n} \leq \frac{\alpha'_n}{q'_n} = \frac{\alpha_n + \alpha_{n+1}}{q_n + q_{n+1}} \leq \frac{\alpha_{n+1}}{q_{n+1}}$. Questo può essere sfruttato osservando il profilo di $\frac{\alpha_n}{q_n}$ e sostituendo le parti piatte o decrescenti con i relativi valori equivalenti, riducendo la complessità del problema (vedi Fig. 3.8).

In Fig. 3.9 si può osservare come l'algoritmo riduce la dimensione del problema, partendo dal profilo di $\frac{\alpha_n}{q_n}$.

Nonostante il vantaggio di ridurre la dimensione del problema, al termine dell'algoritmo si potrebbero avere meno layer di quelli desiderati in fase di progetto. Questo viene ovviato riducendo il valore di R_j tale che $R_j - R_{j+1}$ sia un multiplo intero di $j + 1$ per ogni valore del layer j -esimo. Tale approccio, anche se euristico ottiene dei risultati prossimi alla soluzione ottimale [21]. Oltre all'algoritmo appena presentato, in letteratura sono presenti diversi approcci [28], [29].

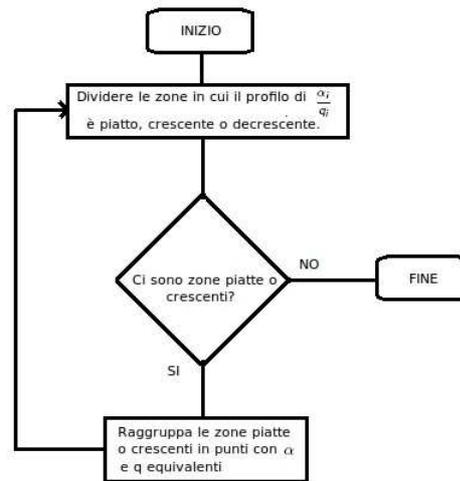


Figura 3.8. Flow chart dell'algoritmo di allocazione

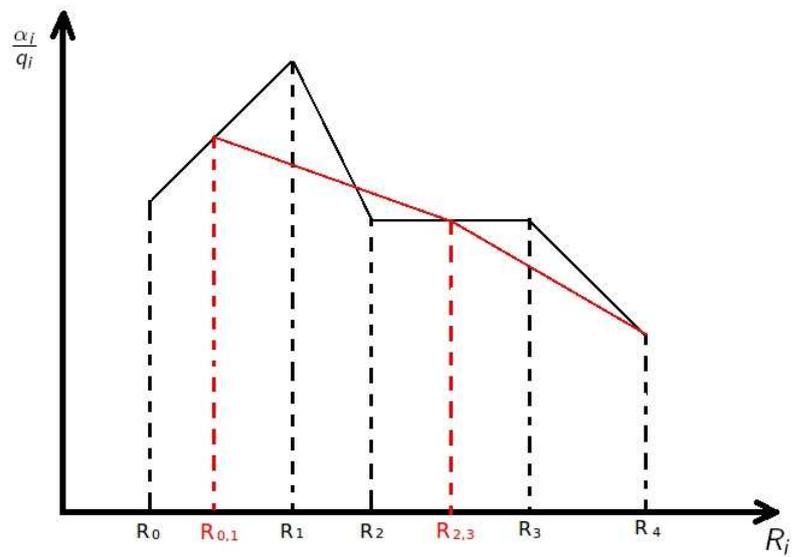


Figura 3.9. Esempio di allocazione mediante l'algoritmo

Capitolo 4

Teoria dei Giochi

4.1 Introduzione

La Ricerca Operativa classica suppone che le decisioni relative ad un problema siano prese in modo centralizzato da un unico decisore che può operare in totale autonomia e con completa libertà. La Teoria dei Giochi cerca di trovare dei possibili punti di equilibrio corrispondenti a soluzioni subottime, quando le scelte non vengono effettuate da una una unica unità di decisione ma da più unità in competizione. Tali unità detti *giocatori*, operano perseguendo obiettivi che possono risultare comuni, ma non identici, differenti ed eventualmente contrastanti; possono essere presenti anche aspetti aleatori.

A seguire, viene riportato un esempio classico di Teoria dei Giochi.

Esempio 4.1.1 (Dilemma del prigioniero). *Due individui I e II sono stati arrestati per lo stesso reato e vengono interrogati separatamente dal giudice; ognuno può scegliere indipendentemente dall'altro di confessare (C) o non confessare (NC). Se entrambi non confessano vengono condannati per reati minori a due anni ciascuno; se entrambi confessano vengono condannati a cinque anni ciascuno; se uno confessa e l'altro no, quello che confessa ha uno sconto di pena e viene condannato a un anno, mentre l'altro ha un'aggravante e viene condannato a sei anni. Le pene sono riportate in tabella come coppia (I,II).*

I/II	C	NC
C	-5,-5	-1,-6
NC	-6,-1	-2,-2

Ragionevolmente I sceglie C poichè consegue una condanna minore qualunque

sia la scelta di II ($-5 > -6$; $-1 > -2$) e analogamente II sceglie C. La decisione attesa è quindi (C, C), con una condanna a 5 anni per ciascuno (equilibrio in strategie dominanti), mentre per entrambi sarebbe vantaggiosa la scelta (NC,NC) con 2 anni per ciascuno.

Esempio 4.1.2 (Battaglia dei sessi). Due fidanzati devono scegliere tra andare al teatro (T) o alla partita (P). Lei (I) preferisce il teatro mentre lui (II) preferisce la partita, ma entrambi non hanno interesse a restare da soli, come riportato nella tabella. In questo caso non esiste una strategia dominante per nessun giocatore (ad esempio per I scegliendo T si ha che $2 > 0$ ma $0 < 1$).

I/II	T	P
T	2,1	0,0
P	0,0	1,2

Nell'esempio 4.1.2 una telefonata per conoscere la decisione dell'altro giocatore, o un accordo al 50%, o una strategia correlata possono risolvere facilmente il problema. Nell'Esempio 4.1.1 invece questo non succede in quanto la possibilità di comunicare renderebbe probabile un accordo per la strategia NC, ma al momento della decisione sia I che II risceglierebbero C, poichè $-1 > -2$.

Secondo la classificazione di Harsanyi [30] si distinguono due classi di giochi:

- giochi non cooperativi. Non sono possibili accordi vincolati tra i giocatori;
- giochi cooperativi. Sono possibili accordi vincolati tra i giocatori.

Osservazione 4.1.1.

- Nelle convenzioni comuni sulla Teoria dei Giochi in un gioco non cooperativo i giocatori non possano nemmeno comunicare in quanto ciò potrebbe alterare il risultato.
- I giochi cooperativi sono distinti in due sottoclassi: giochi a utilità non trasferibile (NTU) o senza pagamenti laterali, e giochi a utilità trasferibile (TU) o a pagamenti laterali, che costituiscono un caso particolare dei giochi NTU.

4.2 Rappresentazione di un gioco

Un gioco può essere presentato in tre diverse forme: la forma estesa, la forma strategica definita anche forma normale e la forma caratteristica usata per i giochi cooperativi.

Osservazione 4.2.1.

- Si chiama *funzione dei pagamenti (payoff)* una funzione f che assegna ad ogni giocatore la sua vincita per ogni possibile terminazione del gioco.
- Si chiama *strategia del giocatore* i una funzione σ_i che assegna al giocatore i una mossa per ogni possibile situazione del gioco.

4.2.1 Forma estesa

È una descrizione completa e puntuale del gioco, delle mosse e delle relative probabilità, delle funzioni dei pagamenti, delle strategie, degli insiemi di informazione ecc.; risulta molto informativa ma poco maneggevole.

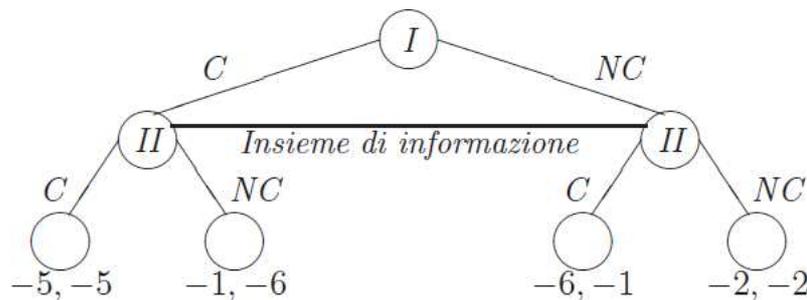


Figura 4.1. Forma estesa del dilemma del prigioniero

Per descrivere il gioco è possibile utilizzare un albero (Fig. 4.1). Ogni ramo rappresenta una possibile mossa del giocatore e ad ogni nodo è associata una possibile situazione del gioco. Infine ai nodi terminali si associano i valori delle vincite *payoff* di ciascun giocatore.

In generale la forma estesa contiene più informazioni sul gioco rispetto alla forma strategica (vedi oltre), che risulta comunque sufficiente a rappresentare il gioco. Nella Fig. 4.1 si nota il collegamento *insieme di informazione* che sta ad indicare che il giocatore II compie la propria giocata successivamente alla giocata del giocatore I, nel caso i due giocatori giocassero contemporaneamente il collegamento viene rimosso.

4.2.2 Forma strategica

Una rappresentazione più efficiente è data dalla *rappresentazione in forma strategica*.

Se ci sono n giocatori si utilizza una $2n$ -upla $(\Sigma_1, \Sigma_2, \dots, \Sigma_n, f_1, f_2, \dots, f_n)$ dove:

- $\Sigma_1, \Sigma_2, \dots, \Sigma_n$ sono insiemi non vuoti contenenti le possibili strategie o scelte di ogni giocatore.
- f_1, f_2, \dots, f_n sono funzioni reali definite sul prodotto cartesiani degli insiemi Σ_i , cioè:

$$f_i : \prod_{k=1, \dots, n} \Sigma_k \longrightarrow \mathbb{R} \text{ con } i = 1, \dots, n \quad (4.1)$$

Una possibile interpretazione è che tutti i giocatori scelgono contemporaneamente la strategia e la funzione dei pagamenti f_i dice qual'è il guadagno del giocatore i determinato dalle scelte fatte.

Definizione 4.2.1. Una n -upla $(\sigma_1, \sigma_2, \dots, \sigma_n) \in \prod_{k=1, \dots, n} \Sigma_k$ è detta *profilo di strategie*

Da quanto detto a proposito di strategie nella forma estesa, si può intuire la possibilità di passare dalle forma estesa a quella strategica, tuttavia il passaggio inverso risulta più complesso. Gli elementi della forma strategica possono essere riassunti anche in una tabella come nel caso dell' Esempio 4.1.2 o 4.1.1. Nel caso del dilemma del prigioniero le informazioni fornite dalla rappresentazione in forma strategica sono quelle dell'Esempio 4.2.1.

Esempio 4.2.1 (Rappresentazione in forma strategica del dilemma del prigioniero).

$$\Sigma_I = \{C, NC\}; \Sigma_{II} = \{C, NC\}$$

$$f_I(C, C) = -5; f_I(C, NC) = -1; f_I(NC, C) = -6; f_I(NC, NC) = -2$$

$$f_{II}(C, C) = -5; f_{II}(C, NC) = -6; f_{II}(NC, C) = -1; f_{II}(NC, NC) = -2$$

4.2.3 Forma caratteristica

Questa forma può essere usata solo per i giochi cooperativi in quanto fa riferimento alla nozione di *coalizione*.

Definizione 4.2.2.

- Detto N l'insieme dei giocatori, ogni sottoinsieme S di N è detto coalizione. Se $S=N$ si ha la grande coalizione.
- Si dice funzione caratteristica di un gioco ad n giocatori una funzione indicata con v (se il gioco è senza pagamenti laterali si usa V ed è più complessa) per cui si ha:

$$v : \wp(N) \longrightarrow \mathbb{R} \text{ con } v(\emptyset) = 0 \quad (4.2)$$

dove $\wp(N)$ rappresenta l'insieme delle parti dell'insieme N .

- Se per ogni coppia di coalizioni disgiunte S e T si ha $v(S \cup T) = v(S) + v(T)$ la funzione v è detta additiva; se si ha $v(S \cup T) \geq v(S) + v(T)$ la funzione v è detta superadditiva; se si ha $v(S \cup T) \leq v(S) + v(T)$ la funzione v è detta subadditiva.

In altre parole v assegna ad S la massima vincita possibile indipendentemente dal comportamento degli altri giocatori. Inoltre, se la funzione caratteristica è additiva o superadditiva o subadditiva anche il gioco è detto *additivo* o *superadditivo* o *subadditivo*. Se per la coalizione S si ha $v(S) = v(N \setminus S)$ il gioco è detto *a somma costante*. Nel caso dell'esempio del dilemma del prigioniero, una rappresentazione in forma caratteristica si presenta come nell'Esempio 4.2.2.

Esempio 4.2.2 (Rappresentazione in forma caratteristica del dilemma del prigioniero).

$$N = \{I, II\}$$

$$v(\emptyset) = 0; v(I) = v(II) = -5; v(I, II) = -4$$

4.3 Razionalità e utilità

Come già accennato, la Teoria dei Giochi è fondata su alcune ipotesi che caratterizzano il modo fondamentale di agire (e di pensare) degli individui.

I giocatori cercano di adottare una strategia che permetta di raggiungere l'esito finale a loro più favorevole, cioè massimizzare la propria utilità. Al fine di identificare quale risultato si vuole perseguire è necessario prendere in considerazione valori differenti: economico, sentimentale, sociale, ecc.

A tal proposito la funzione utilità dei payoff è stata generalizzata con il concetto di “preferibilità”.

Definizione 4.3.1.

- *Dati due eventi A e B si dice che A è preferibile a B per un giocatore se egli cerca di conseguire A invece di B e si indica con $A \succ B$.*
- *Dati due eventi A e B si dice che A è indifferente a B per un giocatore se nessuno è preferibile all'altro e si indica con $A \equiv B$.*

Assiomi 4.1.

A1 *Dati due eventi A e B allora $A \succ B$ oppure $B \succ A$ oppure $A \equiv B$.*

A2 $A \equiv A$.

A3 $A \equiv B \Rightarrow B \equiv A$.

A4 $A \equiv B, B \equiv C \Rightarrow A \equiv C$.

A5 $A \succ B, B \succ C \Rightarrow A \succ C$.

A6 $A \succ B, B \equiv C \Rightarrow A \succ C$.

A7 $A \equiv B, B \succ C \Rightarrow A \succ C$.

4.3.1 Lotterie

Gli eventi possono essere certi oppure incerti secondo una probabilità nota; tale situazione viene rappresentata tramite il concetto di lotteria.

Definizione 4.3.2. *Dati due eventi A e B si chiama lotteria l'evento $rA + (1-r)B$, $0 \leq r \leq 1$ in cui A si verifica con probabilità r e l'evento B con probabilità $1-r$.*

Osservazione 4.3.1.

La lotteria non è una combinazione lineare di eventi, in quanto il risultato può essere solo A o B e non qualcosa di intermedio; la lotteria permette di valutare l'evento “esce A o esce B ”.

Proprietà 4.1.

$$\mathbf{P1} \quad A \equiv C \Rightarrow \{rA + (1-r)B\} \equiv \{rC + (1-r)B\} \quad \forall r, \forall B$$

$$\mathbf{P2} \quad A \succ C \Rightarrow \{rA + (1-r)B\} \succ \{rC + (1-r)B\} \quad r > 0, \forall B$$

$$\mathbf{P3} \quad A \succ C \succ B \Rightarrow \exists! r, 0 < r < 1 \text{ t.c. } \{rA + (1-r)B\} \equiv C$$

Osservazione 4.3.2.

Se un decisore soddisfa gli assiomi **A1 - A7** e le proprietà **P1 - P3** viene considerato “razionale”.

Esempio 4.3.1 (Preferenze). Siano date le lotterie:

$$E_1 = \{0, 100 \text{ con } \mathbb{P}(0) = 1/2, \mathbb{P}(100) = 1/2\}$$

$$E_2 = \{40, 60 \text{ con } \mathbb{P}(40) = 3/4, \mathbb{P}(60) = 1/4\}$$

$$E_3 = \{0, 100, 40, 60 \text{ con } \mathbb{P}(0) = 1/4, \mathbb{P}(100) = 1/4, \mathbb{P}(40) = 3/8, \mathbb{P}(60) = 1/8\}$$

Il guadagno atteso di E_1 è 50, quello di E_2 è 45 e quello di E_3 è 47.5, ma questo non impone una preferenza tra i tre eventi, nel senso che E_1 permette guadagni maggiori, E_2 garantisce rischi minori ed E_3 è intermedio; le uniche relazioni da soddisfare sono:

$$E_1 \equiv E_2 \Rightarrow E_1 \equiv E_3, E_2 \equiv E_3$$

oppure

$$E_1 \succ E_2 \Rightarrow E_1 \succ E_3, E_3 \succ E_2$$

oppure

$$E_2 \succ E_1 \Rightarrow E_2 \succ E_3, E_3 \succ E_1$$

Dato un insieme di eventi E , una relazione di preferenza su E può essere rappresentata come una funzione utilità $u : E \rightarrow \mathbb{R}$ (detta anche utilità di von Neumann-Morgenstern) tale che per ogni coppia di eventi $E_1, E_2 \in E$ si ha:

$$E_1 \succ E_2 \Leftrightarrow u(E_1) > u(E_2)$$

$$u(rE_1 + (1-r)E_2) = ru(E_1) + (1-r)u(E_2)$$

Osservazione 4.3.3.

- *La funzione di utilità permette di quantificare le preferenze.*
- *L'utilità di von Neumann-Morgenstern impone la linearità sulle lotterie.*

La funzione u è unica a meno di trasformazioni affini, cioè u è una funzione di utilità se e solo se lo è anche:

$$\hat{u} = \alpha u + \beta \text{ con } \alpha > 0 \quad (4.3)$$

4.4 Concetto di “soluzione” del gioco

Introdotta il concetto di gioco e come lo si può rappresentare, è necessario specificare cosa vuol dire risolverlo o determinare una soluzione.

Poichè le variabili decisionali non dipendono da un unico giocatore, non è possibile dare una soluzione nel senso della Ricerca Operativa classica; risolvere un gioco consiste nel fornire delle indicazioni ad uno o più giocatori, sulle strategie da adottare se il gioco ad esempio è non cooperativo. Tali indicazioni non possono essere assolute in quanto bisogna tener conto di altri fattori (alcuni aleatori, altri legati a preferenze e sensazioni del singolo giocatore).

In altre parole il termine “concetto di soluzione” indica quella che secondo alcuni criteri assoluti è una scelta che può risultare accettabile a tutti i giocatori secondo i loro criteri soggettivi.

4.4.1 Equilibrio di Nash

Il più semplice e importante concetto di soluzione per un gioco non cooperativo lo si deve a John Nash il quale ha dimostrato che sotto opportune ipotesi esiste sempre una condizione di equilibrio. In questa condizione di equilibrio un dato giocatore sceglie la sua mossa strategica in modo da massimizzare la sua funzione utilità fermo restando che il comportamento degli altri giocatori non varierà a motivo della sua scelta.

Definizione 4.4.1. *Dato un gioco G si dice che la n -upla di strategie $(\sigma_1^*, \sigma_2^*, \dots, \sigma_n^*)$ con $\sigma_i^* \in \Sigma_i$ costituisce un equilibrio, o è in equilibrio, se nessun giocatore ha interesse ad essere l'unico che cambia strategia, cioè se:*

$$f_i(\sigma_1^*, \dots, \sigma_i^*, \dots, \sigma_n^*) \geq f_i(\sigma_1^*, \dots, \sigma_i, \dots, \sigma_n^*), \forall \sigma_i \in \Sigma_i, \forall i \in N$$

Infatti, come si può desumere dalla disuguaglianza, se il giocatore i sceglie una qualunque strategia a sua disposizione diversa da σ_i^* mentre tutti gli altri hanno scelto la propria strategia $(\sigma_1^*, \dots, \sigma_{i-1}^*, \sigma_{i+1}^*, \dots, \sigma_n^*)$, può solo peggiorare il proprio guadagno o, al più, lasciarlo invariato. Se ne deduce quindi che se i giocatori raggiungono un equilibrio di Nash, nessuno può migliorare il proprio risultato modificando solo la propria strategia, ed è quindi vincolato alle scelte degli altri.

Ovviamente possono esistere differenti strategie per uno o più giocatori a cui corrispondono payoff migliori, come nel caso del dilemma del prigioniero in cui l'equilibrio (C, C) risulta inefficiente e il risultato più ovvio per un "supervisore" non è di equilibrio.

Poichè questo vale per tutti i giocatori, è evidente che se esiste un equilibrio di Nash ed è unico, esso rappresenta la soluzione del gioco, in quanto nessuno dei giocatori ha interesse a cambiare strategia.

4.5 Giochi a somma zero

Verrà ora presentata una classe particolare di giochi dalla quale sono stati sviluppati e estesi alcuni Teoremi fondamentali sull'esistenza degli equilibri di Nash.

Definizione 4.5.1.

Un gioco G si dice a somma zero se per ogni terminazione del gioco la somma dei payoff è nulla.

In altre parole tutto quello che viene guadagnato da qualche giocatore viene perso da qualche altro giocatore.

Nel caso più semplice a due giocatori la matrice dei pagamenti può essere espressa indicando la vincita, positiva o negativa, del primo giocatore poichè la vincita del secondo è in ogni caso l'opposto. Si può utilizzare una matrice A in cui la riga i è associata alla strategia σ_i del giocatore I, la colonna j alla strategia σ_j del giocatore II e l'elemento a_{ij} rappresenta quanto il primo giocatore riceve se giocano la coppia di strategia (σ_i, σ_j) .

