dispense del corso di

ELABORAZIONE E TRASMISSIONE DI INFORMAZIONI MULTIMEDIALI

raccolte durante il corso tenuto da

Juan Carlos De Martin

Angelo Raffaele Meo

Politecnico di Torino

Introduzione

Analogico e digitale

La comunicazione **analogica** è utilizzata da tutti gli strumenti che trasmettono una grandezza il cui valore varia con continuità e che rispecchia un'analoga variazione continua della grandezza che si vuole rappresentare.
I segnali analogici, dunque, consistono in una tensione elettrica che segue nel tempo l'andamento del segnale originale. Nei segnali audio la tensione elettrica è molto simile all'andamento dell'onda sonora originale (molto simile, non identica, perché vi è sempre l'introduzione di una quota di distorsione e di rumore, non presenti nel segnale originale).

Nel caso dei segnali **digitali** invece, il segnale viene rappresentato da una serie di numeri, ciascuno dei quali indica il valore della pressione istantanea in un dato istante.
I dispositivi basati sul sistema di comunicazione digitale rappresentano, per mezzo di un codice in cifre, i valori delle grandezze da trasmettere, anche se queste variano con continuità. Utilizzano quindi la campionatura di tali grandezze a successivi intervalli di tempo, molto ravvicinati, e la codifica dei valori campionati in un sistema numerico prefissato.

Perché il digitale

Benché la natura umana sia meglio disposta a relazionarsi con il formato analogico dei segnali, motivi di economicità e rapidità della comunicazione, accompagnati da un rapido e intenso progresso tecnologico nel campo dell'elettronica, hanno portato nell'ultimo decennio la forma digitale di comunicazione a prendere il sopravvento su quella analogica.
Già all'epoca della II Guerra mondiale erano in corso sperimentazioni sull'audio digitale, con conversioni di onde sonore analogiche in valori *discreti*. Questi studi furono portati a termine "campionando" l'onda sonora molte volte al secondo. Ogni campione registrava l'ampiezza dell'onda in quel punto.

Il processo di conversione

Il processo di conversione da analogico a digitale inizia con l'ingresso di segnali audio analogici. L'intensità del segnale è misurata a intervalli di tempo discreti, ma abbastanza ravvicinati da permettere la ricostruzione fedele del segnale: il numero di volte in cui un segnale audio in ingresso è misurato in un determinato periodo di tempo è definito **frequenza di campionamento (***sample rate* o *sample frequency*).

Il teorema di Shannon

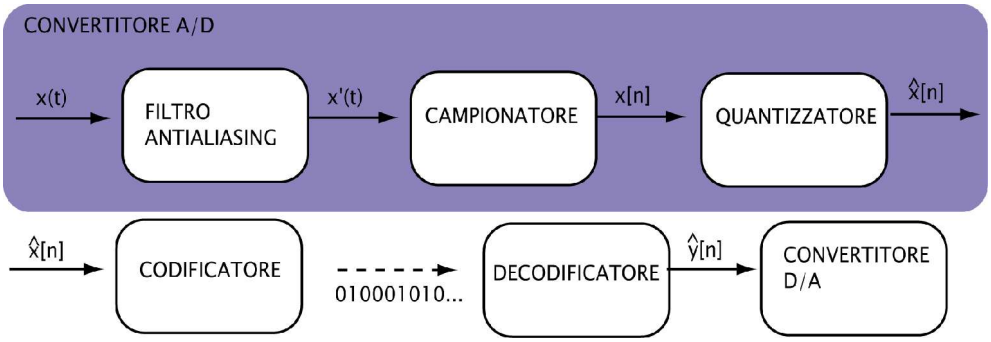
In base al **teorema di Claude Shannon** è possibile campionare un segnale analogico senza perdere informazione a patto che:

- Il segnale sia limitato in banda
- La frequenza di campionamento sia il doppio della massima frequenza del segnale.
- L'interpolazione venga eseguita con un filtro passa-basso ideale.
- I campioni abbiano precisione infinita.

Quantizzazione

Per risolvere il problema di avere un filtro ideale (con pendenza infinita) è possibile sovracampionare il segnale (oversampling) in modo da poter utilizzare un filtro reale.
Rimane il problema che i campioni non possono avere precisione infinita ma costituiscono un insieme *discreto*. Al contrario i segnali analogici non sono discreti, ma *continui*. Al momento di misurare l'ampiezza di ogni gradino (**quantizzazione**) questa dovrà essere approssimata ad un intero, commettendo così un errore più o meno grande. La quantizzazione implica di conseguenza una perdita di informazione la quale provoca nel segnale una distorsione ed è quindi un processo irreversibile.