4.5.1 Gioco a due giocatori in forma normale

Se l'elemento a_{ij} è il più piccolo della riga i e il più grande della colonna j , allora siamo in presenza di un equilibrio di Nash dovuto alle strategie $\sigma_i \in \Sigma_i$ e $\sigma_j \in \Sigma_j$. L'equilibrio viene anche detto *punto di sella*.

L'obiettivo dei giocatori è in ogni caso di massimizzare il proprio *payoff*, cioè massimizzare la vincita o minimizzare la perdita; se i giocatori scelgono σ_i e σ_j , con vincita a_{ij} , il giocatore I cerca di massimizzarlo intervenendo solo sulla scelta di σ_i e il giocatore II cerca di minimizzarlo intervenendo solo sulla scelta di σ_j . La non conoscenza della strategia scelta dall'altro giocatore impedisce di poter raggiungere con certezza l'obiettivo, ma l'esistenza di un punto di sella fa sì che ragionevolmente entrambi scelgano quella coppia di strategie.

Esempio 4.5.1 (Punto di sella).

Nel gioco in forma normale rappresentato nel seguito, la strategia (σ_1, σ_2) costituisce un equilibrio anche se entrambi i giocatori hanno a disposizione dei payoff migliori.

$$\begin{pmatrix} 6 & 3 & 4 \\ 5 & 2 & -2 \\ 7 & -1 & 3 \end{pmatrix} \quad (4.4)$$

L'esistenza del punto di sella o equilibrio di Nash non impone ai giocatori la scelta delle corrispondenti strategie; in questo esempio il giocatore I sa di poter ottenere payoff superiori a 3 unità e il giocatore II sa di poter pagare payoff inferiori a 3 unità; d'altra parte il giocatore I sa che scegliendo la terza strategia può ottenere 7 unità ma può pagarne 1 se il giocatore II sceglie la seconda strategia e il giocatore II sa che scegliendo la terza strategia può ottenere 2 unità ma può pagarne 4 se il giocatore I sceglie la prima strategia.

Teorema 4.5.1. *In un gioco a due persone a somma zero se (σ_i, σ_j) e (σ_h, σ_k) sono equilibri di Nash, allora lo sono anche (σ_i, σ_k) e (σ_h, σ_j) .*

Esempio 4.5.2. *Nel gioco in forma normale rappresentato di seguito, le strategie (σ_1, σ_2) e (σ_2, σ_3) sono equilibri di Nash allora lo sono anche (σ_1, σ_3) e (σ_2, σ_2) .*

$$\begin{pmatrix} 2 & 1 & 1 & 4 \\ 5 & 1 & 1 & 2 \\ 3 & -1 & 0 & -1 \end{pmatrix} \quad (4.5)$$

4.5.2 Gioco a due giocatori a somma zero senza equilibri puri di Nash

La maggior parte dei giochi non presenta alcun punto di sella, come il gioco sottostante:

$$\begin{pmatrix} 4 & 2 \\ 1 & 3 \end{pmatrix} \quad (4.6)$$

in questo caso il primo giocatore con la prima strategia si garantisce una vincita minima di 2 (*gain-floor*) e il secondo giocatore con la seconda strategia si garantisce una perdita massima di 3 (*loss-ceiling*).

La vincita minima per il giocatore I si indica con v'_I e si ha:

$$v'_I = \max_i \min_j \{a_{ij}\} \quad (4.7)$$

La perdita massima per il giocatore II si indica con v'_{II} e si ha:

$$v'_{II} = \min_j \max_i \{a_{ij}\} \quad (4.8)$$

È facile verificare che $v'_I \leq v'_{II}$; se $v'_I = v'_{II}$ allora esiste un punto di sella.

In generale un comportamento “razionale” fa sì che il giocatore I vinca almeno v'_I e il giocatore II perda al più v'_{II} e il risultato è peggiore se l'altro giocatore può sfruttare la “razionalità” e prevedere la mossa. Pertanto per migliorare il risultato è necessario non giocare “razionalmente”, cioè non giocare solo la strategia di vincita minima o di perdita massima.

Esempio 4.5.3 (Pari e dispari modificato). *I due giocatori possono lanciare 1, 2, 3; il giocatore I vince se la somma dei numeri è pari, altrimenti vince il giocatore II. Apparentemente il gioco è favorevole al giocatore I che può vincere in 5 casi su 9.*

I/II	1	2	3
1	P	D	P
2	D	P	D
3	P	D	P

D'altra parte il giocatore II potrebbe giocare 2 e quindi avrebbe 2 risultati vincenti su 3; ma a questo punto il giocatore I giocando 2 è “sicuro” di vincere. Analogamente il giocatore I potrebbe giocare 1 (o 3) per avere 2 risultati vincenti su 3 e a questo punto il giocatore II giocando 2 è “sicuro” di vincere.

Cercare di aumentare le proprie possibilità di vincita porta alla sconfitta “sicura”.

Osservazione 4.5.1.

Si possono evidenziare alcuni limiti dell'equilibrio di Nash (in strategie pure):

- inefficienze (*dilemma del prigioniero*);
- non unicità (*battaglia dei sessi*);

- non esistenza (pari e dispari).

4.6 Strategie miste

Nei casi precedentemente esposti, sono state valutate le conseguenze di giocate deterministiche delle strategie disponibili. Da come si è visto, in tali situazioni può accadere che non esista una condizione di equilibrio. Tuttavia, inserendo una componente aleatoria sulla giocata della singola strategia, si vedrà come tale condizione permetta l'esistenza di un equilibrio di Nash in ogni gioco finito.

Definizione 4.6.1. Si chiama *strategia mista* per un giocatore una distribuzione di probabilità sull'insieme delle sue strategie pure.

Nell'Esempio 4.5.3 il giocatore II che parte svantaggiato può riequilibrare le sue possibilità giocando a caso, con probabilità 0.5 sia 1 che 2 (o altre strategie equivalenti); in questo modo il giocatore I non può in alcun modo avere maggiori probabilità di vincere.

Supponiamo ora che l'insieme delle strategie pure sia formato da n elementi, allora una strategia mista si può indicare con un vettore $x = (x_1, x_2, \dots, x_n)$ con $x_i \geq 0$ e $\sum_{i=1, \dots, n} x_i = 1$.

L'insieme delle strategie miste del giocatore I si indica con X e l'insieme delle strategie miste del giocatore II si indica con Y .

Definizione 4.6.2. Dato un gioco G a due giocatori a somma zero in forma normale con matrice A è detta *vincita attesa* se il giocatore I gioca la strategia mista $x \in X$ e il giocatore II gioca la strategia mista $y \in Y$:

$$V(x, y) = \sum_{i=1, \dots, n} \sum_{j=1, \dots, m} x_i a_{ij} y_j = x^T A y \quad (4.9)$$

È possibile definire la vincita minima per il giocatore I se sceglie la strategia mista $x \in X$ come:

$$v(x) = \min_{y \in Y} \{x^T A y\} = \min_j \{x^T A_{.j}\} \quad (4.10)$$

e la perdita massima per il giocatore II se sceglie la strategia mista $y \in Y$ come:

$$v(y) = \max_{x \in X} \{x^T A y\} = \max_i \{A_{i.} y\} \quad (4.11)$$

dove $A_{.j}$ e A_i sono la colonna j e la riga i di A e le seconde uguaglianze derivano dal fatto che il minimo e il massimo cercati si ottengono con strategie pure.

L'obiettivo del giocatore I è massimizzare $v(x)$ ottenendo la quantità:

$$v_I = \max_{x \in X} \min_j \{x^T A_{.j}\} \quad (4.12)$$

e quello del giocatore II è minimizzare $v(y)$ ottenendo la quantità:

$$v_{II} = \min_{y \in Y} \max_i \{A_i \cdot y\} \quad (4.13)$$

Definizione 4.6.3. *La strategia mista x che permette al giocatore I di ottenere v_I è detta *maxmin*; la strategia mista y che permette al giocatore II di ottenere v_{II} è detta *minimax*. v_I e v_{II} sono detti *valore del gioco* per i giocatori I e II, rispettivamente.*

Teorema 4.6.1 (Teorema del minmax).

Dato un gioco finito a somma zero, si ha:

$$v_I = v_{II} \quad (4.14)$$

Il Teorema 4.6.1 fu dimostrato da von Neumann [31] e assicura l'esistenza di un equilibrio misto in un gioco finito (cioè con numero di strategie finito) a somma zero. Tale risultato fu esteso da Nash, senza più richiedere la condizione che il gioco sia a somma zero [32].

Teorema 4.6.2. *Qualsiasi gioco finito ammette almeno un equilibrio di Nash in strategie miste.*

4.6.1 Calcolo equilibrio di Nash in strategie miste

Considerando l'Esempio 4.1.2, se il giocatore I gioca la strategia mista $(p, 1 - p)$ e il giocatore II gioca la strategia mista $(q, 1 - q)$ la vincita attesa del giocatore I è data da:

$$v_I(p) = 2pq + 0(1-p)q + 0p(1-q) + 1(1-p)(1-q) = 3pq - p - q + 1 = (3q-1)p - (q-1)$$

Il secondo termine non dipende da p , cioè dal giocatore I; si hanno quindi tre casi:

$$3q - 1 > 0 \Rightarrow p = 1(\text{strategia pura})$$

$$3q - 1 = 0 \Rightarrow p(\text{qualsiasi})$$

$$3q - 1 < 0 \Rightarrow p = 0(\text{strategia pura})$$

Analogamente la vincita del giocatore II è data da:

$$v_{II}(q) = 1pq + 0(1-p)q + 0p(1-q) + 2(1-p)(1-q) = 3pq - 2p - 2q + 2 = (3p-2)q - 2(p-1)$$

a cui corrispondono i tre casi:

$$3p - 2 > 0 \Rightarrow q = 1(\text{strategia pura})$$

$$3p - 2 = 0 \Rightarrow q(\text{qualsiasi})$$

$$3p - 2 < 0 \Rightarrow q = 0(\text{strategia pura})$$

Quindi, oltre agli equilibri di Nash in strategie pure (T, T) e (P, P) , si ha un equilibrio in strategie miste se:

$$3q - 1 = 0 \Rightarrow q = \frac{1}{3}$$

$$3p - 2 = 0 \Rightarrow p = \frac{2}{3}$$

cioè $\left(\left(\frac{2}{3}, \frac{1}{3} \right), \left(\frac{1}{3}, \frac{2}{3} \right) \right)$

Osservazione 4.6.1.

La vincita attesa $v_I = v_{II} = \frac{2}{3}$, quindi inferiore alla vincita minima derivante da una strategia pura (cioè 3).

Capitolo 5

Approccio adottato nella realizzazione

5.1 Introduzione

Le tipologie di reti CDN e P2P adottate per la trasmissione video, hanno determinato negli ultimi tempi un notevole incremento nella qualità del servizio offerto. Tuttavia, oltre alla tipologia di rete impiegata, un ulteriore progresso relativo al QoS è stato conseguito con l'introduzione di nuove tecniche di codifica scalabile e MDC.

La codifica MDC supporta la trasmissione della sequenza video attraverso stream indipendenti che permettano di variare in maniera progressiva la qualità della sequenza ricevuta in assenza di dipendenze in decodifica tra i vari stream (tipiche della codifica scalabile).

Trattandosi di trasmissioni che non presentano solamente una singola sorgente (caso P2P o CDN), la trasmissione contemporanea e indipendente tra le varie sorgenti causerebbe inevitabili congestioni che renderebbero il servizio (specialmente se real time) inaccettabile. Al fine di evitare congestioni, sono state definite delle soluzioni di accesso alla rete (vedi DiffServ srTCM [33] o trTCM [34]) che permettono di regolare e prevenire le congestioni, oltre a controllare la QoS del servizio richiesto.

Il lavoro di tesi consiste nell'ottimizzazione della trasmissione video su reti CDN e P2P, utilizzando una tecnica di codifica MDC basata sul codificatore scalabile H.264/SVC. Successivamente, creando diverse tipologie di rete simulate mediante NS2, è stata applicata la Teoria dei Giochi tra i diversi peer (ognuno con una o più descrizioni dello stesso frame) in modo da pre-assegnare delle priorità e utilizzarle in un sistema srTCM secondo quanto definito dallo standard [33].

5.2 Da bitstream H.264/SVC a MDC

Lo schema del codificatore MDC è basato sulla soluzione fornita da [21] dove a partire da un qualsiasi flusso scalabile, utilizzando codici FEC, vengono create un numero arbitrario di descrizioni. Questo ci consente sia di creare un numero arbitrario di “giocatori” (intesi come descrizioni) per avere un “gioco” più complesso su cui compiere i vari test, sia per ridurre la dimensione delle varie descrizioni adattandole ai vincoli di banda.

Nel caso generale viene creato un flusso scalabile H.264/SVC composto da l layer: un base layer e $l - 1$ enhancement layer. A tal scopo viene adottata una scalabilità di qualità (del tipo MGS) in modo da limitare la differenza di qualità in caso di mancata decodifica di un enhancement layer (ad esempio il cambiamento repentino di risoluzione, nel caso di scalabilità di risoluzione, o la variazione della sequenza temporale, nel caso di scalabilità temporale). Come sorgente sono state adottate sequenze video con formato CIF (risoluzione spaziale pari a 352x288 pixel) e con frame rate di 30 frame/s. Le sequenze scelte presentano diverse caratteristiche di movimento al fine di testare l'algoritmo su segnali differenti

Dal flusso scalabile disponibile, si possono creare M descrizioni indipendenti utilizzando una codifica UEP mediante codici FEC. Tuttavia è stato necessario modificare opportunamente l'algoritmo presentato in [21] e nel Capitolo 3, a causa dei seguenti vincoli:

- l'algoritmo è stato creato per la sola codifica di immagini statiche e quindi non prevedeva alcuna limitazione nel tempo di elaborazione richiesto. A tal proposito, si ricorda che il calcolo dei rate ottimi (R_i) dipendeva dalla distribuzione di probabilità del packet loss del canale che essendo tempo-variante, si necessitava di eseguire delle stime attraverso sequenze di “training set” che avrebbero richiesto un eccessivo dispendio di tempo;
- prevede la possibilità di partizionare lo stream scalabile arbitrariamente e ottenere a sua volta un flusso decodificabile (mediante l'utilizzo del codificatore Set Partitioning in Hierarchical Trees basato su decomposizione a sottobande applicando la trasformata Wavelet [35]).

Analizzando questi aspetti è stato deciso di modificare e ampliare l'algoritmo considerando le seguenti motivazioni:

- inizialmente si prevedeva di applicare l'algoritmo di allocazione dei layer nella matrice FEC considerando il bitstream relativo a più frame codificati. Tuttavia è

stato riscontrato che questo approccio forniva descrizioni di dimensione eccessiva per la trasmissione real time. Per evitare questo inconveniente è stato deciso di applicare l'algoritmo ad ogni singolo frame;

- sebbene la codifica scalabile basata su MGS fornisca un'alta granularità del bitstream (con NAL di dimensione ridotta), in modo da gestirlo con più facilità, essa non permette di troncare il flusso in ogni suo punto ottenendo un bitstream altrettanto decodificabile. Perciò è stato deciso di tenere i singoli layer (del bitstream scalabile) come unità inscindibili.

Oltre alle modifiche apportate è nata l'esigenza di aver un controllo sulla dimensione della singola descrizione, in modo da adattarla alle condizioni di banda imposte dalla rete. In Fig. 5.1 viene rappresentata la disposizione dei singoli layer (relativi a un frame) nella matrice FEC.

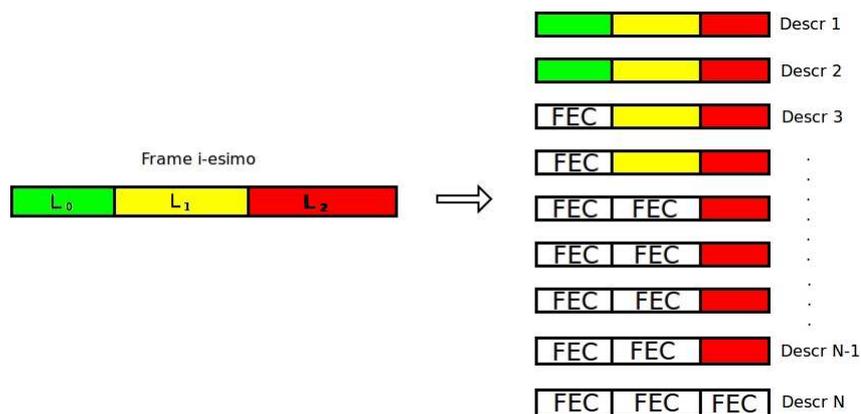


Figura 5.1. Disposizione dei layer all'interno della matrice FEC

Nel caso generale, partendo da un flusso scalabile composto da l layer ($L_0, L_1, L_2, \dots, L_{l-1}$) sarà possibile creare N descrizioni secondo il seguente algoritmo:

1. l'enhancement layer $l - 1$ viene distribuito in $N-1$ blocchi (in modo da avere una capacità correttiva di una sola descrizione persa o errata);
2. l'enhancement layer $l - 2$ viene distribuito in $\lceil \frac{N-1}{2} \rceil$ blocchi, che comporta una capacità correttiva di: $N - \lceil \frac{N-1}{2} \rceil$ descrizioni perse o errate;

3. l'enhancement layer i -esimo o il base layer ($i = 0$) viene distribuito in $\left\lceil \frac{N-1}{2^{l-i-1}} \right\rceil$ blocchi, ottenendo una capacità correttiva di: $N - \left\lceil \frac{N-1}{2^{l-i-1}} \right\rceil$.

In questo modo, aumentando il numero di descrizioni si avrà una diminuzione della dimensione della singola descrizione, in accordo con la possibilità di avere un controllo, anche se non dei più accurati, del rate totale.

Inoltre, adottando l'algoritmo appena presentato si otterrà una capacità correttiva (in termini di numero di pacchetti FEC inseriti) molto alta relativamente ai layer più importanti. Tuttavia, tale protezione decrescerà secondo la relazione precedente con l'aumentare degli enhancement layer. In Fig. 5.2 viene rappresentata la capacità correttiva dei singoli layer partendo da un flusso scalabile composto da $l = 5$ ricavando 12, 19, 22 e 25 descrizioni.

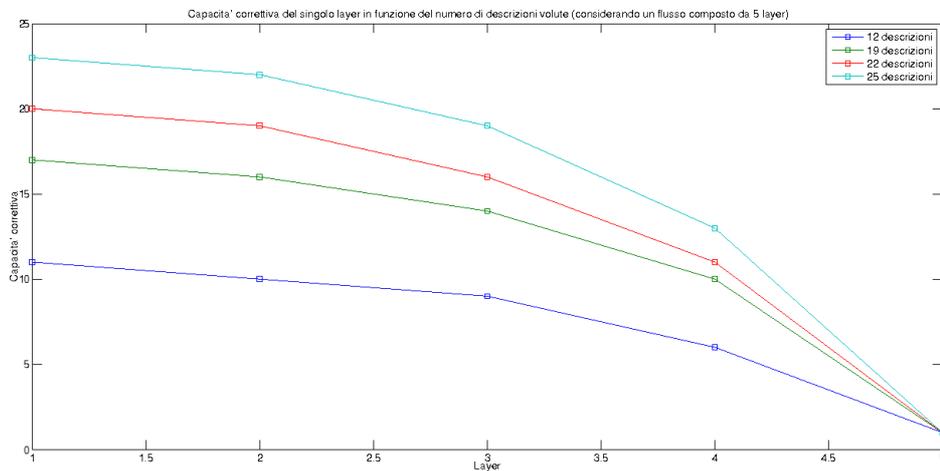


Figura 5.2. Andamento della capacità correttiva in funzione del layer considerato

5.3 Classificazione basata sulla Teoria dei Giochi

Al fine di adattare la QoS alla trasmissione video streaming e tenendo in considerazione tutti i vincoli che tale servizio comporta, i peer o i server possono trasmettere le proprie descrizioni assegnandovi opportune priorità. Quest'ultime, essendo legate al tempo di attesa della descrizione nelle code interne ai nodi di instradamento, determinano il ritardo con cui la descrizione verrà ricevuta a destinazione. Tuttavia, l'assegnazione di queste priorità da parte di un peer sarà fatta indipendentemente dalle scelte degli altri peer, creando in tal caso delle congestioni sulla rete a discapito dell'interesse comu-

ne. Per ovviare a questo inconveniente, una possibile soluzione consiste nell'eseguire un'assegnazione di priorità che tenga conto non solo di quella assegnata dal singolo peer ma anche delle rispettive assegnazioni fatte dagli altri utenti.

L'approccio adottato per ottimizzare l'assegnazione di tali priorità è stato basato sulla Teoria dei Giochi. Innanzitutto si supponga che la rete sia composta da n peer (che d'ora in poi chiameremo "giocatori") e ogni giocatore possieda una o più descrizioni. Inoltre, ogni giocatore può assegnare una sola priorità (che chiameremo "strategia") alla singola descrizione (o all'insieme di descrizioni). La strategia giocata dal giocatore G_d al tempo i , indicata con $c_i^{G_d} \in N_c = \{Green, Yellow, Red\}$ sarà una colorazione della propria descrizione in un sistema DiffServ di tipo srTCM [33]. Inoltre, le strategie giocate da tutti i giocatori nell'istante i verranno indicate mediante il vettore $\mathbf{c}_i = (c_i^{G_0}, c_i^{G_2}, \dots, c_i^{G_{n-1}})$.

Il meccanismo più diffuso di DiffServ per l'assegnazione delle priorità è l'algoritmo srTCM. La sua struttura è rappresentata Fig. 5.3.

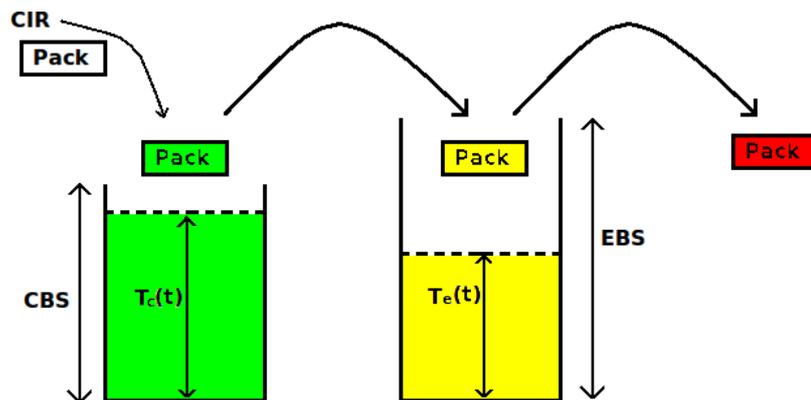


Figura 5.3. Sistema di assegnazione di priorità DiffServ srTCM

Tale meccanismo di assegnazione di priorità (indicate come Green, Yellow, Red) è caratterizzata da un flusso entrante Committed Information Rate (CIR) [pacchetti/s] e due burst size, Committed Burst Size (CBS) [pacchetti] e un Excess Burst Size (EBS) [pacchetti]. Con $T_c(t)$ e $T_e(t)$ si indica lo spazio libero (in pacchetti) all'interno dei relativi buffer e tali variabili vengono inizializzate come $T_c(0) = CBS$, $T_e(0) = EBS$. L'algoritmo di assegnazione delle priorità aggiorna $T_c(t)$ e $T_e(t)$ ogni CIR volte/s secondo la seguente regola:

- se $T_c(t) \leq CBS$ allora $T_c(t) = T_c(t) + 1$, altrimenti;
- se $T_e(t) \leq EBS$ allora $T_e(t) = T_e(t) + 1$, altrimenti;

- né $T_c(t)$ né $T_e(t)$ vengono incrementati.

Se l'algoritmo opera in Color-Blind mode (cioè se i pacchetti non sono stati pre-colorati) allora:

- se $T_c(t) - 1 \geq 0$ allora il pacchetto viene marcato Green e $T_c(t) = T_c(t) - 1$, altrimenti;
- se $T_e(t) - 1 \geq 0$ allora il pacchetto viene marcato Yellow e $T_e(t) = T_e(t) - 1$, altrimenti;
- il pacchetto viene marcato Red e né $T_c(t)$ né $T_e(t)$ vengono decrementati.

Altrimenti, opera in Color-Aware mode (cioè con pacchetti pre-colorati) come:

- se il pacchetto è stato pre-colorato Green e $T_c(t) - 1 \geq 0$ allora il pacchetto viene marcato Green e $T_c(t) = T_c(t) - 1$, altrimenti;
- se il pacchetto è stato pre-colorato Green o Yellow e $T_e(t) - 1 \geq 0$ allora il pacchetto viene marcato Yellow e $T_e(t) = T_e(t) - 1$, altrimenti;
- il pacchetto viene marcato Red e né $T_c(t)$ né $T_e(t)$ vengono decrementati.

Prendendo spunto dalla struttura DiffServ, ogni classe di priorità (Green, Yellow e Red) viene associata ad un relativo buffer con una certa capacità (Green Capacity, Yellow Capacity e Red Capacity) indicata con C_k dove, $C_G = CBS$, $C_Y = EBS$, $C_R \geq \max\{C_Y, C_R\}$. Inoltre, il CIR, indicato nello standard srTCM, viene modellato come un processo di Poisson di parametro λ pacchetti/s.

Successivamente, per gestire i singoli buffer relativi ad ogni classe, si è utilizzato il seguente modello. Con B_k rappresentiamo il buffer relativo alla singola classe ($Green \Rightarrow k = G$, $Yellow \Rightarrow k = Y$, $Red \Rightarrow k = R$) e quindi al tempo i nel buffer k il numero di pacchetti con priorità k sarà rappresentato dall'Eq. (5.1):

$$B_k = \left(\sum_{t \leq i} \sum_{d=0}^{n-1} \chi(c_t^{G_d} = k) \right) - i \cdot R_k \quad (5.1)$$

con $\chi(c_t^{G_d} = k)$ si rappresenta la variabile indicatrice che è uguale a 1 se al tempo t la strategia giocata è stata k e 0 altrimenti. Con R_k si intende il rate con il quale vengono svuotati il buffer B_k , che dipende dal rate di congestione del link.

Pertanto, se il giocatore G_d all'istante i gioca la strategia k , la probabilità che la descrizione con priorità k arrivi fuori un tempo limite (rappresentato con T_L) sarà data dall'Eq. (5.2):

$$P_{i,k}^{late}(B_k) = P\left[\sum_{m=0}^{B_k-1} \Delta T_m > T_L\right] \quad (5.2)$$

dove ΔT_m rappresenta il tempo di servizio del pacchetto m -esimo nel buffer k , approssimato come una variabile aleatoria esponenziale di parametro $\beta = 30 \cdot \frac{N}{n}$ pacchetti/s (si ricorda che le sequenze video analizzate presentano un frame rate di 30 frame/s, inoltre è stato assunto che le N descrizioni per frame vengano equamente divise tra gli n giocatori). Come valore di soglia è stato scelto $T_L = 0.2s$ che risulta un ritardo accettabile per trasmissioni video streaming. Tuttavia si può notare che nel calcolo di tale probabilità (Eq. (5.2)), B_k tiene conto della strategia giocata sia dal giocatore G_d che da tutti gli altri giocatori, fornendo un'informazione globale dell'intero gioco.

A questo punto, conoscendo la n -upla \mathbf{c}_i di strategie giocate da tutti i giocatori al tempo i , è possibile stabilire la probabilità che la descrizione (o l'insieme delle descrizioni) appartenente a G_d , classificata come $c_i^{G_d} = k$ venga persa (o a causa del ritardo nel tempo di attesa o a causa di un'eventuale congestione dei link o politiche di drop) Eq. (5.3):

$$P_{loss,k}(c_i) = \begin{cases} P_{i,k}^{late}(B_k) + (1 - P_{i,k}^{late}(B_k)) \cdot P_{lossChan,k} & \text{se } B_k < C_k \\ 1 & \text{se } B_k \geq C_k \end{cases} \quad (5.3)$$

con $P_{lossChan,k}$ viene indicata la probabilità che la descrizione (classificata come k) venga persa a causa di eventuali congestioni della rete o politiche di drop.

5.3.1 Funzione utilità adottata

Nella Teoria dei Giochi, al fine di quantificare l'utilità che un singolo giocatore (G_d) può trarre in corrispondenza alla giocata di un vettore di strategie \mathbf{c}_i , viene definita una "funzione utilità" $f_{G_d}(\mathbf{c}_i)$:

$$f_{G_d} : N_c^n \longrightarrow \mathbb{R} \quad (5.4)$$

Nell'ambito di questo lavoro di tesi, tale funzione verrà definita per ogni istante i -esimo in corrispondenza della trasmissione di ogni frame.

L'idea di partenza (presa da [22]) è quella di realizzare tale funzione in modo tale

che fornisca un valore della qualità ricevuta pesata opportunamente dalla probabilità che la trasmissione del pacchetto non vada a buon fine (come espresso dall' Eq. (5.3)). La funzione utilità di partenza sarà data dall'Eq. (5.5):

$$f_{G_g}(\mathbf{c}_i) = P_{loss,k}(\mathbf{c}_i) \cdot \text{Quality} \quad (5.5)$$

Per definire tale funzione è quindi necessario trovare un parametro che in qualche modo quantifichi la qualità video ottenuta per ogni frame i -esimo, nelle varie circostanze di perdita.

A tale scopo si è reso necessario utilizzare il parametro di “activity” definito per ogni frame, che in qualche modo fornisca un livello di “importanza” al frame trasmesso. Tenendo in considerazione che H.264/SVC effettua una predizione gerarchica dei frame (a livello temporale) l'activity definita per il frame i -esimo consiste nella correlazione che intercorre tra questo frame e il frame (o i frame nel caso di B-frame) che è stato utilizzato come *reference picture*. In questo modo più “distanti” sono i frame e più l'activity sarà elevata. Con riferimento alla Fig. 5.4 e agli indici che indicano l'ordine di codifica, se il frame 2 viene indicato con x_2 mentre i frame 0 e 1 vengono indicati rispettivamente con x_0 e x_1 , l'activity del frame 2 $A(2)$ sarà espressa dall'Eq.(5.6):

$$A(2) = \frac{PSNR(x_2, x_1) + PSNR(x_2, x_0)}{2} \quad (5.6)$$

dove $PSNR(x_i, x_j)$ verrà definito in seguito nell'Eq. (5.15). Come conseguenza, la perdita di un frame (o un layer del frame considerato) con activity elevata, influirà in modo diverso sulla qualità totale calcolata sull'intero GOP. Nell'esempio di Fig. 5.4 la perdita del frame 3 (o il suo peggioramento di qualità prodotto dalla perdita di uno dei layer) influirà sulla predizione dei frame 4 e 5 comportando un degrado consistente della qualità dell'intero GOP. Mentre, una perdita del frame 7 comporterà una perdita di qualità minima.

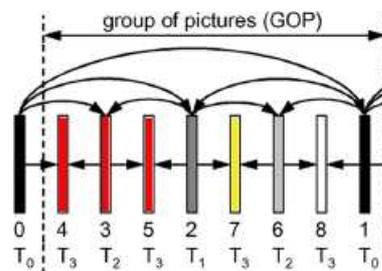


Figura 5.4. Propagazione dell'errore a causa di un frame corrotto

La qualità dell'intero GOP è stata misurata attraverso il calcolo del PSNR (vedi oltre, Eq. (5.15)) per ogni frame e successivamente mediato sull'intera lunghezza del GOP Eq. (5.7):

$$PSNR_{GOP} = \frac{\sum_{k=1}^{L_{GOP}} PSNR(x_k, \widehat{x}_k)}{L_{GOP}} \quad (5.7)$$

dove con L_{GOP} si intende la lunghezza del GOP in esame, mentre \widehat{x}_k rappresenta il frame k -esimo ricostruito in decodifica e x_k il k -esimo frame originale.

Nel caso in cui l'intero frame venisse a mancare è stata utilizzata una politica di error concealment che rimpiazzasse il frame mancante con quello precedente.

Quindi, con riferimento al lavoro di tesi, per ogni sequenza video da trasmettere composta 112 frame (un I-frame più 7 GOP di lunghezza 16), è stata calcolata l'activity di ogni B-frame interno al singolo GOP (come nell'Eq. (5.6)).

Successivamente, è stata eseguita una stima della qualità del frame i -esimo, calcolando il $\delta PSNR_i^L$ sul singolo GOP, associato alla perdite del layer $L = 0, 1, 2, \dots, l-1$ (vedi Eq. (5.8)).

$$\delta PSNR_i^L = \frac{PSNR_{Max} - PSNR_i^L}{PSNR_{Max}} \quad (5.8)$$

Con $PSNR_{Max}$ si fa riferimento al PSNR calcolato tra il file video sorgente e il flusso decodificato privo di perdite (affetto dal solo errore di quantizzazione) sempre mediato sul GOP di riferimento (calcolato mediante Eq. (5.7)). In Fig. 5.5, 5.6, 5.7 vengono rappresentate le distribuzioni di $\delta PSNR^0$, $\delta PSNR^1$, $\delta PSNR^2$ in funzione dell'activity, per le sequenze prese "Foreman", "News", "Mobile", "Crew", in questo caso è stato utilizzato un flusso scalabile composto da tre layer.

I risultati sperimentali mostrano come sia possibile approssimare la relazione tra $\delta PSNR_i^L$ e $A(i)$ attraverso un modello lineare (vedi Eq. (5.9)):

$$\delta PSNR_i^L = m_L \cdot A(i) + q_L, \text{ con } i \in \{\text{B-frame}\} \quad (5.9)$$

dove m_L e q_L sono i parametri calcolati sulla sequenza codificata.

Successivamente, trovato il parametro che desse un valore della qualità ricevuta a seconda dell'importanza del frame trasmesso (e delle relative descrizioni), è stato

pensato di formulare la funzione utilità nel modo espresso dall'Eq. (5.10).

$$f_{G_d}(\mathbf{c}_i) = \sum_{L=0}^{l-1} P[\text{Perdita layer L del frame i-esimo}] \cdot (m_L \cdot A(i) + q_L) \quad (5.10)$$

Per stabilire la $P[\text{Perdita layer L del frame i-esimo}]$ si è ragionato analogamente al seguente esempio: si supponga che al gioco partecipino 4 giocatori e che per perdere il layer L sia necessario che 3 o più descrizioni vengano perse (in relazione alla capacità correttiva del codice FEC e a come è stata costruita la matrice di codifica). Inoltre all'istante i-esimo il vettore di strategie giocate sia $\mathbf{c}_i = (\text{Red}, \text{Yellow}, \text{Green}, \text{Red})$. La probabilità voluta sarà data dall'Eq. (5.11):

$$\begin{aligned} P[\text{Perdita layer L del frame i-esimo}] = & P_{loss,R}(\mathbf{c}_i)P_{loss,Y}(\mathbf{c}_i)P_{loss,G}(\mathbf{c}_i)P_{loss,R}(c_i) + \\ & + P_{loss,R}(\mathbf{c}_i)P_{loss,Y}(\mathbf{c}_i)P_{loss,G}(\mathbf{c}_i)(1 - P_{loss,R}(\mathbf{c}_i)) + \\ & + P_{loss,R}(\mathbf{c}_i)P_{loss,Y}(\mathbf{c}_i)P_{loss,R}(\mathbf{c}_i)(1 - P_{loss,G}(\mathbf{c}_i)) + \\ & + P_{loss,Y}(\mathbf{c}_i)P_{loss,G}(\mathbf{c}_i)P_{loss,R}(\mathbf{c}_i)(1 - P_{loss,R}(\mathbf{c}_i)) + \\ & + P_{loss,R}(\mathbf{c}_i)P_{loss,G}(\mathbf{c}_i)P_{loss,R}(\mathbf{c}_i)(1 - P_{loss,Y}(\mathbf{c}_i)) \end{aligned} \quad (5.11)$$

Lo stesso sistema si applica per tutti gli altri layer del flusso scalabile. Si osserva come tale probabilità non dipenda solo dalla giocata del singolo giocatore, ma attraverso $P_{loss,k}(\mathbf{c}_i)$ è funzione delle strategia giocate dagli altri concorrenti.

5.3.2 Tipologia di “Gioco” adottato e equilibrio di Nash

Definendo opportunamente la funzione utilità è stato possibile stabilire il “gioco” che viene rappresentato in *forma strategica* in Tab. 5.1 (per semplicità di notazione è stato rappresentato il caso con solo due giocatori).

G_1/G_2	$c_i^{G_2} = G$	$c_i^{G_2} = Y$	$c_i^{G_2} = R$
$c_i^{G_1} = G$	$f_{G_1}(G, G)$ $f_{G_2}(G, G)$	$f_{G_1}(G, Y)$ $f_{G_2}(G, Y)$	$f_{G_1}(G, R)$ $f_{G_2}(G, R)$
$c_i^{G_1} = Y$	$f_{G_1}(Y, G)$ $f_{G_2}(Y, G)$	$f_{G_1}(Y, Y)$ $f_{G_2}(Y, Y)$	$f_{G_1}(Y, R)$ $f_{G_2}(Y, R)$
$c_i^{G_1} = R$	$f_{G_1}(R, G)$ $f_{G_2}(R, G)$	$f_{G_1}(R, Y)$ $f_{G_2}(R, Y)$	$f_{G_1}(R, R)$ $f_{G_2}(R, R)$

Tabella 5.1. Rappresentazione normale del gioco delle classificazioni.

Come si può notare tale gioco presenta un numero finito sia di giocatori che di strategie, quindi sarà un *gioco finito*. Con tale ipotesi si è dimostrato che esiste almeno un equilibrio di Nash in *strategie miste*. In questa condizione di equilibrio tutti i giocatori (nell'ipotesi che siano razionali) vengono "accontentati" e nessuno sarà spinto a giocare una nuova strategia al fine di migliorare la propria utilità. Cioè, fissata la n -upla dell'equilibrio di Nash \mathbf{c}_i , allora per ogni giocatore G_d con $d \in \{0, 1, 2, \dots, n-1\}$ varrà l'Eq. (5.12):

$$f_{G_d}(c_i^{G_0}, \dots, c_i^{G_{d-1}}, c_i^{G_d}, c_i^{G_{d+1}}, \dots, c_i^{G_{n-1}}) \leq f_{G_d}(c_i^{G_0}, \dots, c_i^{G_{d-1}}, \alpha^{G_d}, c_i^{G_{d+1}}, \dots, c_i^{G_{n-1}}),$$

con $\alpha^{G_d} \in \{Green, Yellow, Red\}$

(5.12)

Proprio per tali caratteristiche, le soluzioni del gioco che sono state prese in considerazione sono quelle fornite dall'equilibrio di Nash. Nel caso in cui le soluzioni fossero più di una, cioè $\mathbf{c}_{OPT} \in \{\mathbf{c}_{OPT}^1, \mathbf{c}_{OPT}^2, \dots\}$ è stata scelta quella che minimizzasse il $\delta PSNR$ medio tra gli n giocatori (vedi Eq (5.13)).

$$\delta PSNR_{OPT} = \min_{\mathbf{c}_i \in \{\mathbf{c}_{OPT}^1, \mathbf{c}_{OPT}^2, \dots\}} \left\{ \frac{1}{n} \sum_{k=0}^{n-1} f_k(\mathbf{c}_i) \right\}$$
(5.13)

L'operazione di calcolo dell'equilibrio di Nash e l'eventuale valutazione tra più equilibri viene eseguita per ogni singolo frame. Ottenute le strategie ottime, queste verranno assegnate come priorità alla descrizione o alle $\frac{N}{n}$ descrizioni che ogni utente possiede.

5.4 Valutazione della qualità video

Per valutare la qualità ricevuta del file decodificato, sono stati presi in considerazione i parametri di PSNR e VQM. Sebbene presentino algoritmi di valutazione della qualità differenti, è stato ritenuto che i risultati evidenziati da questi due metodi potessero essere soddisfacenti per una buona analisi delle prestazioni totali.

5.4.1 Peak Signal-to-Noise Ratio (PSNR)

Il Peak Signal to Noise Ratio è una misura adottata per valutare la qualità di un'immagine decodificata rispetto all'originale. Questo indice di qualità delle immagini è definito come il rapporto tra la massima potenza di un segnale e la potenza di rumore che può

invalidare la fedeltà della sua rappresentazione compressa. Poichè molti segnali hanno una gamma dinamica molto ampia, il PSNR è solitamente espresso in termini di scala logaritmica (dB).

Il PSNR è più comunemente usato come misura della qualità per le compressioni di tipo lossy (ad esempio per la compressione delle immagini). Maggiore è il valore del PSNR maggiore è la “somiglianza” con l’immagine originale.

Il PSNR è più facile da definire attraverso l’errore quadratico medio (MSE). Indicando con \mathbf{I} il frame originale e con \mathbf{K} il frame decodificato, entrambi di dimensione $M \times N$, si definisce nell’Eq. (5.14) il $MSE(\mathbf{I}, \mathbf{K})$ (Mean Square Error) tra i due frame all’istante i -esimo.

$$MSE(\mathbf{I}, \mathbf{K})_i = \frac{1}{M \times N} \sum_{x=1}^M \sum_{y=1}^N |I(x, y) - K(x, y)|^2 \quad (5.14)$$

Il $PSNR(\mathbf{I}, \mathbf{K})_i$ viene quindi definito come in Eq. (5.15):

$$PSNR(\mathbf{I}, \mathbf{K})_i = 10 \cdot \log\left(\frac{255^2}{MSE(\mathbf{I}, \mathbf{K})_i}\right) \quad (5.15)$$

Nel caso di sequenze video di durata pari a L frame, il PSNR verrà calcolato come in Eq. (5.16):

$$PSNR = \frac{\sum_{i=1}^L PSNR(\mathbf{I}, \mathbf{K})_i}{L} \quad (5.16)$$

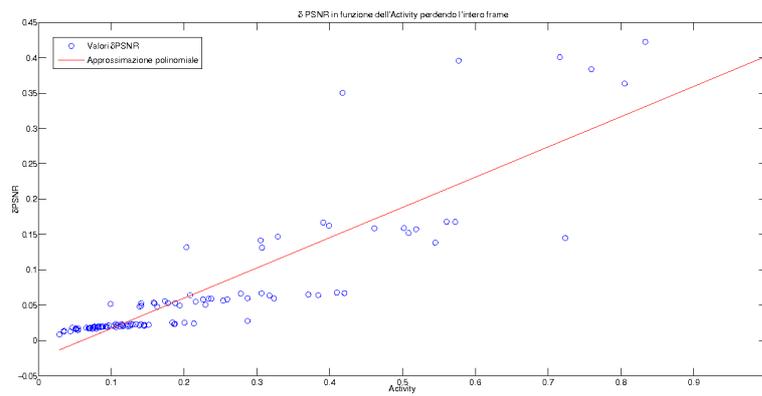
dove ad ogni istante i -esimo \mathbf{I} rappresenta il frame originale, mentre \mathbf{K} rappresenta il frame decodificato.