Schema del processo di campionamento



Sample rate per l'audio

Negli anni '70, quando Philips e Sony iniziarono a cercare un modo di migliorare la qualità audio della musica registrata, si rivolsero al campionamento digitale. Fu scelto un sample rate di 44.100 campioni per secondo (**44.1 kHz**) sia perché era superiore all'obiettivo fissato e cioè superiore ai 40 kHz (che rappresenta il doppio della massima frequenza, 20 kHz, percepibile dall'orecchio umano), sia perché rappresentava il massimo di informazioni che potevano essere immagazzinate su nastro.

Voce

Lossy e lossless

L'obbiettivo delle tecniche di compressione è ovviamente quello di ridurre lo spazio necessario ad immagazzinare determinati dati o la banda necessaria per trasmetterli. Una prima classificazione delle tecniche di compressione distingue tra tecniche che mantengono perfettamente inalterate le informazioni dopo la compressione (tecniche lossless cioè senza perdita) e tecniche che prevedono un certo degrado delle informazioni (lossy). E' abbastanza ovvio che nel comprimere informazioni come testi, documenti o programmi non ci si possa permettere la perdita di nessun bit di informazione, percui dovremo utilizzare necessariamente tecniche lossless come quelle adottate dallo Zip. Nel caso dell' audio, delle immagini e dei filmati, un certo livello di degradazione è un compromesso accettabile per ridurre (e di molto) l'occupazione o la banda richiesta dal file. Le tecniche di compressione audio che analizzeremo sono infatti tutte di tipo lossy.

Cosa si vuole preservare

Parlando di segnale vocale a valle della compressione intendiamo preservare:

- **intelligibilità** (fondamentale)
- **naturalezza**

In particolare per naturalezza intendiamo

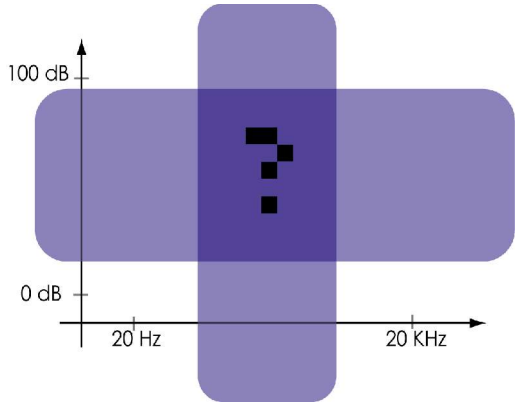
- identificazione del parlatore
- stato d'animo del parlatore
- background noise

Per valutare l'intelligibilità si utilizza un test soggettivo chiamato *Diagnostic Rhyme Test*.

La parte più significativa dello spettro del segnale vocale si trova alle medio-basse frequenze come anche gran parte della sua energia. Si è scelto quindi di campionare il segnale vocale mantenendo solamente lo spettro compreso nell'intervallo di frequenza di 300÷3400 *Hz,* limitando così l'impatto sull'intelligibilità della voce. La frequenza di campionamento *Fc,* per il teorema del campionamento di Shannon, sarà più del doppio della massima frequenza del segnale ossia pari a 8000 *Hz.*
Per quanto concerne la dinamica del segnale in ambito telefonico si è scelto un range di *60 dB* con un *n* pari a 12-13 bit per campione.
Otteniamo così un bit rate *R* pari a

$$R = Fc \times N = 8000 \times 12 = 96 \text{ kbit/s}$$

La scelta della banda e della dinamica



Quantizzazione uniforme o logaritmica

La quantizzazione del segnale può avvenire in maniera **uniforme** o **non uniforme**.
La quantizzazione uniforme è caratterizzata dal fatto che l'ampiezza degli intervalli è costante. Una tecnica di questo tipo è consigliabile quando il segnale da quantizzare ha una distribuzione uniforme all'interno del proprio range dinamico. Considerando un quantizzatore uniforme a N livelli l'errore introdotto dalla quantizzazione sarà

$$|e[n]| < \frac{\Delta}{2} \text{ dove } \Delta = \frac{2X_m}{2^N}$$

con $2X_m$ pari alla dinamica del segnale

Per calcolare il rapporto segnale rumore SNR conviene lavorare con le energie sia per comodità matematica sia poiché sembra rispecchiare il funzionamento della nostra percezione.