Si può notare come il valore del PSNR sia un confronto puramente “matematico” tra i singoli frame e non come effettivamente l’occhio umano si comporta in presenza delle diverse tipologie di rumore all’interno del frame. Perciò tale sistema di valutazione viene utilizzato come una prima valutazione della qualità della sequenze ricostruita.

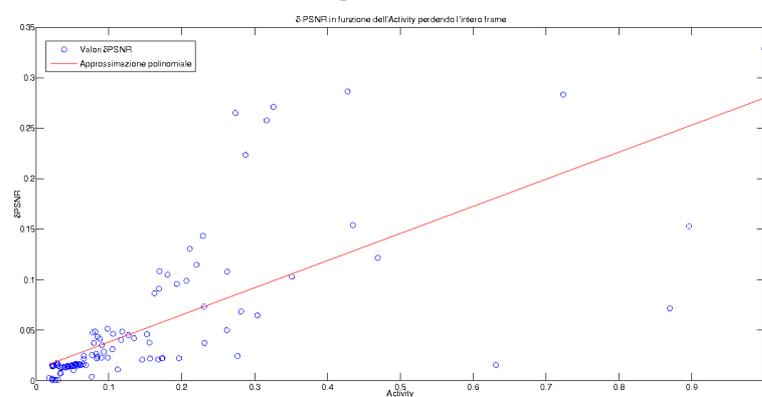
5.4.2 Video Quality Metric (VQM)

Per raffinare l’accuratezza con cui viene stimata la qualità video, è stato utilizzato il sistema di valutazione chiamato Video Quality Metric [36], [37]. Il VQM utilizza una valutazione dell’immagine del punto di vista frequenziale (mediante DCT) quindi pe-

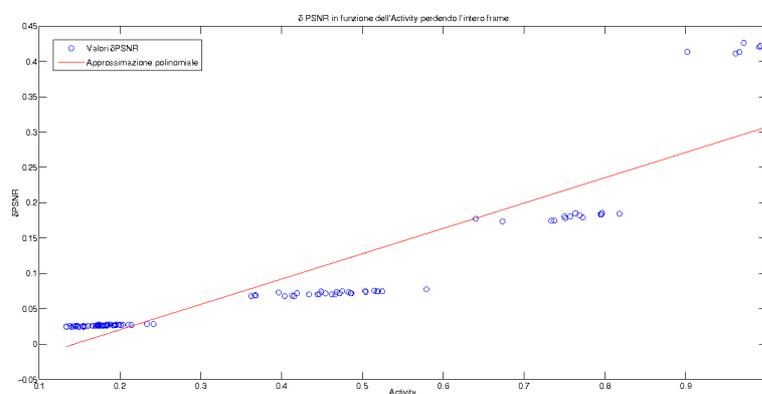
sando diversamente eventuali rumori visivi. Inoltre analizza i possibili artefatti a seconda del contesto nel quale sono posti (ad esempio in un'area fortemente "danneggiata" la presenza di un artefatto sarà percepita in modo diverso che in un'area totalmente limpida). Introducendo questi aspetti si ritiene che le informazioni fornite da questo parametro siano sufficienti per una esaustiva valutazione della qualità video finale.



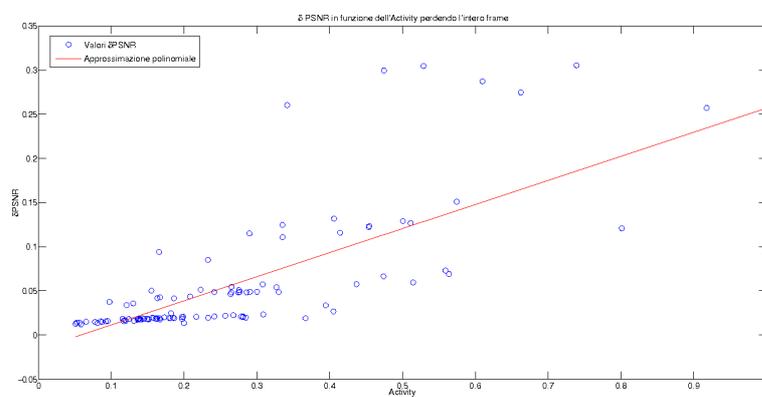
(a) Sequenza "Foreman"



(b) Sequenza "News"

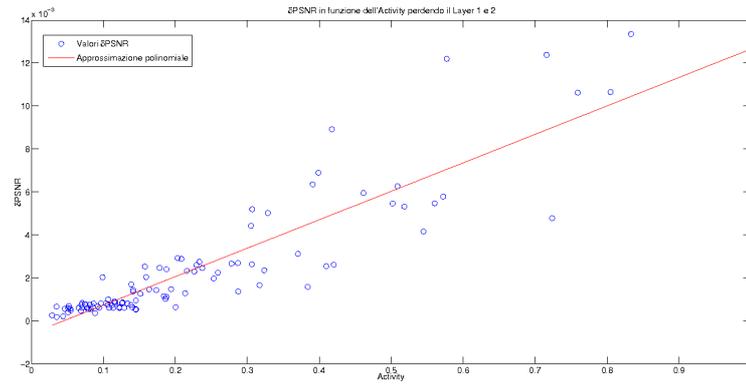


(c) Sequenza "Mobile"

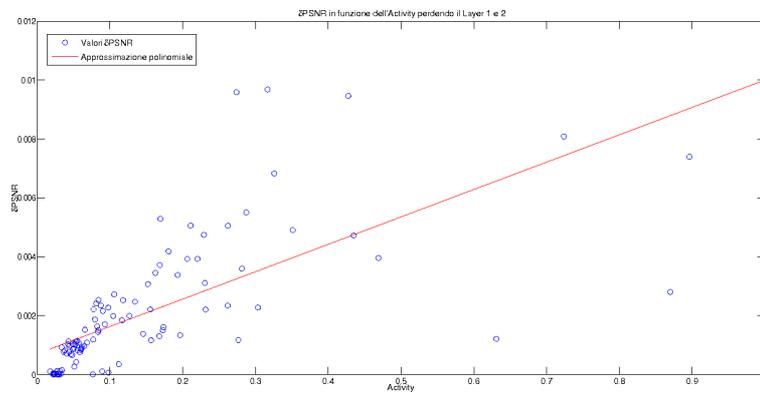


(d) Sequenza "Crew"

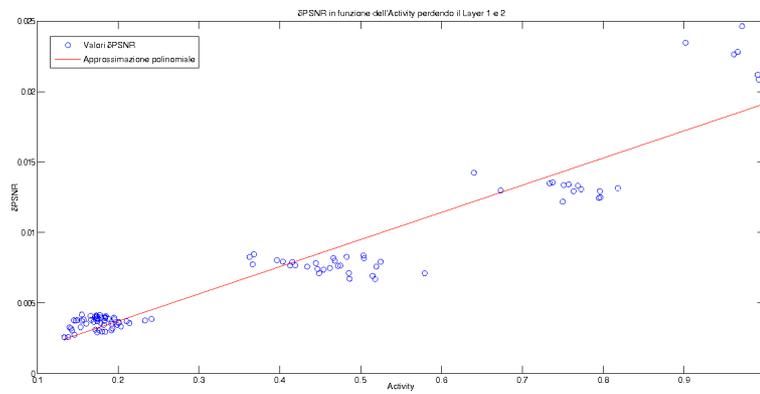
Figura 5.5. $\delta PSNR^0$ in funzione dell'activity



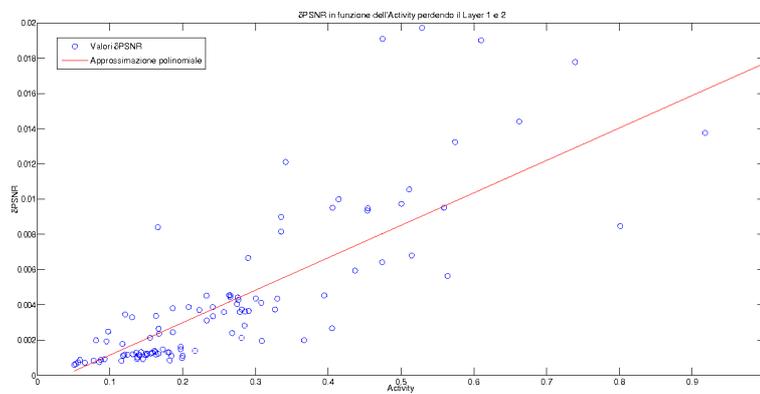
(a) Sequenza "Foreman"



(b) Sequenza "News"

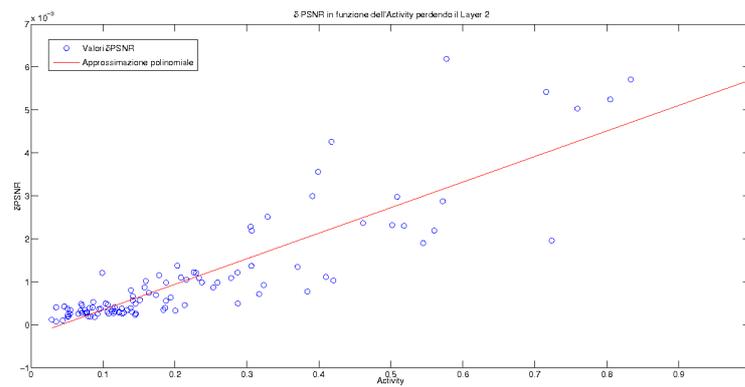


(c) Sequenza "Mobile"

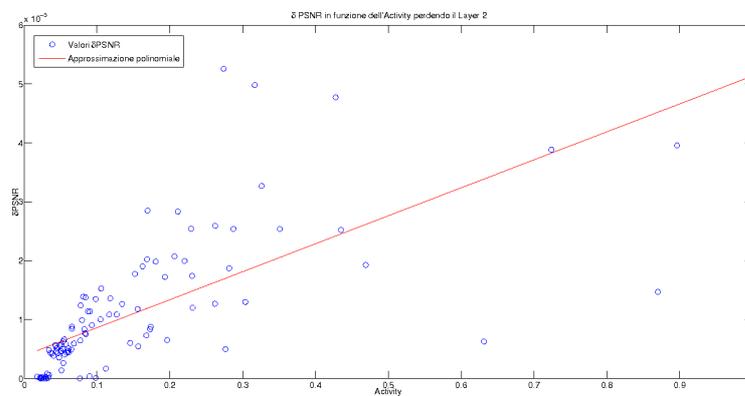


(d) Sequenza "Crew"

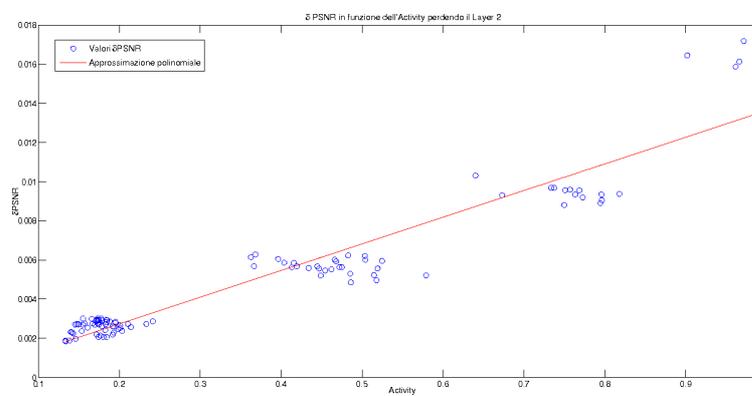
Figura 5.6. $\delta PSNR^1$ in funzione dell'activity



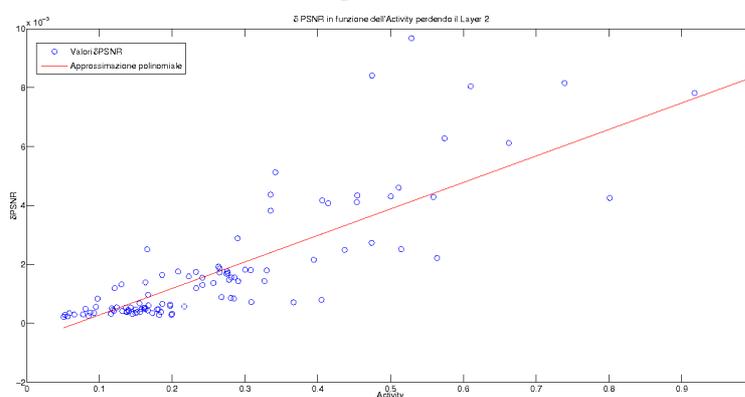
(a) Sequenza "Foreman"



(b) Sequenza "News"



(c) Sequenza "Mobile"



(d) Sequenza "Crew"

Figura 5.7. $\delta PSNR^2$ in funzione dell'activity

Capitolo 6

Risultati sperimentali

6.1 Introduzione

Le simulazioni che sono state eseguite in questo lavoro di tesi faranno riferimento a diverse sequenze video scelte per le loro differenti caratteristiche sia per contenuto frequenziale di un singolo frame che per la velocità di variazione tra frame successivi. Questo permette di fornire dei risultati che comprendano uno spettro più ampio possibile di casi reali. Le sequenze video analizzate sono: “Foreman” (ripresa amatoriale con cambi repentini di direzione con colori poco variabili); “News” (presentazione di un telegiornale con immagini pressochè statiche e colori poco variabili); “Mobile” (ripresa lenta di una serie di giocattoli in movimento con molteplici alternanze di colori); “Crew” (ripresa di un gruppo di astronauti in movimento con la presenza di flash scattati da macchine fotografiche); “Soccer” (ripresa di una partita di calcio con un conseguente aumento della velocità di movimento dell’immagine). Tali sequenze vengono fornite in formato CIF (con risoluzione spaziale di 352x288 pixel) a 30 frame/s.

Tali sequenze sono state successivamente codificate in un bitstream scalabile H.264/SVC composto da 3 layer. È stata utilizzata una scalabilità di qualità adottando la tecnica MGS che permette di creare la gerarchia dei layer ripartendo in modo opportuno i coefficienti della trasformata DCT.

Successivamente, utilizzando la tecnica UEP introdotta nel capitolo precedente, sono state create 12 descrizioni (che chiameremo “pacchetti”) indipendenti. Secondo l’algoritmo proposto, le capacità correttive di ogni layer (vedi Fig. 6.1) sono le seguenti:

- layer0: fino a 9 pacchetti persi;

- layer1: fino a 6 pacchetti persi;
- layer2: fino a 1 pacchetto perso.

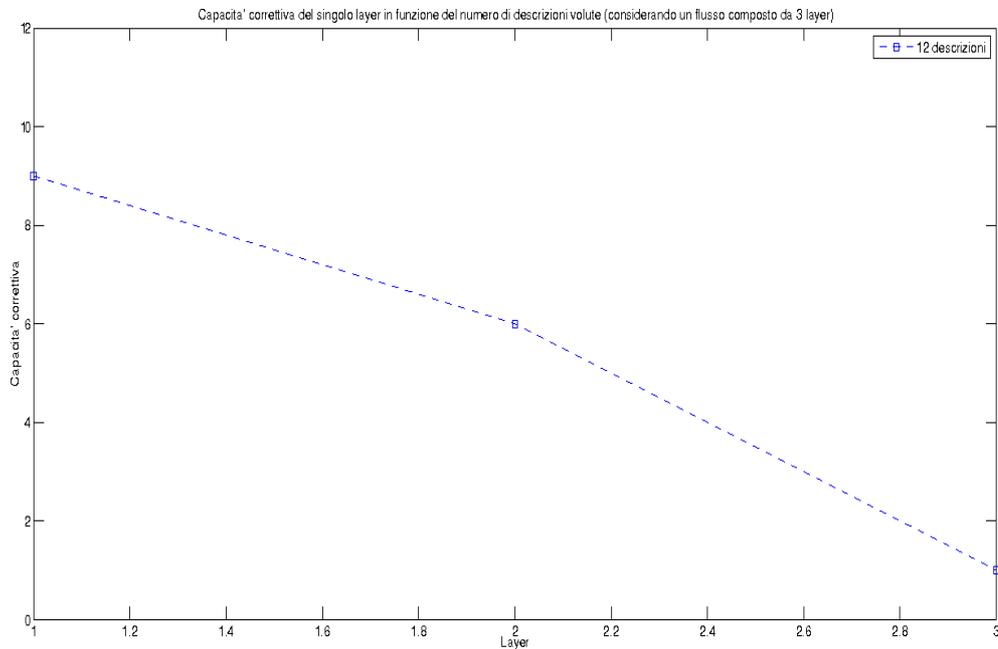


Figura 6.1. Capacità correttiva di ognuno dei tre layer in una MDC a 12 descrizioni

Tuttavia, prima di analizzare i risultati relativi alla trasmissione dei pacchetti su diverse topologie di reti, sono stati eseguiti dei test preliminari relativi alla sola codifica a descrizioni multiple allo scopo di provare la robustezza di tale codifica in presenza di perdite di descrizioni.

6.1.1 Tipo di canale utilizzato (modello di Gilbert)

Al fine di simulare delle condizioni di trasmissione le più realistiche possibili è stato scelto di utilizzare un canale basato sul modello di Gilbert. Tale modello viene caratterizzato dalla probabilità di perdita sul singolo pacchetto P_{Loss} e dalla lunghezza media della serie di perdite di pacchetti consecutivi L_{Burst} . Quest'ultimo parametro consente di tenere in considerazione l'effetto di perdita dei pacchetti causato da un overflow delle code dei nodi di instradamento. Ogni qualvolta si venisse a creare questa condizione, tutti i pacchetti verranno persi in modo sequenziale (creando un effetto di tipo "raffica" o "burst"), finché la congestione non scenderà sotto opportuni livelli. La struttura su cui si basa il principio del modello di Gilbert viene rappresentata in Fig. 6.2.

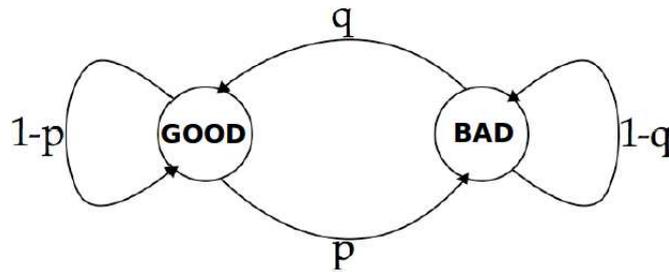


Figura 6.2. Modello di Gilbert del canale

Il modello di Gilbert viene controllato dai parametri p e q che rappresentano le probabilità di transizione relativamente dallo stato “Good” a quello “Bad” e viceversa. Tali probabilità sono legate ai parametri P_{Loss} e L_{Burst} secondo le relazioni nell’ Eq. (6.1).

$$p = \frac{P_{Loss}}{L_{Burst} \cdot (1 - P_{Loss})} \quad (6.1)$$

$$q = \frac{1}{L_{Burst}}$$

Dalla teoria delle catene di Markov, analizzando il comportamento asintotico della catena, si ricava la probabilità che un pacchetto sia corretto Π_G o sia scartato Π_B . Tale probabilità viene descritta nell’Eq. (6.2).