$$SNR = \frac{\sigma_X^2}{\sigma_e^2}$$

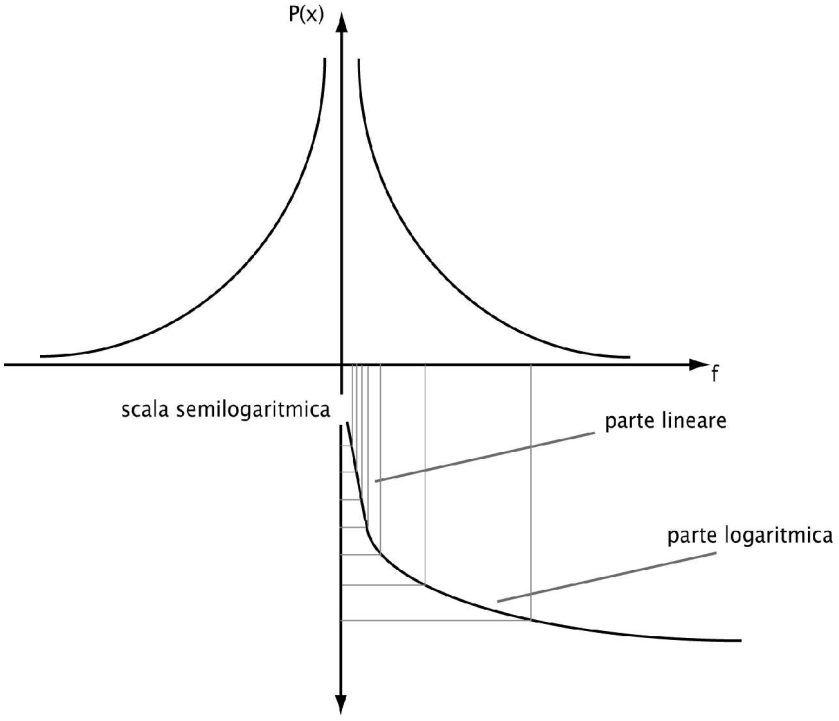
$$\sigma_e^2 = \sum_{-\infty}^{+\infty} e^2[n] = \int_{-\Delta/2}^{+\Delta/2} x^2 f_e(x)\,dx = \frac{1}{\Delta}\int_{-\Delta/2}^{+\Delta/2} x^2\,dx = \frac{1}{\Delta}\left[\frac{x^3}{3}\right]_{-\Delta/2}^{+\Delta/2} = \frac{\Delta^2}{12}$$

$$SNR = \frac{12 \cdot \sigma_X^2}{\Delta^2} = \frac{12 \cdot 2^N \sigma_X^2}{4 X_m^2} = 3 \cdot 2^{2N} \frac{\sigma_X^2}{X_m^2}$$

trasformando in dB otteniamo

$$SNR_{dB} = 10 \log_{10} 3 + 20 N \log_{10} 2 + 20 \log_{10} \frac{\sigma_X}{X_m} \simeq 4.15 + 6N + 20 \log_{10} \frac{\sigma_X}{X_m}$$

Quantizzazione logaritmica



Un'importante codifica audio dove il parlato è rappresentato mediante un numero fisso di campionamenti al secondo è quella **PCM** (*Pulse Code Modulation*). **PCM**

ITU-T G.711
64 kbit/s;
Fc = 8KHz;
N = 8 bit /campione ;
quantizzatore
logaritmico

La distribuzione ottima dei livelli del quantizzatore si può calcolare minimizzando l'errore quadratico medio.

$$\min E = E\left[e^2[n]\right] = \sigma_e^2$$

Se disponibile la p.d.f. di *x*[*n*] allora il quantizzatore ottimo è calcolabile in maniera analitica tramite al teorema di *Lloyd-Max.* Per trovare la p.d.f. si parte da quella sperimentale e si cerca la distribuzione analitica che meglio l'approssima (nel caso della voce telefonica quella di *Laplace* o *Gamma*)

DPCM

La **Differential Code Pulse Modulation** è in sostanza una PCM modificata. Poichè la voce umana emette suoni che non passano da volumi bassi a volumi alti o da frequenze basse a frequenze alte in un tempo inferiore ad alcune decine di millisecondi (corrispondenti quindi a centinaia di campioni registrati), posso pensare di trasmettere non tanto il campione attuale, ma la differenza fra il campione passato e quello attuale. Ad esempio se il campione N ha ampiezza pari a 100, il campione N+1 avrà ampiezza pari a 100, 101 o 99. Non potrà avere ampiezza pari a 200 perchè fra un campione e l'altro ci passano solo poche centinaia di microsecondi e la gola e le corde vocali hanno un tempo di rilassamento molto maggiore. Quindi, anzichè trasmettere il valore 101 del campione N+1, trasmetto solo il valore +1. Il decoder che riceve l'informazione vede quanto è il valore di N (es: 100), legge poi che il valore del campione N+1 è pari a quello di N cui va sommato 1, e quindi assume che il valore del campione N+1 è 101.
E' ovvio che per utilizzare questa tecnica è necessario che ci sia una forte correlazione tra un campione e il suo successivo: nel caso della voce il coefficiente di correlazione $\rho \simeq 0.9$.
E' possibile inoltre sfruttare la correlazione non soltanto con il campione precedente ma con *N* campioni precedenti (quantizzatore di ordine *N*)

Esempio di DPCM del 1° ordine

Tx $d[n] = x[n] - x[n-1]$ Rx $\hat{x}[n] = d[n] + \hat{x}[n-1]$

$$d[n] = x[n] - \alpha \hat{x}[n-1]$$

$$\min E\left[d^2[n]\right] = \min E\left[x^2[n] + \alpha^2 \hat{x^2}[n-1] - 2\alpha x[n]\hat{x}[n-1]\right]$$

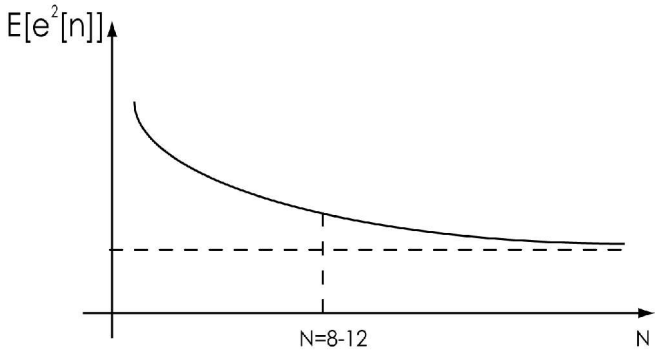
$$\frac{\partial E}{\partial \alpha} = 2\alpha \hat{x^2}[n-1] - 2x[n]\hat{x}[n-1]$$

e ponendolo uguale a 0 troviamo $\alpha_{OTT} = \rho\left(x[n], \hat{x}[n-1]\right)$

Per un generico DPCM di ordine N abbiamo:

$$d[n] = x[n] - \sum_{k=1}^{n} \alpha_k \hat{x}[n-k]$$

La scelta dell'N



Si sceglie l'*N* minore tale per cui per *N* maggiori non vi sia una sostanziale diminuzione dell'errore quadratico medio σ_e^2. Solitamente *N*=8-12 (valore tipico 10 - correlazione a breve termine).
Gli α_k ottimi si calcolano quindi attraverso un sistema di *N* derivate.

APCM

La voce é un processo fortemente non stazionario; è quindi lecito pensare di adattare il fondoscala del quantizzatore al segnale momento per momento. Si arriva in questo modo alle tecniche adatattive.
Per sfruttare tali tecniche occorre identificare una finestra temporale sufficientemente piccola in cui il segnale si possa considerare stazionario (tecniche a frame).
In media per la voce si scelgono finestre di durata compresa tra 10-25 ms (valore tipico 20 ms).
Queste tecniche si dividono principalemente in due sottoinsiemi:

1.
2. *- FeedForward APCM*
3. I passi seguiti da questa tecnica sono:
   - stima Ei (energia per frame i-esima)
   - Determinazione Xm,i
   - Codifica dei campioni nel frame.

In questo modo ho solo un miglioramento del rapporto segnale rumore. Se mi interessa invece costruire delle versioni a bitrate inferiore occorre tenere costante il rapporto SNR e determinare N al punto due.

$$E_i = \frac{1}{M} \sum_{n=-\frac{M}{2}}^{\frac{M}{2}} x^2[n]$$

Con questa tecnica occorre inviare il fondoscala ogni X ms e ciò genera un overhead.

- Feedback APCM
Utilizzo i campioni passati per stimare la variazione del fondoscala. Lo svantaggio è che se vi sono variazioni brusche perdo il segnale d'altra parte ho il vantaggio di poter aggiornare più frequentemente (al limite una volta per campione) il fondoscala.

ADPCM

Si possono anche avere tecniche miste dette **Adaptative Differential PCM**, in cui cioè trasmetto sempre i bit differenza, ma tenendo conto della "storia" dei bit passati. Il meccanismo è molto più complesso poichè si cerca di capire dove potrà andare l'onda cioè quali saranno i suoi campioni futuri sulla base della storia di quelli passati. Il principio è comunque lo stesso, cioè trasmettere l'informazione collegata alle differenze fra i campioni, anzichè i valori effettivi con un numero di bit che dipende dalle caratterisiche del segnale nella porzione in esame (da cui il nome "adattivo"). Questa tecnica raggiunge un ratio di **compressione di 1:2** rispetto all'originale non compresso quindi ancora insufficiente per i nostri scopi.

ITU-T G.726 (1986)
32 kbit/s;
Fc = 8KHz;
N = 4 bit /campione ;

[Utilizzato per cordless DECT]

Vocoder parametrici

Per riuscire a superare i limiti delle tecniche *PCM si è cercato di affrontare il problema in maniera completamente diversa studiando lo “strumento” che produce il segnale in questione:si è arrivato in questo modo ai **vocoder parametrici** costruiti cercando di modellizzare l’apparato di fonazione umano.
Occorre quindi, prima di passare a una più accurata descrizione dei sistemi a codifica di parametri, fare un cenno al modello di generazione del segnale vocale.