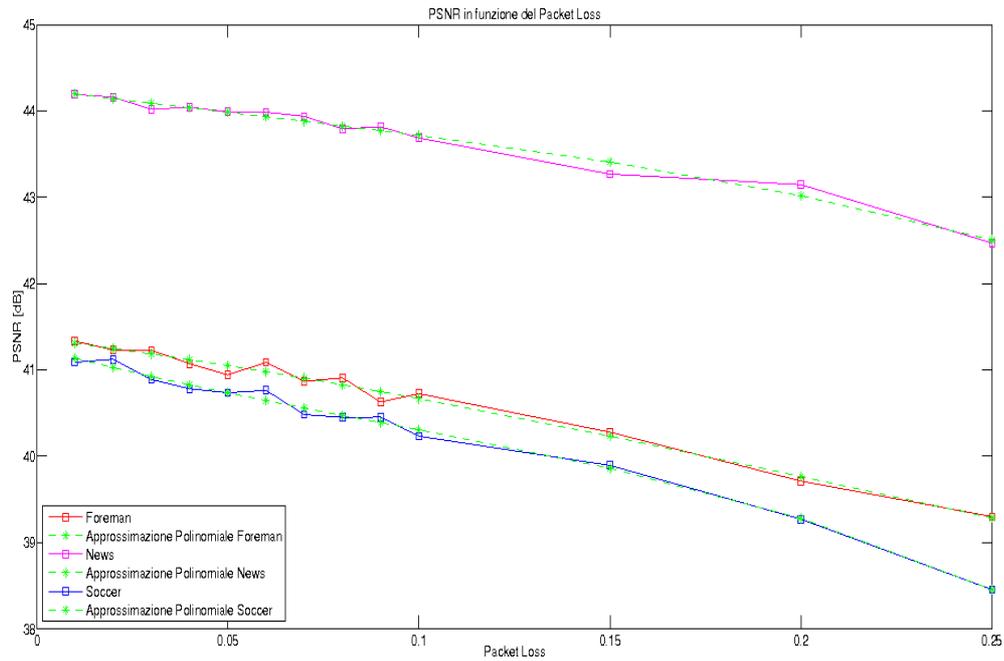
$$\Pi_G = P_{Loss} \quad (6.2)$$

$$\Pi_B = 1 - P_{Loss}$$

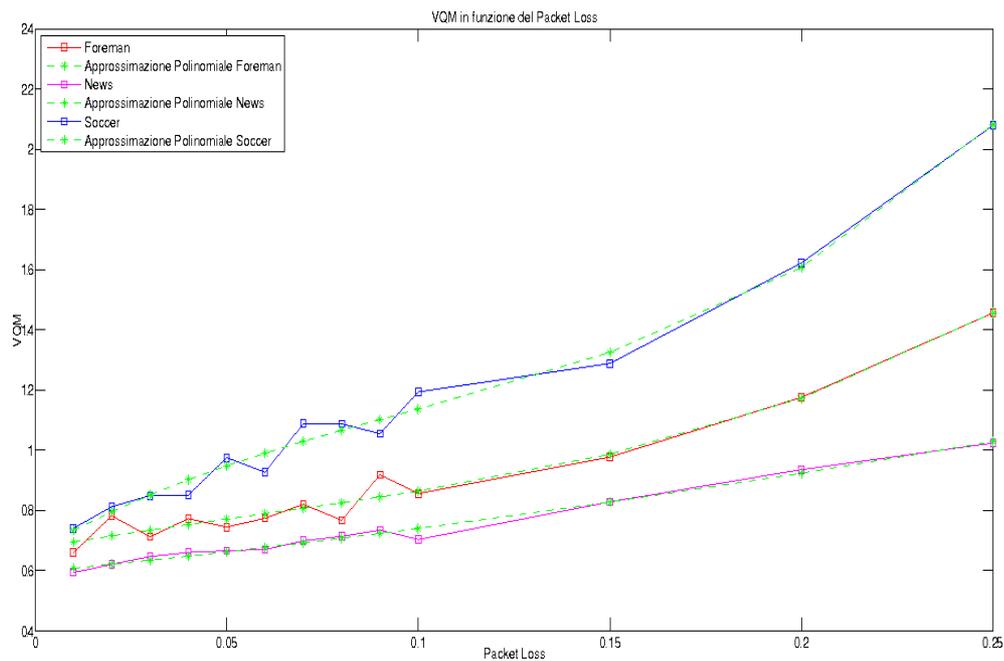
Tuttavia, il comportamento istantaneo presenta un probabilità di passare dallo stato *Bad* a *Good* che decrescerà secondo l’Eq. (6.1), con l’aumentare di L_{Burst} . Questo ad indicare che più grande sarà quest’ultimo parametro e più alta sarà la probabilità che il pacchetto successivo venga perso (ad imitazione del caso di congestione dei nodi di instradamento).

6.2 Risultati codifica a descrizioni multiple

Il primo test è stato eseguito inviando tutte le dodici descrizioni (delle sequenze “Foreman”, “News” e “Soccer”) attraverso **un singolo** canale con perdite (utilizzando un modello di Gilbert) con lunghezza di burst pari a $L_{Burst} = 4$. È stata quindi misurata la qualità ricevuta (PSNR e VQM) in funzione di diverse probabilità di perdita P_{Loss} (vedi Fig. 6.3). Si ricorda che la tecnica di error concealment adottata prevedeva di rimpiazzare il frame perso con quello precedente.



(a) PSNR in funzione della probabilità di perdita del canale



(b) VQM in funzione della probabilità di perdita del canale

Figura 6.3. Qualità ricevuta in funzione della probabilità di perdita del canale (P_{Loss}) per le sequenze “Foreman”, “News”, “Soccer”

È possibile notare come la qualità della sequenza “News”, misurata mediante PSNR sia di gran lunga superiore rispetto alle altre, non in perfetto accordo con la stessa valutazione eseguita con VQM, che invece riduce tale differenza. Questo ad indicare che il PSNR può essere utilizzato come parametro di stima della qualità per una valutazione grossolana di quello che potrebbe essere il risultato finale. Per una valutazione più approfondita è consigliato basarsi sull’informazione data dal VQM.

Inoltre, si può notare come la qualità migliora rispettivamente alle sequenze con minor rapidità di movimento. Questo è dovuto al fatto che essendo i frame tra loro vicini altamente correlati, la tecnica di error concealment utilizzata (rimpiazzando il frame mancante con quello precedente) non introduce una distorsione significativa nel segnale video ricostruito. Ovviamente questo effetto sarà più marcato al diminuire della correlazione tra frame vicini (ad esempio nella sequenza “Soccer”).

Per tutte le sequenze si nota come l’andamento della qualità decresca con il crescere della probabilità di perdita, assumendo un andamento quasi esponenziale per $P_{Loss} > 0.15$. Tuttavia, per P_{Loss} relativamente basse si osserva una non perfetta monotonicità dell’andamento della curva. Tale fenomeno potrebbe essere causato dal fatto che in H.264/SVC la predizione dei frame avviene in modo gerarchico e nel caso in cui venisse perso un frame con activity alta, si andrebbe incontro ad un netto peggioramento della qualità media di tutta la sequenza. Inoltre, tale fenomeno verrebbe accentuato dal fatto che essendoci una probabilità di perdita molto piccola l’alterazione o la perdita dei frame verrebbe risentita maggiormente.

Infine si può notare come per P_{Loss} vicine al 0.25% (che identifica un canale fortemente corrotto), utilizzando la tecnica UEP si ha una perdita di qualità che si aggira attorno ai 2 dB, meno della metà della qualità originale, seppur mantenendo dei valori di PSNR sopra i 38 dB ad indicare un alto livello qualità visiva tale da nascondere l’eventuale perdita.

Ulteriori prove

Al fine di testare la qualità ricevuta limitando la banda occupata, delle dodici descrizioni sono state trasmesse solamente le prime i , inviandole rispettivamente su i canali **indipendenti** (modello di Gilbert) con $P_{Loss} = 0.15$ e $L_{Burst} = 4$. La probabilità utilizzata rappresenta un canale con un livello “medio” di perdita. L’andamento del PSNR in funzione del numero di descrizioni inviate viene rappresentato in Fig. 6.4.

Da questo risultato si vede come al crescere delle descrizioni inviate aumenti anche la capacità correttiva del codice che permette di decodificare i tre layer di cui è compo-

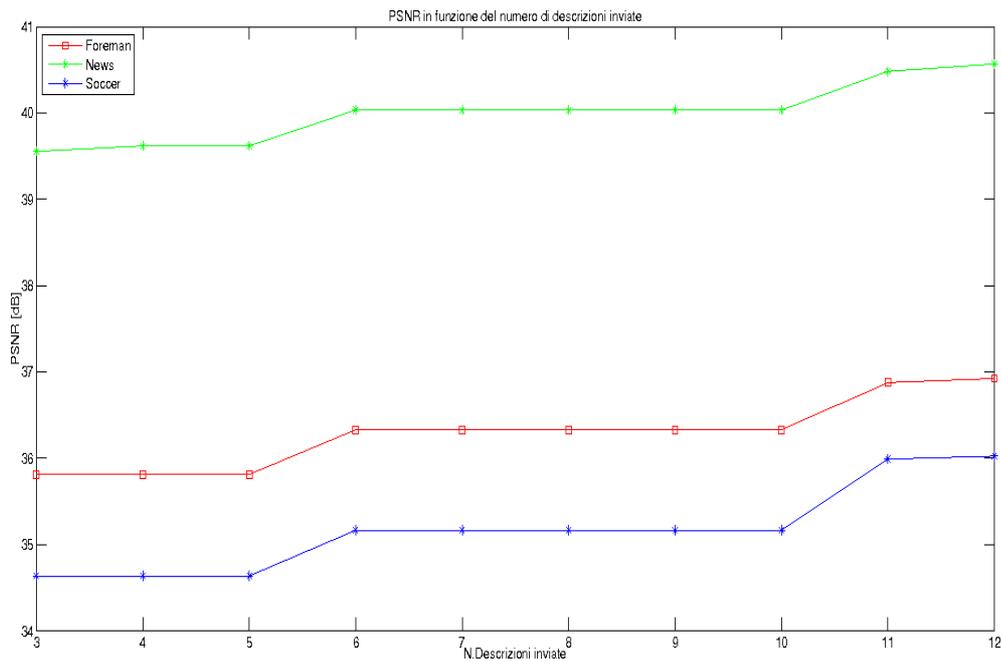


Figura 6.4. PSNR in funzione del numero di descrizioni inviate

sto il flusso scalabile. I diversi layer sono visibili in corrispondenza delle zone piatte di Fig. 6.4. In questo modo è possibile ridurre l'informazione inviata e di conseguenza la banda utilizzata, a discapito della capacità di identificare e correggere gli eventuali pacchetti persi.

6.3 Risultati della classificazione basata sulla Teoria dei Giochi

Per valutare le performance della classificazione basata sulla Teoria dei Giochi, sono state prese in esame tre tipologie di congestione, simulate utilizzando NS2. Inoltre, è stato eseguito il paragone di tale classificazione con altri due approcci: il primo assegnando delle priorità "fisse" che non variano a seconda delle condizioni di congestione [24]; mentre il secondo utilizzando il sistema DiffServ srTCM in modalità Color-Blind mode [33].

6.3.1 Congestione sulla core network

La prima configurazione adotta la topologia di Fig. 6.5.

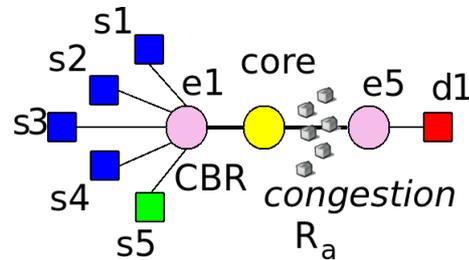
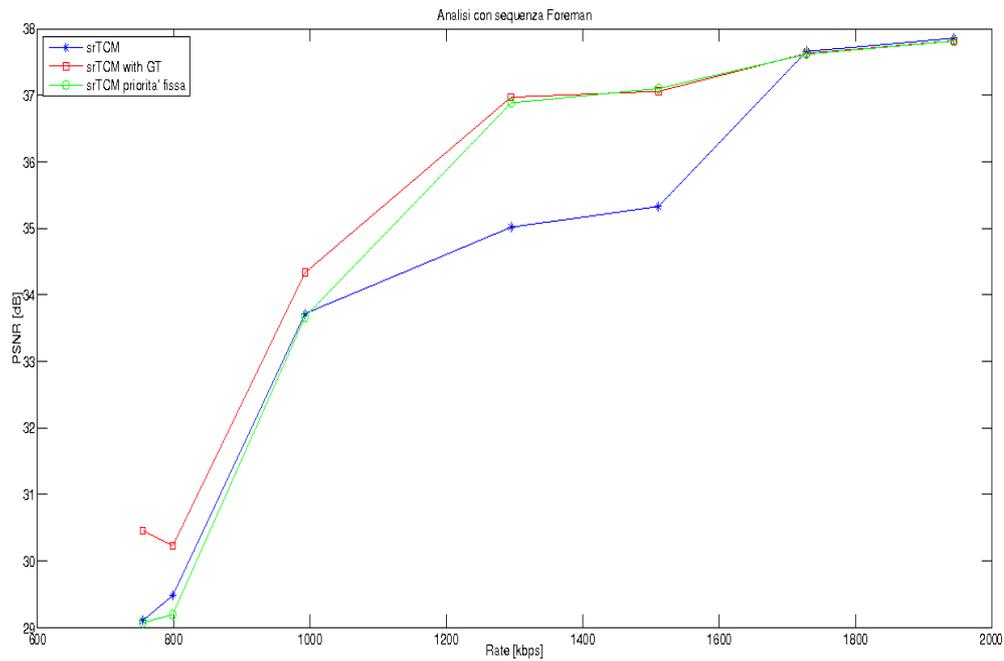


Figura 6.5. Configurazione 1: congestione sulla core network

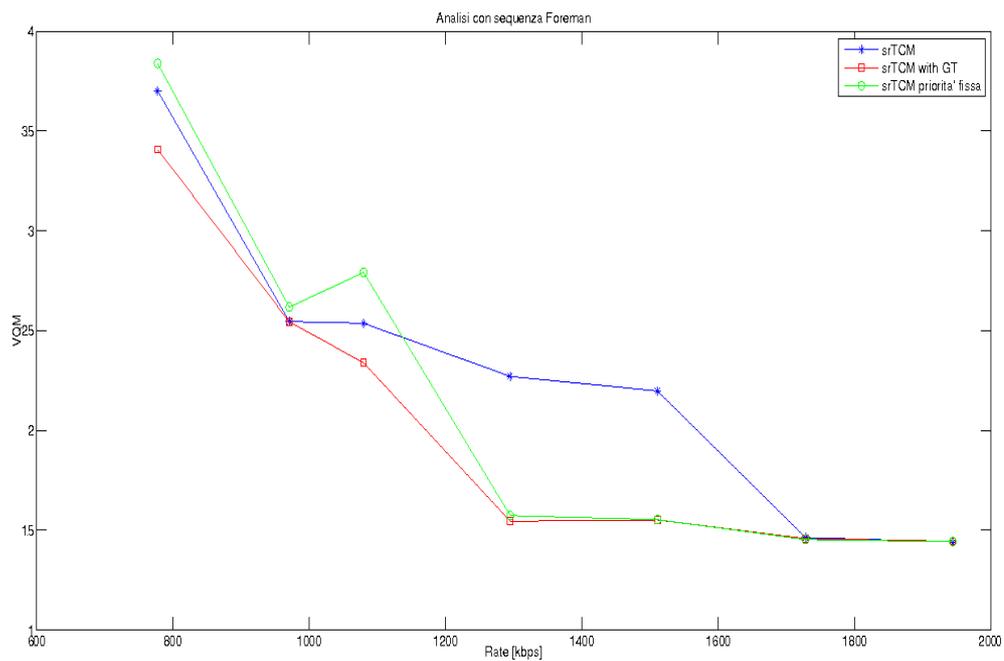
In tale configurazione, il nodo *d1* sta eseguendo il downloading delle descrizioni posizionate nelle sorgenti *s1*, *s2*, *s3*, *s4*, mentre il nodo *s5* fornisce un flusso a bit rate costante CBR. Delle dodici descrizioni create, è stato scelto di partizionarle nei nodi *s1* – *s4* a gruppi di tre. Inoltre, la congestione sulla core network *core* – *e5* è stata simulata scalando la banda massima ($R_a = 5 \cdot R_i$, con R_i si intende il rate medio del nodo *i*-esimo) per i fattori $\{0.30, 0.40, 0.50, 0.60, 0.70, 0.90\}$. Le simulazioni sono state eseguite considerando tre tipi classificatori posizionati nel nodo di Edge *e1* in modo da fornire un servizio di tipo DiffServ al nodo di Core *e5*. Sono state prese in considerazione le classificazioni basate su: Teoria dei Giochi (introdotta nel capitolo precedente); classificazione “fissa”, cioè pre-colorando i pacchetti del nodo *s1* come Green, dei nodi *s2*, *s3* come Yellow e del nodo *s4* come Red; e infine sul sistema DiffServ srTCM in modalità Color-Blind mode. Nelle Fig. 6.6, 6.7, 6.8, 6.9 sono presenti i valori della qualità ricevuta (misurata con PSNR e VQM) in funzione del rate della core network relativamente alle sequenze “Foreman”, “News”, “Mobile”, “Crew”.

In questo tipo di configurazione si nota immediatamente un guadagno nell’utilizzo della classificazione basata sulla Teoria dei Giochi, la quale consente un incremento massimo della qualità di circa 2 dB rispetto alla semplice applicazione del sistema srTCM. Tale guadagno può diminuire restando pur sempre positivo rispetto alla classificazione *fissa*. Inoltre, si osserva che per bassi bitrate (forte congestione) le prestazioni determinate delle tre tipologie di assegnazione di priorità sono pressochè uguali. Questo ad indicare che oltre un certo limite inferiore di packet loss anche un’assegnazione “mirata” delle priorità risulta poco efficace (se pur migliore delle altre) nel contrastare la perdita di pacchetti. Un comportamento analogo si osserva anche per elevati bitrate (bassa congestione) tali che la congestione viene ridotta al minimo se

non eliminata completamente. In quest'ultimo caso, una classificazione (di qualunque tipo) non produrrebbe alcun vantaggio ma seguirebbe l'andamento del sistema privo di pre-classificazione.

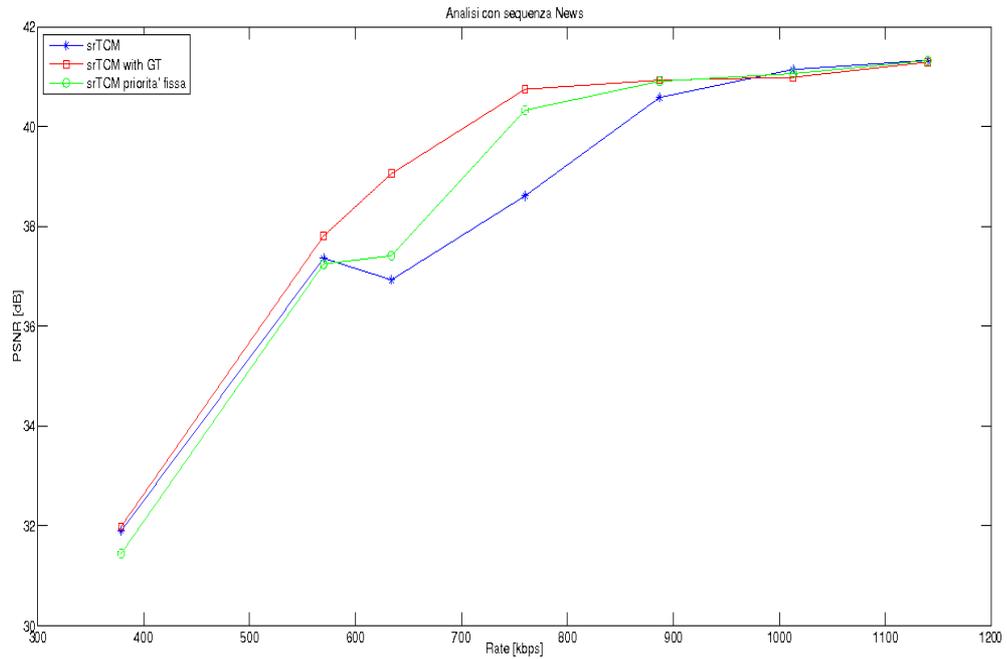


(a) PSNR in funzione del rate nella core network

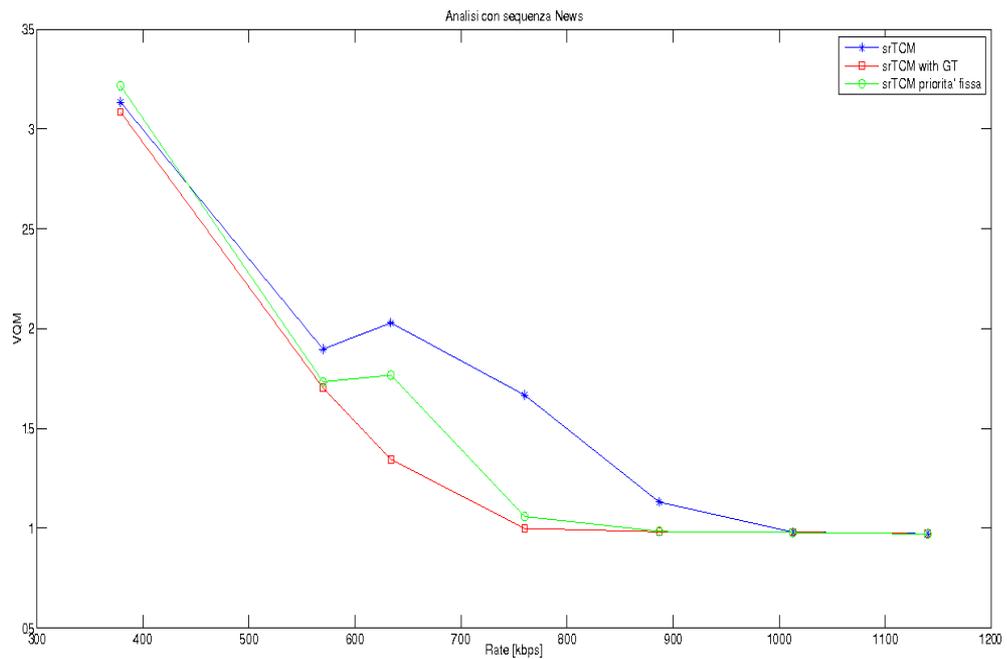


(b) VQM in funzione del rate nella core network

Figura 6.6. Analisi delle classificazioni alla sequenza “Foreman” nella topologia di rete di Fig. 6.5

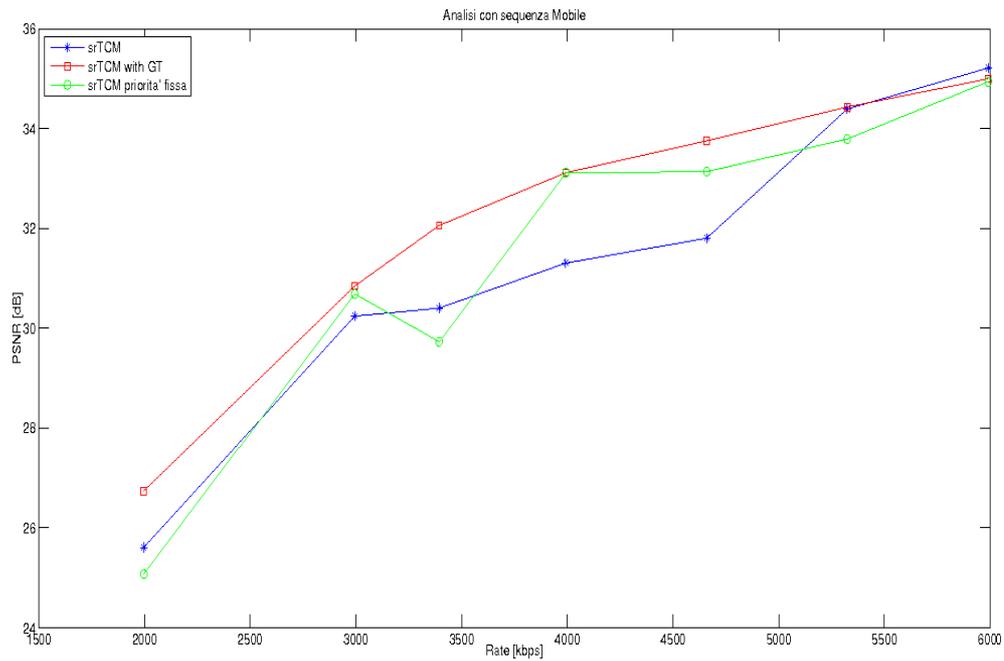


(a) PSNR in funzione del rate nella core network

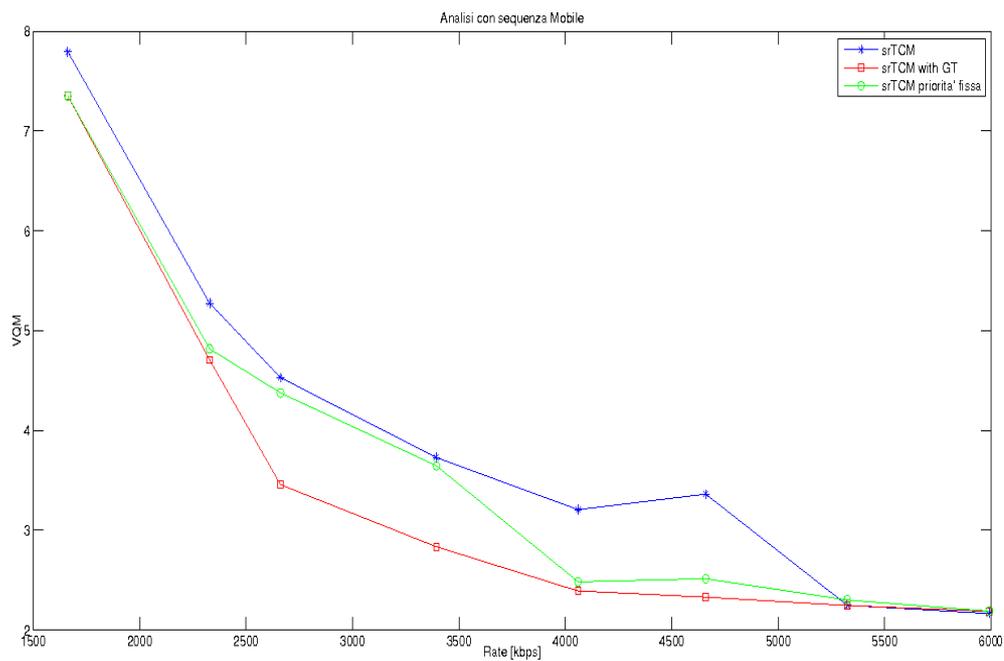


(b) VQM in funzione del rate nella core network

Figura 6.7. Analisi delle classificazioni alla sequenza “News” nella topologia di rete di Fig. 6.5

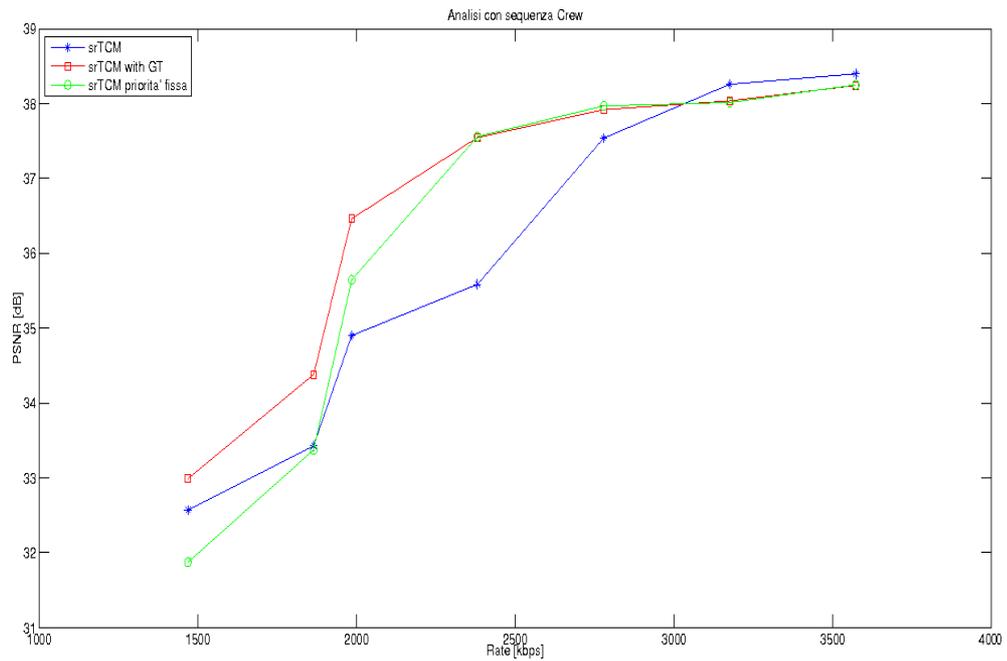


(a) PSNR in funzione del rate nella core network

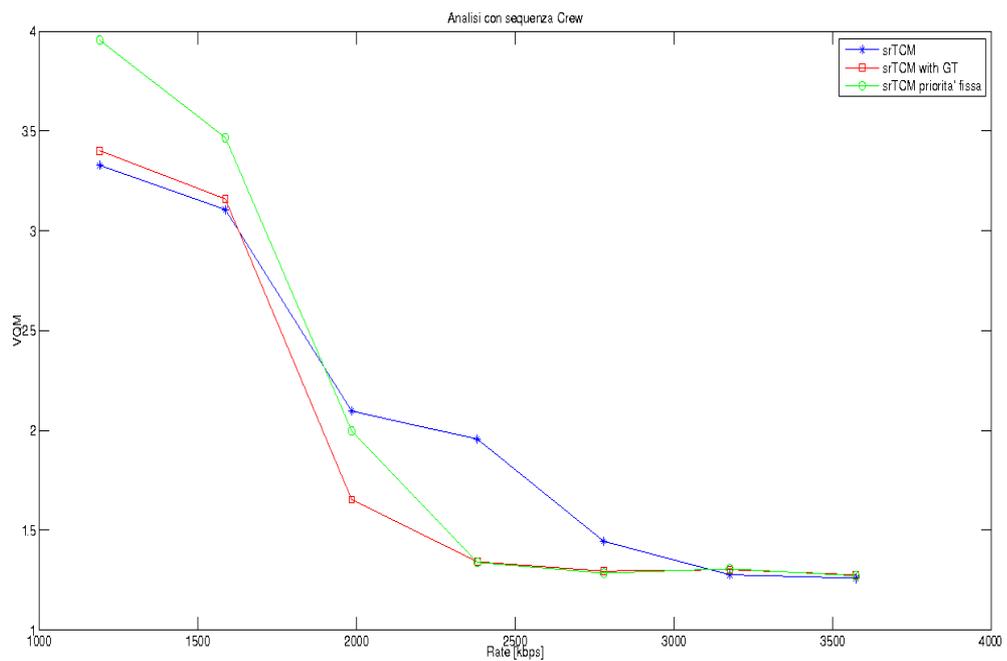


(b) VQM in funzione del rate nella core network

Figura 6.8. Analisi delle classificazioni alla sequenza “Mobile” nella topologia di rete di Fig. 6.5



(a) PSNR in funzione del rate nella core network



(b) VQM in funzione del rate nella core network

Figura 6.9. Analisi delle classificazioni alla sequenza “Crew” nella topologia di rete di Fig. 6.5

6.3.2 Congestione su link non condiviso

La seconda configurazione di rete adottata è rappresentata in Fig. 6.10.

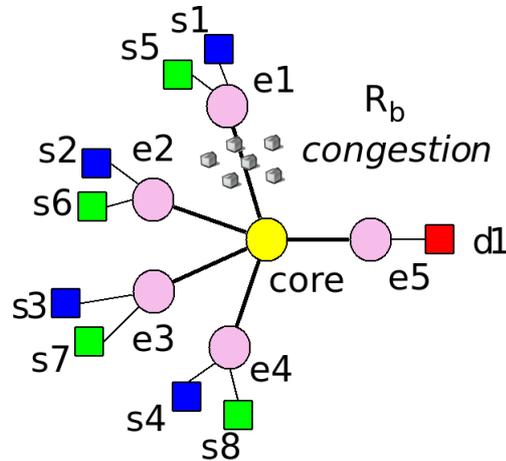


Figura 6.10. Configurazione 2: congestione su link non condiviso

In tale situazione la congestione interessa solamente le tre descrizioni poste nel nodo s_1 , lasciando invariata la banda della core network e quella relativa agli altri tre link. La classificazione dei pacchetti viene effettuata in tutti i nodi di Edge $e_1 - e_4$. Nel caso in esame, la congestione è stata simulata riducendo la banda disponibile, $R_b = 2 \cdot R_i$ del link $e_1 - core$ con riferimento ai seguenti fattori di scalamento $\{0.15, 0.25, 0.35, 0.45, 0.55, 0.65, 0.75, 0.85, 0.95, 1.0\}$. Nelle Fig. 6.11, 6.12, 6.13, 6.14 sono presenti i valori della qualità ricevuta (misurata con PSNR e VQM) in funzione del rate del link $e_1 - core$ relativamente alle sequenze “Foreman”, “News”, “Mobile”, “Crew”.

Dai risultati ottenuti si nota che la classificazione, mediante Teoria dei Giochi, comporta un lieve peggioramento delle prestazioni, con un gap massimo di 1.2 dB relativo alla sequenza “Mobile” di Fig. 6.13(a), rispetto alle altre due classificazioni che sono approssimativamente uguali. Tuttavia, notando che la qualità del video decodificato si aggira sui 33 - 34 dB (per la sequenza “Mobile”) mentre per le altre si sale fino a 41 dB. In tal caso un peggioramento della qualità di 1.2 dB risulterebbe pressochè impercettibile. Inoltre si vuole ricordare che nella Teoria dei Giochi si parte dall’ipotesi che i giocatori giochino allo **stesso** gioco. Nel caso di Fig. 6.10 la classificazione (quindi la valutazione del gioco) avviene nei nodi Edge $e_1 - e_4$ che appartengono a flussi **indipendenti**. In tal modo l’ipotesi precedente non viene rispettata comportando un peggioramento, se pur minimo, delle prestazioni globali.

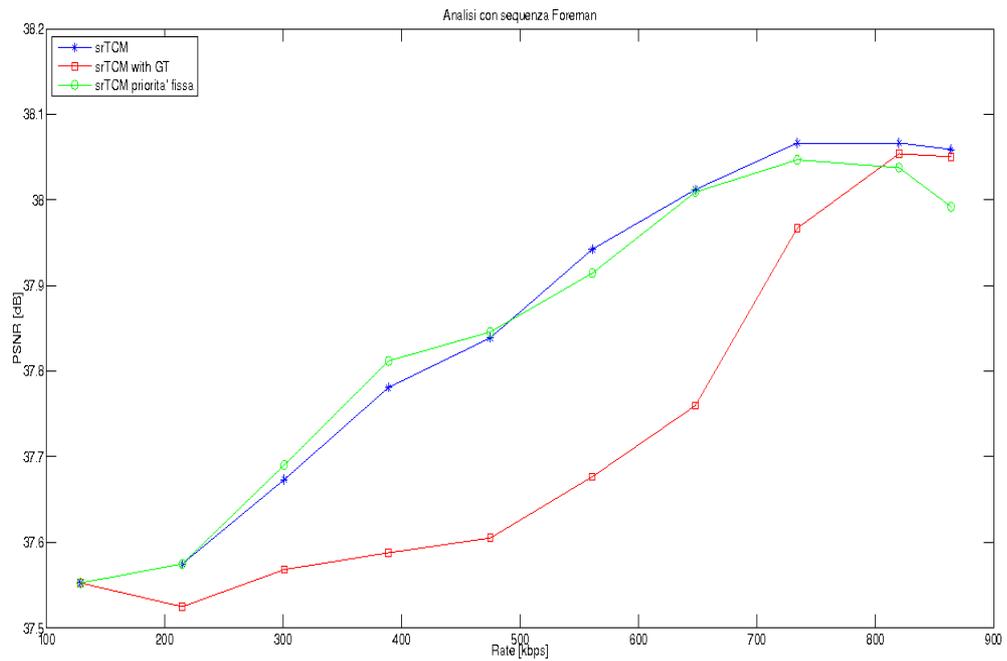
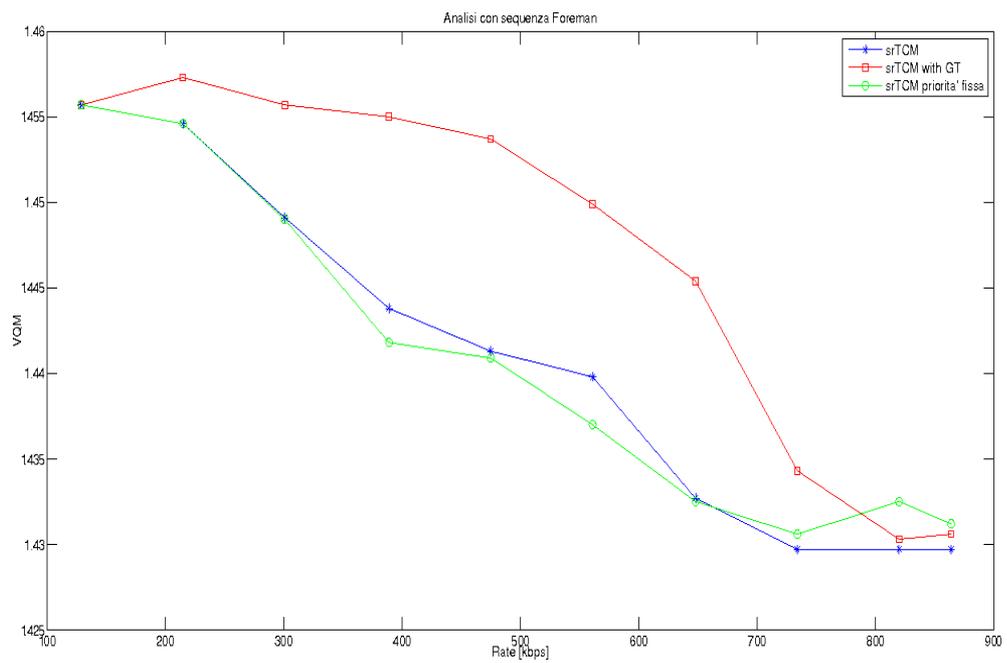
(a) PSNR in funzione del rate nel link $e1 - core$ (b) VQM in funzione del rate nel link $e1 - core$

Figura 6.11. Analisi delle classificazioni alla sequenza “Foreman” nella tipologia di rete di Fig. 6.10

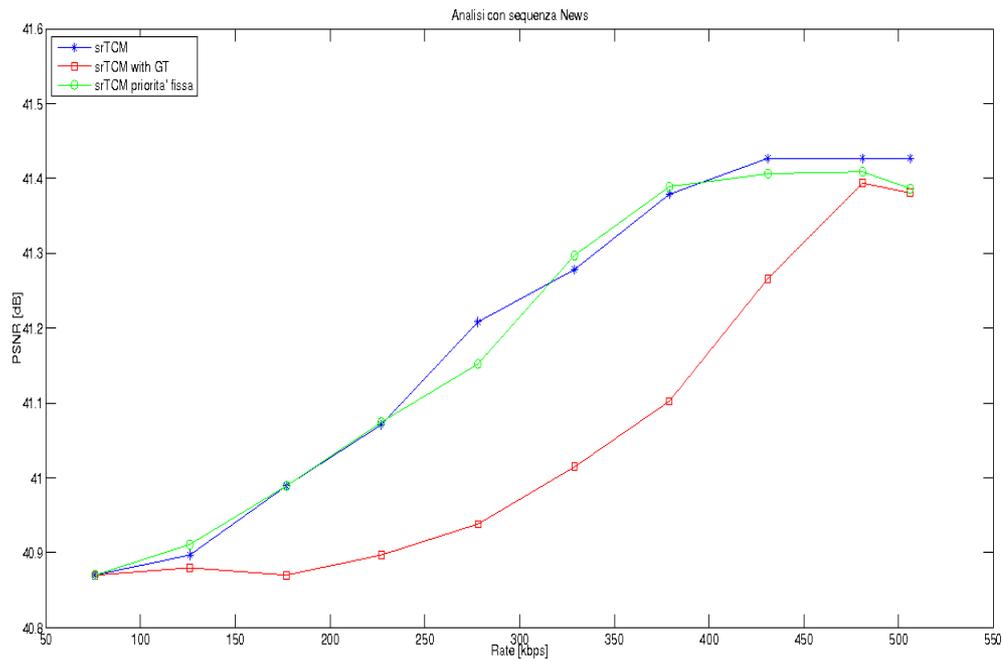
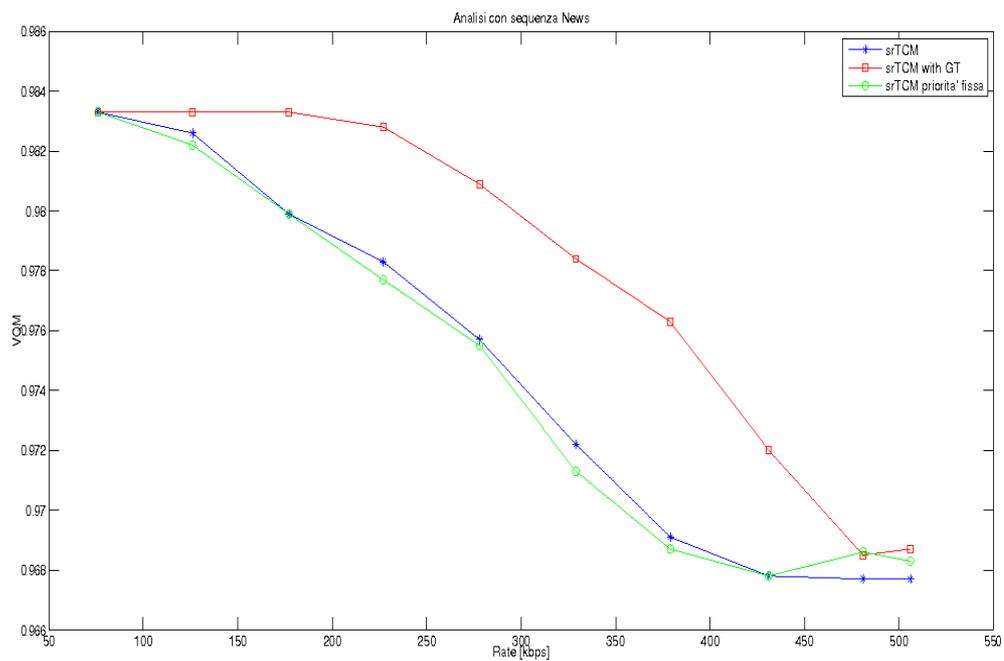
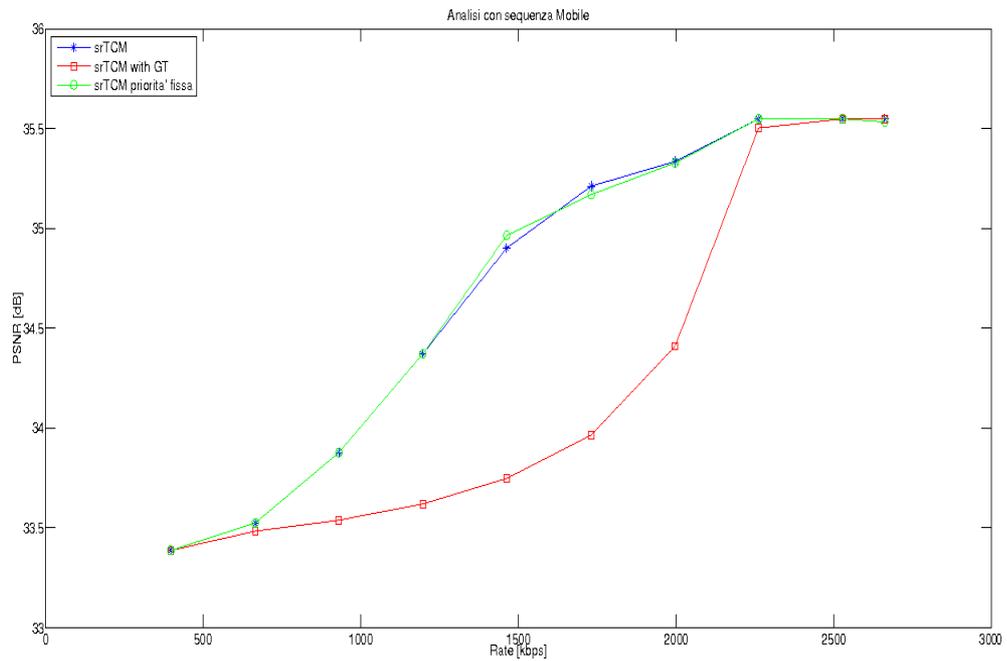
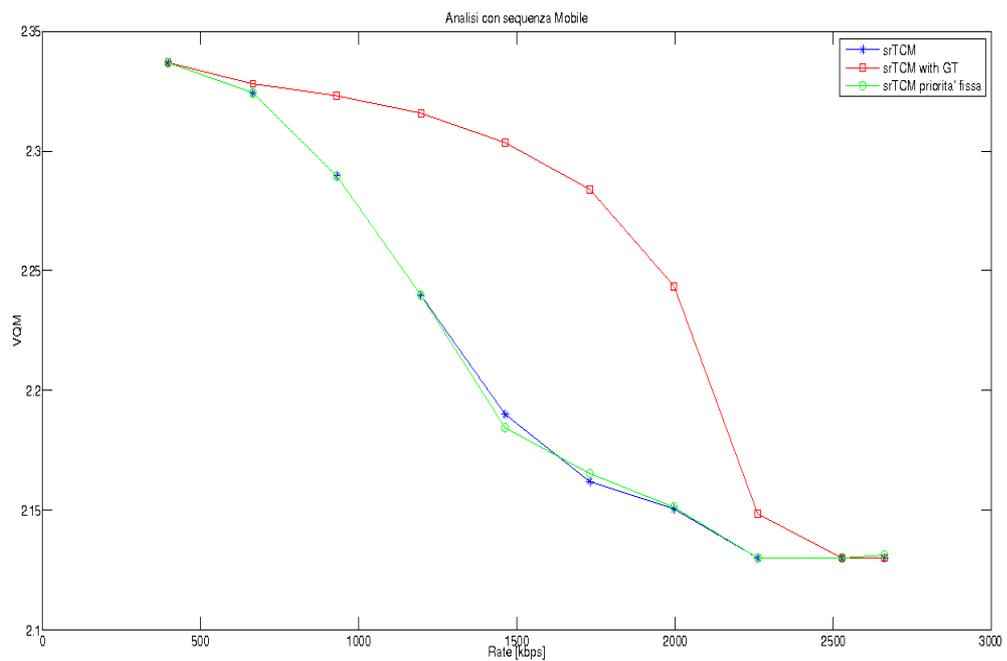
(a) PSNR in funzione del rate nel link $e1 - core$ (b) VQM in funzione del rate nel link $e1 - core$