L’apparato di fonazione

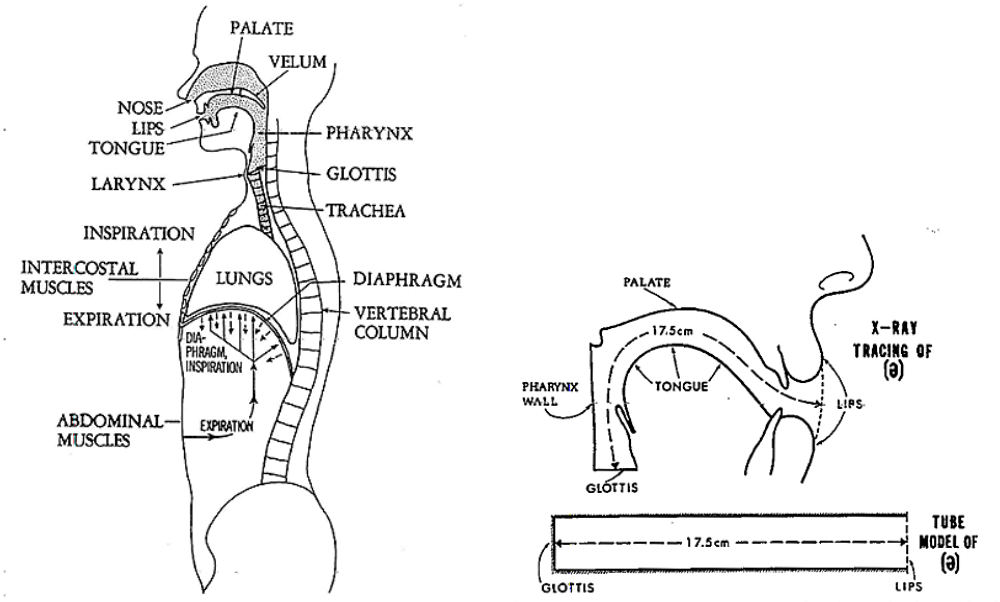
Gli organi di fonazione comprendono:

- diaframma : muscolo cupoliforme che si estende all'estremità inferiore della cassa toracica
- polmoni
- trachea
- laringe e corde vocali
- cavità faringea : gola
- cavità orale
- cavità nasale

A seconda della loro posizione vengono prodotti diversi tipi di suoni. Brevemente si può dire che, se le corde vocali non sono tese e vi è un’improvvisa apertura lungo il tratto vocale vengono emessi suoni sordi o

non vocalizzati *(unvoiced)*; al contrario se le corde vocali sono tese l'aria non riesce a fuoriuscire se non quando l'accresciuta pressione prodotta dai polmoni non riesca a vincere l'azione di chiusura dei muscoli che tendono le corde vocali. In questo caso l'aria fuoriesce a sbuffi e la pressione acustica a valle delle corde vocali ha un andamento periodico. La formante fondamentale, che varia in genere tra gli 50 e i 250 Hz nell'uomo e tra i 120 e i 500 Hz nella donna è detta **frequenza di *pitch*** *Fo*, intendendo con la parola *pitch* il *tono* della voce. I suoni emessi in questo caso vengono detti vocalizzati *(voiced)*.

Apparato di fonazione



I picchi spettrali sono localizzati a frequenze dette *formanti* o *risonanze spettrali* (da 3 a 5 se si considera la tipica banda telefonica, 300-3400 Hz) e sono le frequenze di risonanza del tratto vocale associate quindi alla sua forma e dimensione.

Il modello

Lo schema utilizzato per modellizare il processo di creazione della voce è il seguente:

Organo	**Segnale**	**Parametro**

Polmoni	Rumore	Gain (0-60 dB)
Laringe e corde vocali	Periodico (voiced) Rumore (unvoiced)	Flag V/UV F0 = 1/T (frequenza fondamentale)
Tratto Vocale	Vengono enfatizzate alcune componenti in frequenza	Coefficienti del filtro che genera l'inviluppo

Modellizzare il tratto della laringe e delle corde vocali è fondamentale se intendiamo preservare la naturalezza in quanto ci permette di identificare l'età e il sesso del parlatore (pitch) nonché proprio la sua identità tramite i pattern di frequenza utlizzati (prosodia). Se non modellizzassimo questa parte l'intellegibilità verrebbe comunque mantenuta poiché compito del tratto vocale. Riguardo quest' ultimo occorre modellizzarlo con un filtro almeno del VI ordine (3 picchi dell'inviluppo). Di solito si utlizza un filtro di ordine 10.
Per ottenere un parlato naturale occorre che i parametri del nostro modello siano aggiornati ogni 10-20 ms che è la durata media di un fonema.

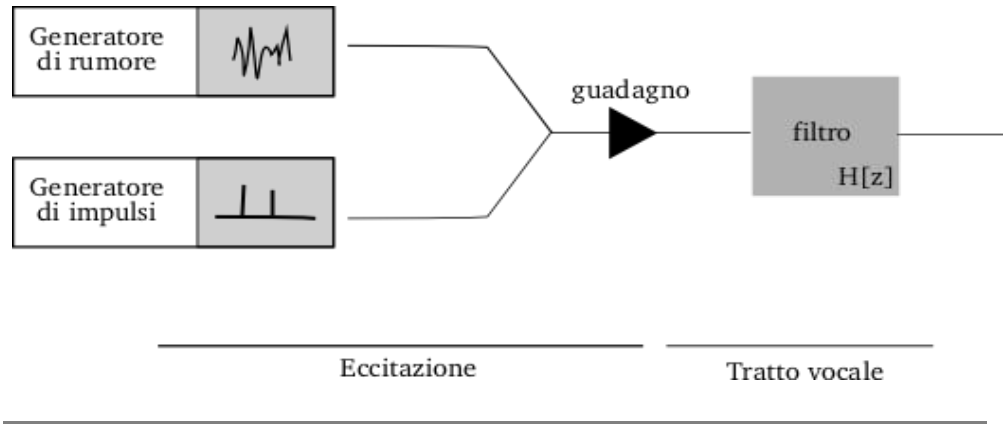
In conclusione:

- **Gain** : 5bit quantizzati mediante quantizzatore non uniforme ottimo percettivamente trasparente
- **Flag V/UV** : 1 bit
- **Pitch**: 7 bit
- **10 coefficienti del filtro** : 45 bit (4/5 bit ciascuno).

per un totale di 58 bit inviati ogni 20 ms .
Il bitrate risultante sarà quindi di 2.9 kbit/sec notevolmente inferiore a quello della migliore tecnica *PCM (32kbit/s).

LPC

Si parla in questo caso di tecniche LPC (Linear Prediction Coding) in cui i parametri del filtro inviati (che rappresentano l'inviluppo del segnale) sono gli stessi parametri che minimizzavano il segnale differenza nelle tecniche DPCM.



Purtroppo con un modello simile si ottengono dei buoni risultati per quanto riguarda l'intelligibilità ma dei risultati abbastanza scarsi per quanto riguarda la naturalezza (varia a seconda del parlatore) ed è quindi una modalità non sufficiente per il servizio telefonico che vorrebbe garantire anche una buona naturalezza.

Come si determina la qualità?
Per misurare la qualità di un segnale multimediale vi sono due categorie di test possibili:
test oggettivi: $D = f(\hat{x}[n], x_{ref}[n])$
test soggettivi: test statistici di gradimento da parte di soggetti umani. Occorre avere un buon numero di soggetti e sottoporli ad un adeguato numero di stimoli per avere una *confidence* opportuna.

Per la voce:
rapporto segnale rumore (test oggettivo)

$$SNR = \frac{\sum x^2[n]}{\sum (x[n] - \hat{x}[n])^2}$$

(funziona solo per tecniche PCM)
Standard ITU, PESQ(*Perception Evaluation of Speech Quality*)
Mean Opinion Score (test soggettivo a 5 livelli)

MELP e CELP

Per superare i difetti del modello descritto in precedenza senza voler cambiare il tipo di approccio che aveva dato comunque dei buoni risultati di bitarate si è cercato di capire quali fossero le principali parti del modello che non simulavano in maniera soddisfacente l'apparato di fonazione.

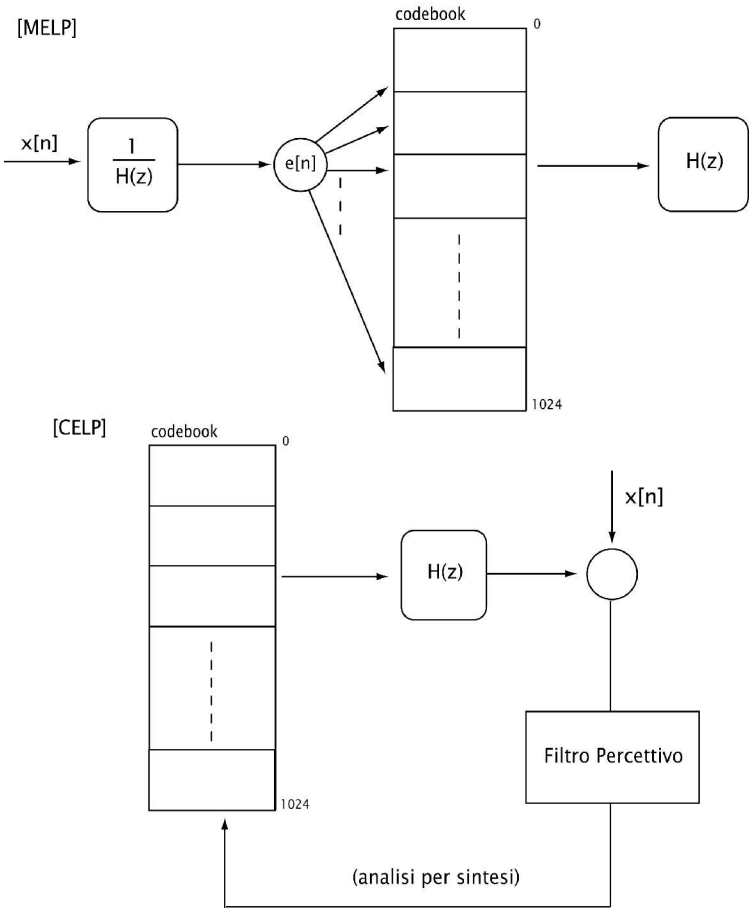
I principali difetti riscontrati sono:
- classificazione binaria V/UV troppo rigida
- spettri perfettamente periodici quando nella realtà sono più irregolari