Figura 6.12. Analisi delle classificazioni alla sequenza “News” nella tipologia di rete di Fig. 6.10

(a) PSNR in funzione del rate nel link $e1 - core$ (b) VQM in funzione del rate nel link $e1 - core$ **Figura 6.13.** Analisi delle classificazioni alla sequenza “Mobile” nella tipologia di rete di Fig. 6.10

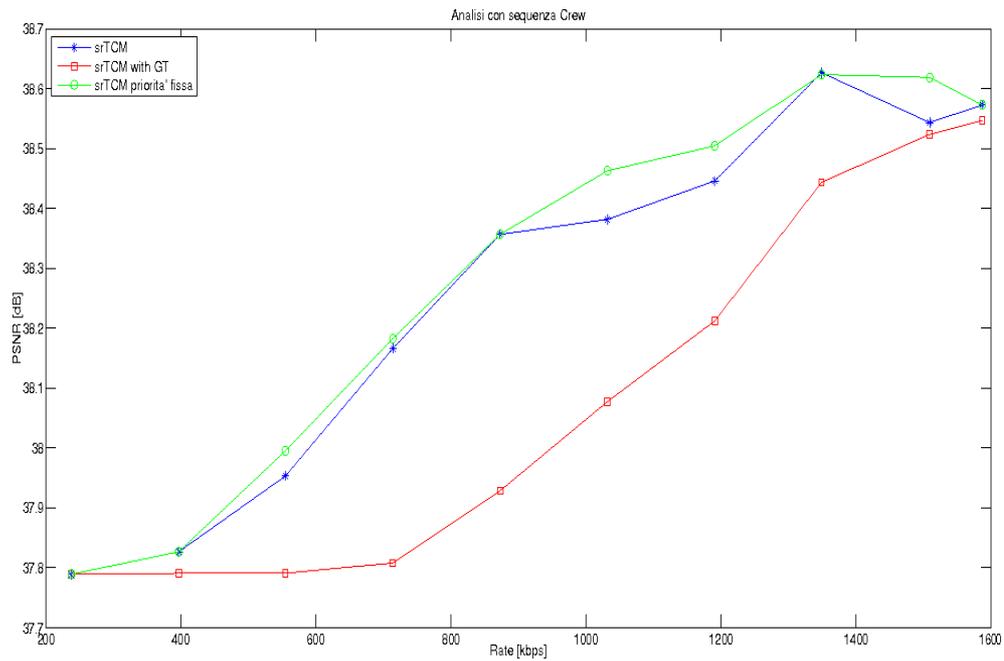
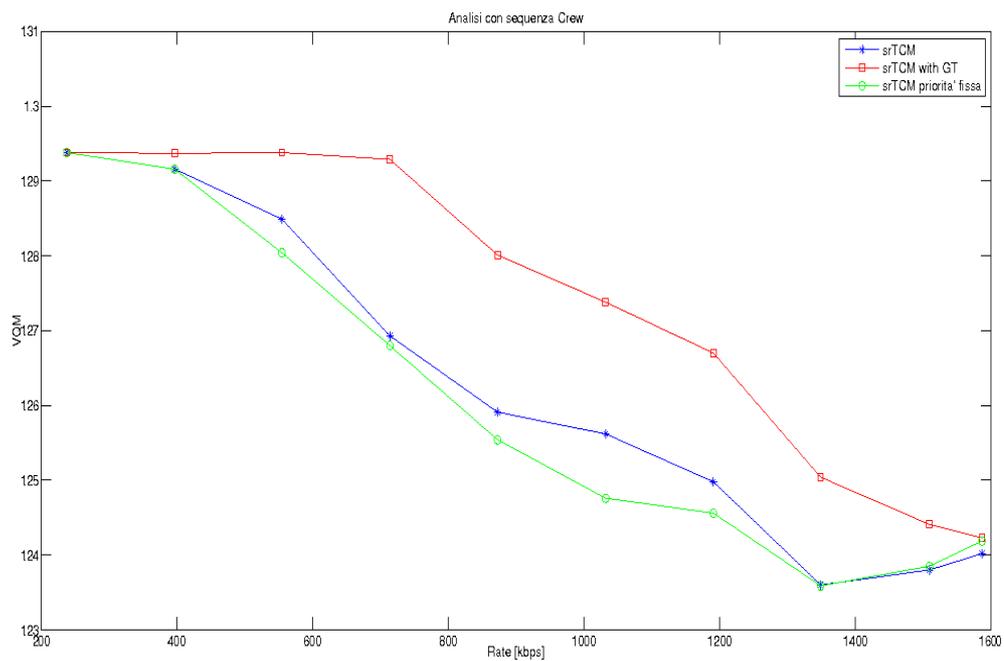
(a) PSNR in funzione del rate nel link $e1 - core$ (b) VQM in funzione del rate nel link $e1 - core$

Figura 6.14. Analisi delle classificazioni alla sequenza “Crew” nella tipologia di rete di Fig. 6.10

6.3.3 Congestione sia su link non condiviso che sulla core network

La terza configurazione di rete adottata è rappresentata in Fig. 6.15.

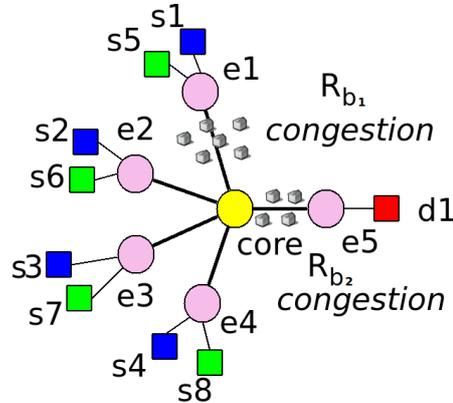


Figura 6.15. Configurazione 3: congestione sia su link non condiviso che sulla core network

In questa configurazione vengono considerati congiuntamente i due casi precedenti, dove la congestione interessa sia il link $e1 - core$ (interessando le descrizioni di $s1$) sia la core network $core - e5$ (interessando tutti i nodi $s1 - s4$), inoltre i nodi $s5 - s8$ producono un flusso CBR. Per la simulazione è stata creata una congestione nella core network scalando la banda R_{b2} di un fattore 0.4, mentre la banda del link $e1 - core$ R_{b1} è stata scalata dei fattori $\{0.15, 0.25, 0.35, 0.45, 0.55, 0.65, 0.75, 0.85, 0.95, 1.0\}$. La classificazione è stata eseguita nei nodi di Edge $e1 - e4$. Nelle Fig. 6.16, 6.17, 6.18, 6.19 sono presenti i valori della qualità ricevuta (misurata con PSNR e VQM) in funzione del rate del link $e1 - core$ relativamente alle sequenze “Foreman”, “News”, “Mobile”, “Crew”; in questo caso si ricorda che è presente anche una congestione nella core network e quindi l’aumentare del throughput del link $e1 - core$ comporta un peggioramento della congestione nel link $core - e5$.

Dai risultati ottenuti si osserva un miglioramento della qualità finale utilizzando la classificazione basata sulla Teoria dei Giochi rispetto agli altri due sistemi di classificazione. Tale guadagno in termini di qualità può arrivare fino a 3 - 4 dB rispetto ad una tecnica di classificazione sia “fissa” che srTCM, comportando un miglioramento di qualità più di due volte superiore agli altri due sistemi di paragone.

Un’osservazione particolare consiste nel fatto che la qualità totale peggiori all’aumentare del rate relativo al link $e1 - core$. A prima vista questa soluzione potrebbe risultare anomala in quanto avendo un incremento del rate nel link $e1 - core$ (con conseguente diminuzione della congestione) ci si aspetterebbe che la qualità aumentasse.

Questo comportamento può essere conseguenza del fatto che con l'aumentare del rate R_{b1} il numero di pacchetti entranti nel link *core - e5* aumenti, ma essendoci una congestione anche nella core network il numero di pacchetti persi crescerà di conseguenza, peggiorando la qualità finale. Questo fa notare che in tale configurazione non è tanto la perdita di descrizioni del nodo *s1* ad influire nella qualità video ricostruito (visto che con la codifica MDC possono essere ricostruite), ma è la congestione della core network che gioca un ruolo fondamentale, in quanto determina la perdita della totalità delle descrizioni inviate, causando un degrado maggiore nella qualità totale.

Inoltre, osservando l'andamento dei grafici, si nota la non perfetta monotonicità delle curve. Questo potrebbe essere causato dalla perdita di un frame con activity elevata che causerebbe un improvviso peggioramento della qualità video ricostruito.

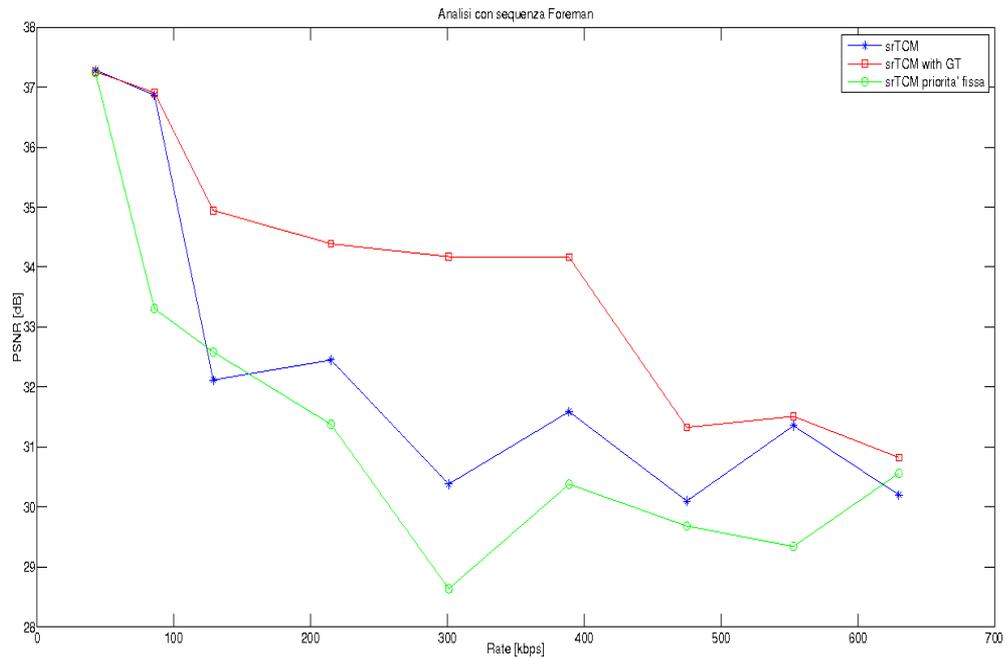
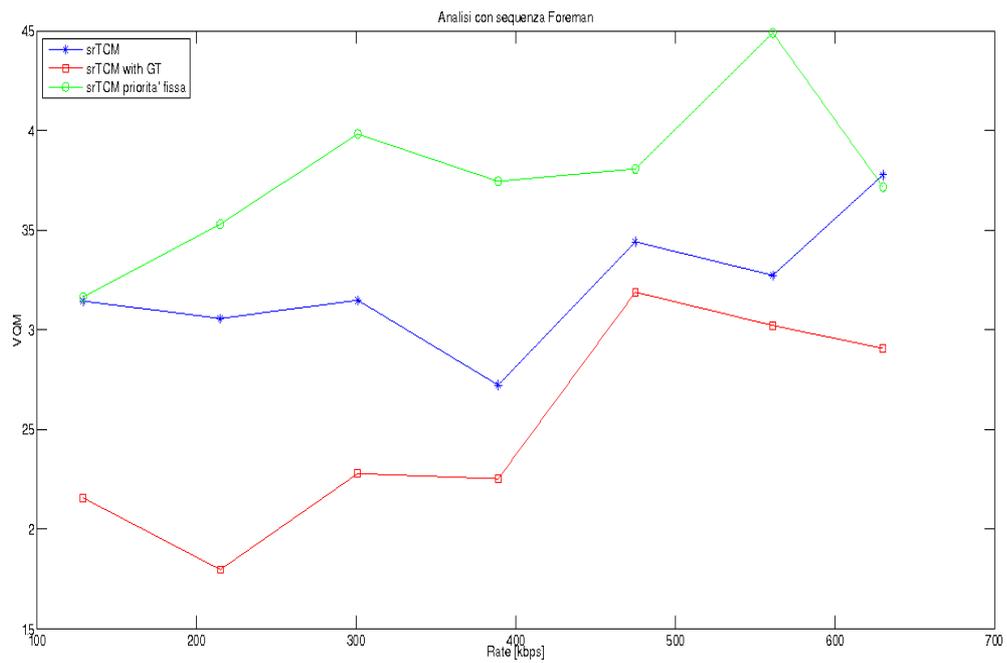
(a) PSNR in funzione del rate nel link $e1 - core$ (b) VQM in funzione del rate nel link $e1 - core$

Figura 6.16. Analisi delle classificazioni alla sequenza “Foreman” nella tipologia di rete di Fig. 6.15

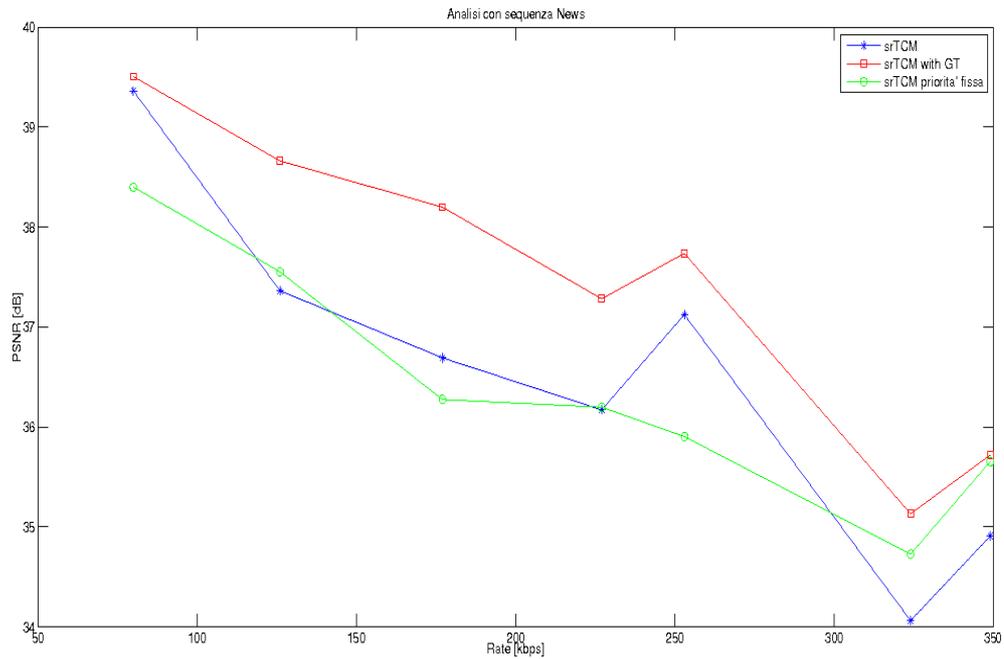
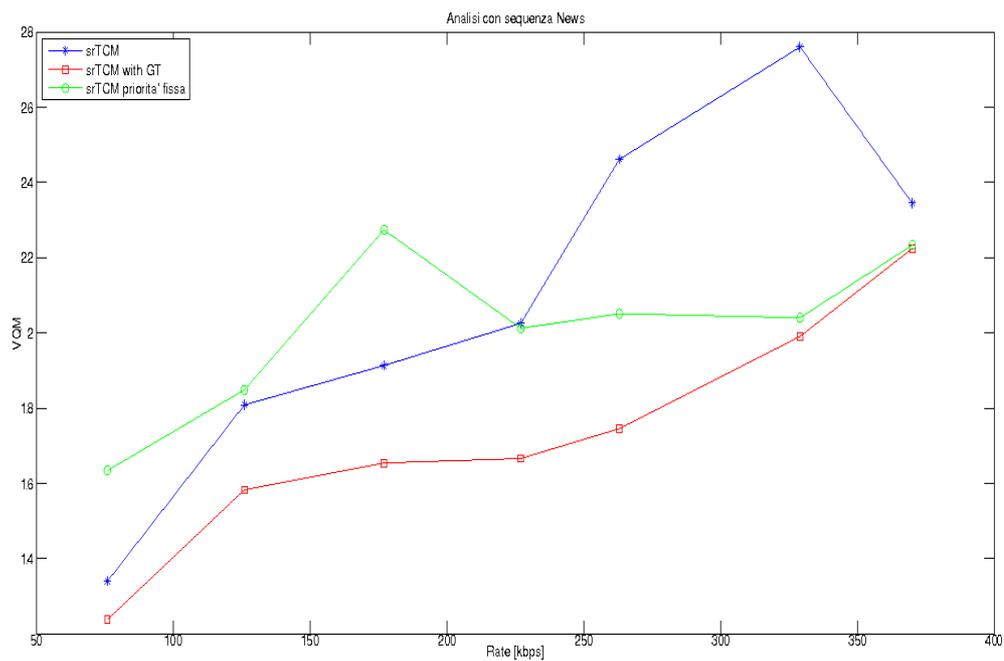
(a) PSNR in funzione del rate nel link $e1 - core$ (b) VQM in funzione del rate nel link $e1 - core$

Figura 6.17. Analisi delle classificazioni alla sequenza “News” nella tipologia di rete di Fig. 6.15