Per far fronte a questi problemi si è pensato di utilizzare la **quantizzazione vettoriale**: avere cioè a disposizione un "catalogo" di forme d'onda e di scegliere di volta in volta quella più simile al segnale di eccitazione del fonema corrente.
Vi sono due modi per eseguire questa scelta:

Quantizzazione vettoriale semplice del segnale di eccitazione

Analisi per sintesi: provo tutte le forme d'onda e scelgo quella che tramite un filtro percettivo mi da a valle un risultato migliore

Nel primo caso abbiamo la tecnica MELP (*Mixed-Excitation Linear Prediction*) nel secondo quella CELP (*Code[book] – excited Linear Prediction*).



Alcuni standard

-Standard ITU

ITU G.726	32 kb/s (ADPCM)
ITU G.728	16 kb/s (LP-CELP)
ITU G.729	8 kb/s (CS-ACELP)
ITU G.723.1	3-6 kb/s
ITU G.4k	4 kb/s

-Standard GSM

	bitrate	**1. MOS**
GSM-FR (full rate)	13 kb/s	3.5
GSM-EFR (enanched full rate)	12.2 kb/s	4.0
GSM-AMR(adaptative multimode)	4.75-12.2 kb/s	4.0

Il GSM-AMR permette di variare il partizionamento della banda tra il codice di correzione degli errori e il segnale vero e proprio (voce).

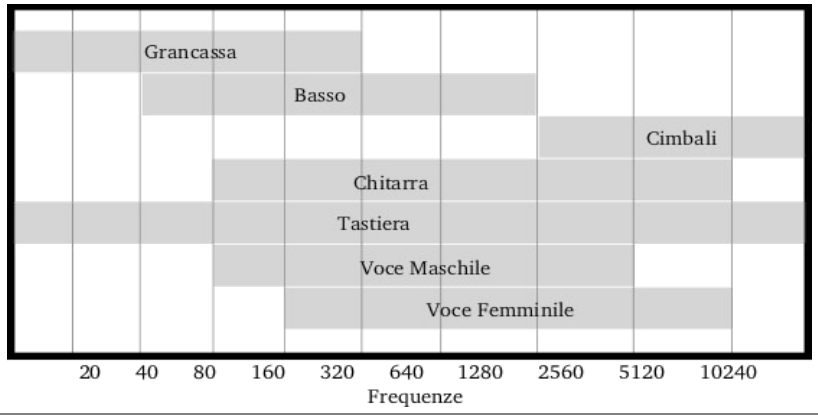
Audio

Per “Audio” intendiamo qualunque cosa sia udibile.

Nel caso di un CD-AUDIO abbiamo:

CD-AUDIO
Fc = 44100 Hz;
N = 16 bit /campione
2 canali (stereo)
Codifica PCM
SNR ≈100dB

Tale standard che riproduce piuttosto fedelmente il segnale audio possiede però un un bit-rate complessivo pari a : 44100 * 16 *2= 176400 bytes/secondo.
È facilmente calcolabile che un brano di 5 minuti trattato in questo modo produce un file di una cinquantina di Megabyte (10MByte/minuto). L'aumento progressivo di potenza dei calcolatori ha però favorito negli ultimi decenni lo sviluppo e la diffusione dei sistemi di compressione dei dati audio che permettono, come vedremo, una forte riduzione delle dimensioni dei file.
I modelli precedentemente visti per la codifica della voce basati sulla “produzione” sono una strada non percorribile per la codifica audio.
Infatti osservando il seguente grafico si capisce subito che la voce umana occupa solo certe frequenze mentre gli strumenti musicali spaziano secondo range diversi a seconda del tipo di strumento impiegato.