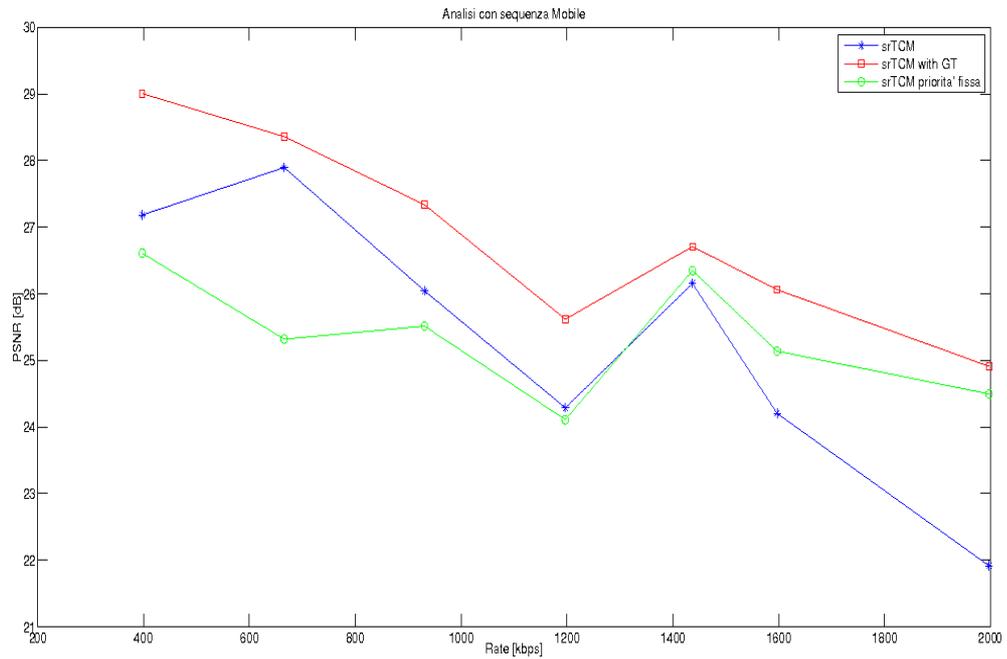
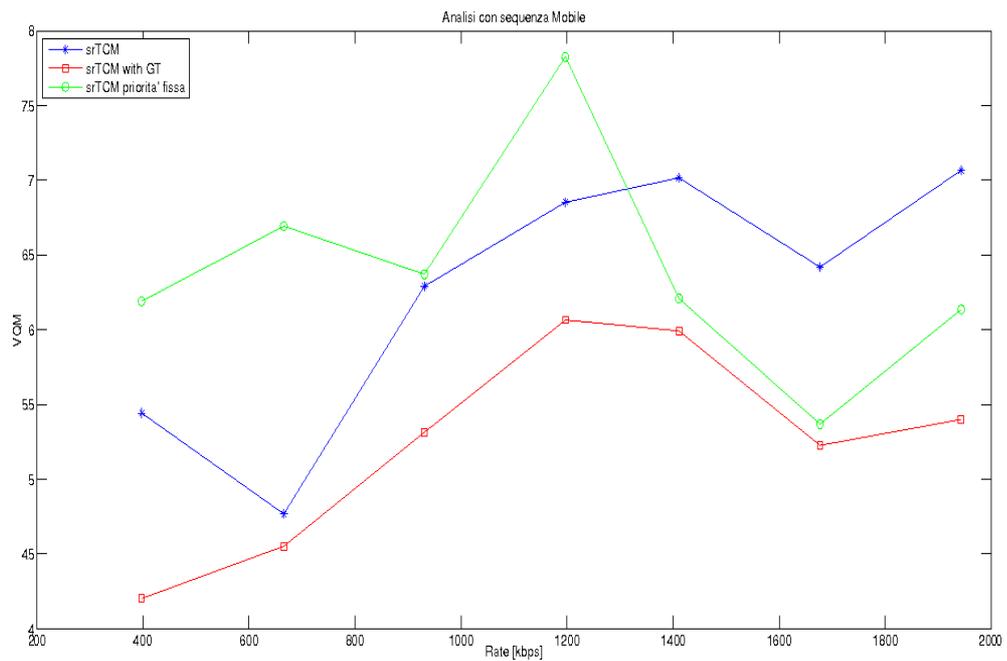
(a) PSNR in funzione del rate nel link $e1 - core$ (b) VQM in funzione del rate nel link $e1 - core$

Figura 6.18. Analisi delle classificazioni alla sequenza “Mobile” nella tipologia di rete di Fig. 6.15

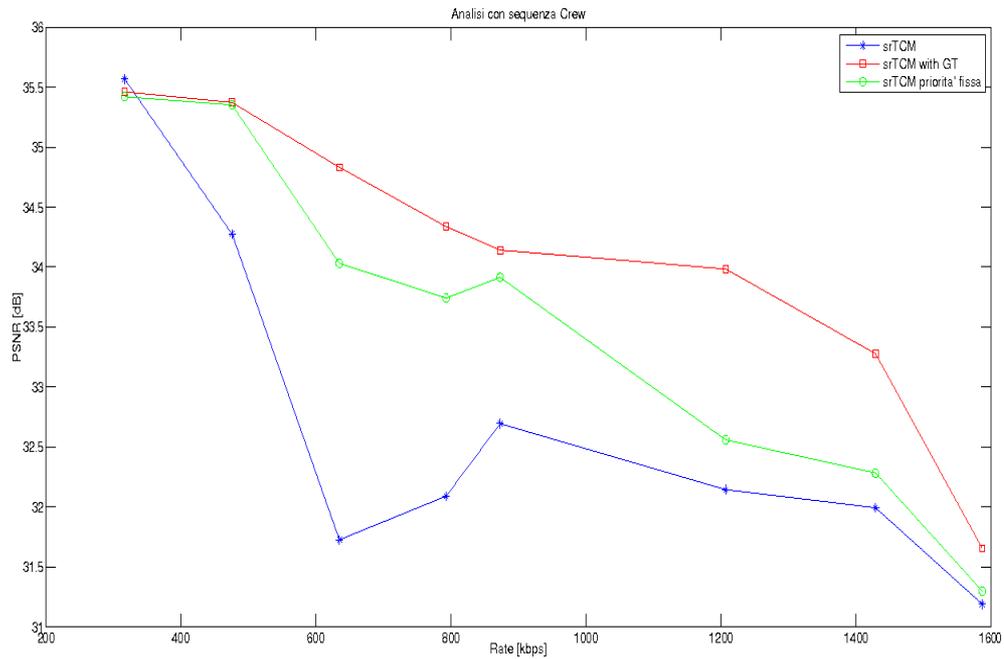
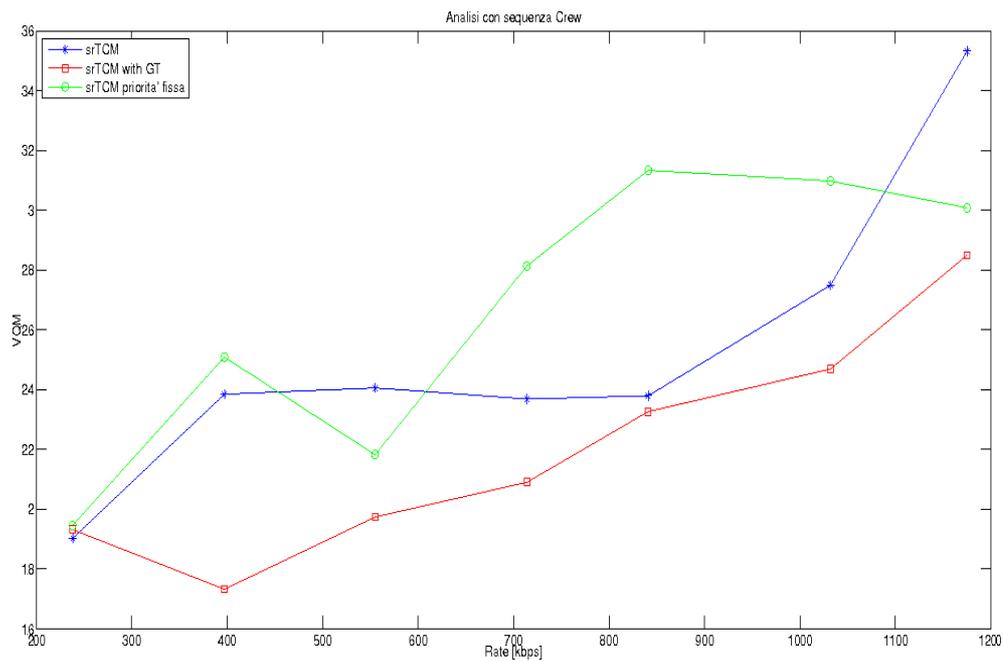
(a) PSNR in funzione del rate nel link $e1 - core$ (b) VQM in funzione del rate nel link $e1 - core$

Figura 6.19. Analisi delle classificazioni alla sequenza “Crew” nella tipologia di rete di Fig. 6.15

Capitolo 7

Conclusioni e sviluppi futuri

Il lavoro di tesi presentato ha portato alla luce alcuni aspetti riguardanti l'ottimizzazione della trasmissione di sequenze video su reti CDN e P2P, utilizzando la tecnica di codifica a descrizioni multiple congiuntamente ad una classificazione dei singoli pacchetti basata sulla Teoria dei Giochi. Nel corso di questo lavoro è stato possibile apprezzare la versatilità dello standard H.264/SVC che ha permesso di creare un bitstream scalabile rispondendo in modo ottimale ai vincoli del progetto. Si ricorda che nello standard precedente H.264/AVC la scalabilità sia di qualità che spaziale non è possibile e questo rende più laboriosa la creazione di eventuali descrizioni multiple. Inoltre sfruttando la predizione inter-layer messa a disposizione da H.264/SVC è stato possibile far fronte ai vincoli di memoria e di tempo computazionale che i servizi di video streaming richiedono. Successivamente, la tecnica MGS adottata per la creazione della scalabilità di qualità si è rivelata utile per la facilità con la quale è stato possibile realizzare le descrizioni della codifica MDC. Questo porta ad affermare che il flusso H.264/SVC si presta ottimamente alla realizzazione di descrizioni multiple mediante tecnica Unequal Error Protection.

Inoltre, dalle simulazioni effettuate si osserva come la codifica a descrizioni multiple fornisca un'elevata protezione al bitstream in caso di congestione del canale, essendo al tempo stesso adattabile alle condizioni di banda disponibili nella rete (inviando solo una parte delle descrizioni riducendo il bitrate totale). Infine si è notato come tale codifica sia perfettamente compatibile ad una trasmissione su reti P2P o CDN, incrementandone notevolmente le prestazioni rispetto ad un semplice flusso scalabile.

Infine è stata presentata la tecnica di classificazione delle descrizioni, basata sulla Teoria dei Giochi, applicata su diverse topologie di reti con diverse situazioni di congestione. Dai risultati ottenuti e qui esposti si osserva un netto miglioramento della

qualità ricevuta (rispetto agli altri due classificatori) nel caso in cui la congestione interessi un link condiviso da tutti gli utenti. Tuttavia, nel caso di congestione **solo** al singolo link le prestazioni della classificazione basata sulla Teoria dei Giochi possono essere ritenute uguali a quelle degli altri due metodi, in quanto gli scostamenti tra le varie curve sono nell'ordine dei 0.2 – 0.3 dB (arrivando nel caso peggiore a 1.2 dB), perciò assolutamente impercettibili all'occhio umano. Come esposto nel capitolo dei “Risultati Sperimentali”, tale peggioramento apportato dalla Teoria dei Giochi avviene perchè in quest'ultima configurazione i giocatori giocano in modo indipendente l'uno dall'altro, venendo meno all'ipotesi sulla quale si fonda la Teoria dei Giochi. Considerando reti P2P e CDN dove l'approccio di rete è quello di condividere le risorse (compresi i link) tra tutti gli utenti, avremo che il sistema di classificazione adottato si presta perfettamente a tale scopo con un notevole miglioramento delle prestazioni rispetto agli altri approcci di classificazione fissa.

Con l'obiettivo di migliorare le prestazioni finora trovate, si potrebbe osservare il cambiamento della qualità totale prendendo come riferimento altri tipi di equilibri, oltre a quello di Nash sperimentato in questo lavoro. Inoltre, si potrebbe testare la classificazione presa in esame su altre tipologie di rete per trovare la configurazione ottimale sulla quale far lavorare le classificazioni basate sulla Teoria dei Giochi.

Infine, visto il grande sviluppo che sta interessando le applicazioni 3D, tecniche di Unequal Error Protection potrebbero essere impiegate per migliorarne le prestazioni in caso di trasmissione su canali con perdite.

Ringraziamenti

Ringrazio i miei genitori Marino e Nadia per avermi supportato ma soprattutto sopportato in ogni momento di questo percorso di studi.

Un ringraziamento particolare va al Prof. Giancarlo Calvagno e al Dott. Milani Simone che con l'enorme esperienze, professionalità e disponibilità mi hanno fornito tutti gli elementi per svolgere al meglio l'intero lavoro di tesi.

Infine ringrazio tutti i miei amici per aver condiviso ogni esperienza fin qui vissuta.

Gaggio Marco.

Bibliografia

- [1] F. Kozamernik, "Media streaming over Internet - an overview of delivery technologies," EDU Technical Review No. 292, pp. 1–15, Oct. 2002.
- [2] Borella, M. S., D. Swider, S. Uludag, and G. B. Brewster, "Internet packet loss: Measurement and Implication for End-to-End QoS," Proceedings of International Conference on Parallel Processing, 1998.
- [3] J. Apostolopoulos and T. Wong and W. Tan and S. Wee and D. Tan, "On Multiple Description Streaming with Content Delivery Networks," in Proc. of IEEE INFOCOM 2002, (New York, USA), pp. 1736 – 1745, 23 - 27 Jun. 2002.
- [4] S. Khan, "Multiple Description streaming in Peer-to-Peer and Content Delivery Networks," in Proc. of the 2003 Australian Telecommunications, Networks and Application Conference (ATNAC 2003), (Melbourne, Australia), 8 - 10 Dec. 2003.
- [5] L. Qui, V. Padmanabhan, and G. Voelker, "On the placement of web server replicas," INFOCOM, vol. 3, pp. 1587–1596, 2001.
- [6] N. Magharei and R. Rejaie and Y. Guo, "Mesh or Multiple Tree: A Comparative Study of Live P2P Streaming Approaches," in Proc. of IEEE INFOCOM 2007, (Anchorage, Alaska, USA), pp. 1424 – 1432, 6 - 12 May 2007.
- [7] "BitTorrent." Online available <http://www.bittorrent.com/>.
- [8] Weishuai Yang and Nael Abu-Ghazaleh, "GPS: A General Peer-to-Peer Simulator and its Use for Modeling BitTorrent," Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2005. 13th IEEE International Symposium, pp. 425–432, 27-29 Sept. 2005.

- [9] “BBC iPlayer.” Online available www.bbc.co.uk/iplayer/.
- [10] Antony Rose, “BBC iPlayer,” EBU Technical Review, 2008.
- [11] “Swarmplayer – Online video using BitTorrent and HTML5 applied to Wikipedia.” Online available <http://swarmplayer.p2p-next.org/>, 2010.
- [12] “Peercast.” Online available <http://www.peercast.org/>.
- [13] Bertrand Mathieu, Pierre Paris, “A Topology-Aware P2P Vide Streaming System,” Information Infrastructure Symposium, 2009. GIIS '09. Global, pp. 1–8, 23-26 Jun. 2009.
- [14] Haiyong Xie, Arvind Krishnamurthy, Avi Silberschatz, Y. Richard Yang, “P4P: Explicit Communications for Cooperative Control Between P2P and Network Providers,” Distributed Computing Industry Association.
- [15] I. Rimac, V. Hilt, M. Tomsu, M. Tomsu, E. Marocco, “A Survey on Research on the Application-Layer Traffic Optimization (ALTO) Problem,” IRTF - Rfc:6029, 2010.
- [16] Jan Seedorf, Sebastian Kiesel, Martin Stiernerling, “Traffic Localization for P2P-Applications: The ALTO Approach,” Peer-to-Peer Computing, 2009. P2P '09. IEEE Ninth International Conference, pp. 171–177, 9-11 Sept. 2009.
- [17] “Pando Networks.” Online available <http://www.pandonetworks.com/>.
- [18] Meng-Ting Luac, Hung Nienac, Jui-Chieh Wub, Kuan-Jen Pengac, Polly Huan-gade, Jason J. Yaoac, Chih-Chun Laif and Homer H. Chenacd, “A Scalable Peer-to-Peer IPTV System,” Consumer Communications and Networking Conference, 2007. CCNC 2007. 4th IEEE, pp. 313 – 317, 2007.
- [19] M. Zink and A. Mauthe, “P2P Streaming using Multiple Description Coded Video,” in Proc. of the 30th EUROMICRO Conference (EUROMICRO 2004), (Rennes, France), pp. 240 – 247, 31 Aug. - 3 Sept. 2004.
- [20] Yago Sánchez, Cornelius Hellge, Thomas Schierl, Thomas Wiegand, “P2P group communication with layer-aware FEC,” ACM Multimedia 2010 - International Conference, 2010.

- [21] Rohit Puri and Kannan Ramchandran, "Multiple description source coding using forward error correction codes," Signals, Systems, and Computers, 1999. Conference Record of the Thirty-Third Asilomar Conference, pp. 1 – 5, 1999.
- [22] S. Milani and G. Calvagno, "A Game Theory Based Classification for Distributed Downloading of Multiple Description Coded Videos," in Proc. of the 2009 IEEE International Conference on Image Processing (ICIP 2009), (Cairo, Egypt), 24 - 28 Nov. 2009.
- [23] S. Milani and G. Calvagno, "A Game Theory Based Classification for Distributed Downloading of Multiple Description Coded Videos," in Proc. of the IEEE ICIP 2009, (Cairo, Egypt), 24 - 28 Nov. 2009.
- [24] R. Bernardini, M. Durigon, R. Rinaldo, P. Zontone, "Real-time multiple description video streaming over QoS-Based wireless networks," IEEE, International Conference on Image Processing, 2007.
- [25] X. Wu and B. Ma and N. Sarshar, "Rainbow network problems and multiple description coding," Information Theory, 2005. ISIT 2005. Proceedings. International Symposium, vol. 54, pp. 4565 – 4574, Oct. 2008.
- [26] V. K. Goyal, "Multiple Description Coding: Compression Meets the Network," in IEEE Signal Processing Magazine, pp. 74 – 93, Sept. 2001.
- [27] Andrea Vitali, "Multiple Description Coding - A new technology for video streaming over the Internet," EBU TECHNICAL REVIEW, Oct. 2007.
- [28] J. R. Taal and R. L. Lagendijk, "Hybrid Temporal-SNR Multiple Description Coding for Peer-to-Peer Television," in Proc. of 2006 Picture Coding Symposium (PCS 2006), (Beijing, China), 24 - 26 Apr. 2006.
- [29] J. R. Taal and R. L. Lagendijk, "Scalable Multiple Description Coding for Video Distribution In P2P Network," in Picture Coding Symposium, Dec. 2004.
- [30] John C. Harsanyi, "A General Theory of Rational Behavior in Game Situation," Econometrica, vol. 34, pp. 613–634, Jul. 1966.
- [31] Michael A. Goodrich, "Proof of the Minimax Theorem," , pp. 1–8, 17 Sept. 2007.
- [32] John F. Nash, Jr., "The bargaining problem," Econometrica, vol. 18, pp. 155–162, Apr. 1950.

- [33] J. Heinanen, R. Guerin, "A Single Rate Three Color Marker," Rfc:2697, Sept. 1999.
- [34] J. Heinanen, R. Guerin, "A Two Rate Three Color Marker," Rfc:2698, Sept. 1999.
- [35] Amir Said, William A. Pearlman, "A New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees," Circuits and Systems for Video Technology, IEEE Transactions, vol. 6, pp. 243–250, Jun. 1996.
- [36] "VQM." http://compression.ru/video/quality_measure/video_measurement_tool_en.html.
- [37] Andrew B. Watson, James Hu, John F. McGowan III, "Digital video quality metric based on human vision," Journal of Electronic Imaging, pp. 20–29, 2001.
- [38] Stephen Alstrup and Theis Rahue, "A new technology for large-scale streaming over the internet," EBU Technical Review, 2005.
- [39] E. Akyol and A. Murat Tekalp and M. Reha Civanlar, "A Flexible Multiple Description Coding Framework for Adaptive Peer-to-Peer Video Streaming," IEEE J. Select. Topics Signal Processing, vol. 1, pp. 231 – 245, Aug. 2007.
- [40] S. Khan, "Multiple Description streaming in Peer-to-Peer and Content Delivery Networks," in Proc. of ATNAC 2003, (Melbourne, Australia), 8 - 10 Dec. 2003.
- [41] Simone Zezza and Enrico Magli and Gabriella Olmo and Marco Grangetto, "SEACAST: a protocol for peer-to-peer video streaming supporting multiple description coding," in Proc. of the 2009 IEEE International Conference on Multimedia and Expo (ICME 2009), (New York, USA), 28 Jun. - 3 Jul. 2009.
- [42] Simone Zezza and Enrico Magli and Gabriella Olmo and Marco Grangetto, "SEACAST: a protocol for peer-to-peer video streaming supporting multiple description coding," in Proc. of ICME 2009, (New York, USA), 28 Jun. - 3 Jul. 2009.
- [43] M. J. Osborne, An Introduction to Game Theory. Oxford University Press, 2004.
- [44] D. A. Smith and C. Shi and R. A. Berry and M. L. Honig and W. Utschick, "Distributed Resource Allocation Scheme," Signal Processing Magazine, IEEE, vol. 26, pp. 53 – 63, Sept. 2009.
- [45] Vito Fragnelli, "Teoria dei Giochi," 2009 - 2010.

- [46] Fioravante Patrone, “Breve introduzione sulla Teoria dei Giochi,”
- [47] Heiko Schwarz, Detlev Marpe, Thomas Wiegand, “Overview of the Scalable Video Coding Extension of the H.264/AVC Standard,” Circuits and Systems for Video Technology, IEEE Transactions, vol. 17, pp. 1103–1120, Sept. 2007.
- [48] Mark Baker, Rahim Lakhoo, “Peer-to-Peer Simulators,” AMG Technical Report 2007.03, Apr. 2007.
- [49] Kevin Werbach, “The Implication of Video P2P on Network Usage,” DRAFT, 19 Nov. 2004.
- [50] Khan, S.; Schollmeier, R.; Steinbach, E.; , “A performance comparison of multiple description video streaming in peer-to-peer and content delivery networks,” Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference, vol. 1, pp. 503 – 506, 30 Jun. 2004.
- [51] J.G. Apostolopoulos, W. Tan, and S.J. Wee, “Performance of a Multiple Description Streaming Media Content Delivery Network,” IEEE International Conference on Image Processing (ICIP), Sept. 2002.
- [52] Peter Piedad, Jeremy Ethridge, Mandeep Baines, Farhan Shallwani, A Network Simulator Differentiated Services Implementation. Open IP, Nortel Networks, 26 Jul., 2000.