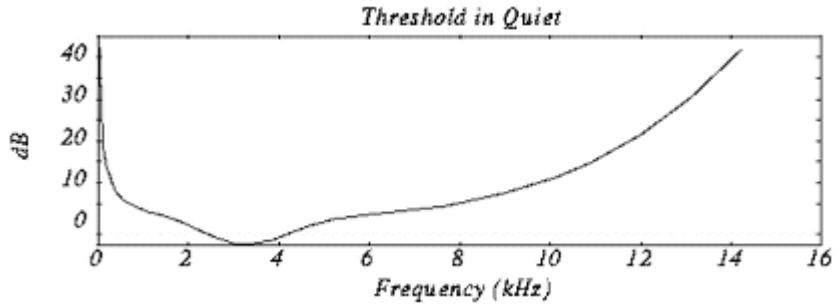


Detto questo, appare chiaro che tentare di comprimere una musica generica utilizzando un approccio basato su modelli risulterebbe estremamente complesso. Limitarsi alla sola voce è "facile" ma prevedere un modello per ogni strumento è, allo stato attuale della tecnologia troppo complesso. La soluzione è quindi porre l'attenzione non sulla sorgente ma sulla "destinazione" dei suoni. In ogni caso la musica dovrà essere "ascoltata da un orecchio" quindi conoscere fino in fondo quello che un uomo medio riesce a sentire o non riesce a sentire può rivelarsi sorprendentemente utile.

Approccio Psico-Acustico alla compressione

Il nostro orecchio infatti non è perfetto e questo è un grande vantaggio. In prima analisi esso è sensibile in misura diversa alle diverse frequenze, come è possibile dedurre esaminando il grafico in basso che esprime l'andamento in frequenza della **soglia di udibilità**.

Sensibilità dell'orecchio umano alle diverse frequenze



Come si evince dal grafico l'orecchio umano è maggiormente sensibile alle frequenze comprese fra 2 e 4 KHz, che richiedono pochissimi dB per essere percepite mentre servono molti più dB per i toni molto bassi e quelli alti. In generale, siccome l'orecchio a queste frequenze perde sensibilità e selettività, si può ridurre la quantità di informazione trasmessa in questa parte di spettro.

Un altro aspetto fondamentale della psico-acustica è il fenomeno del **mascheramento** secondo il quale un segnale detto "mascherante" rende non percepibile un altro detto "mascherato".
Esistono due tipi di mascheramento: quello in frequenza e quello temporale.

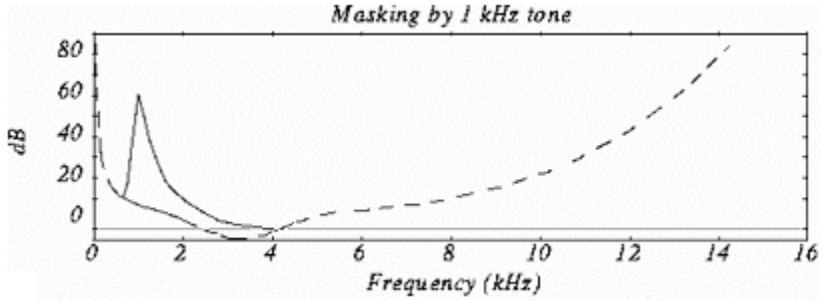
Mascheramento In frequenza

Nel mascheramento in frequenza una frequenza 'debole' può essere benissimo mascherata, cioè risultare inudibile, da una frequenza anche lontana qualche centinaio di Hz, se quest'ultima è bella forte, cioè con una intensità alta.

A livello sperimentale se abbiamo ad esempio un tono alla frequenza di 1KHz (tono maschera) tenuto a 60 dB abbiamo che un secondo tono, detto tono test verrà "coperto" dal primo in alcune circostanze: avvicinandoci sia da sinistra che da destra al tono maschera, dobbiamo alzare il volume del tono test per riuscire a distinguerlo. Oltre i 4 Khz e al di sotto degli 0.5 le

cose tornano a posto, però notiamo che nell'intorno di 1KHz i due toni sono praticamente indistinguibili a meno di non alzare pesantemente il volume del tono test.

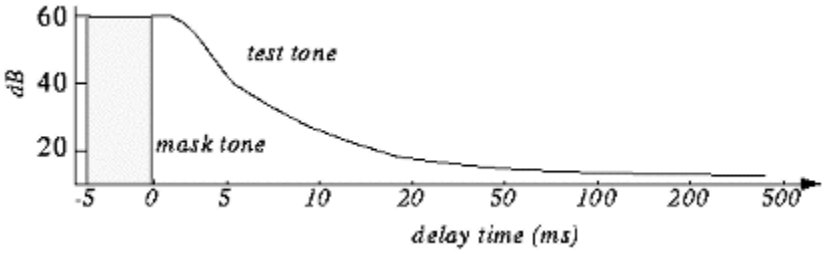
Mascheramento in frequenza



Mascheramento temporale

Esiste però un altro tipo di mascheramento, ed è quello temporale. Supponiamo di avere due toni, uno forte e l'altro, che gli è vicino in frequenza, piuttosto debole. Dall'analisi vista prima sappiamo già che il nostro orecchio sente solo il tono più forte che si comporta così da tono maschera. Ora, se improvvisamente questo tono maschera cessa di esistere, impieghiamo un po' di tempo per avvertirlo perché la membrana del nostro timpano deve assestarsi. Il tempo che impieghiamo dipende dal volume del tono maschera e da quello del tono test. Se il tono maschera ci assorda, il nostro orecchio impiegherà un bel po' prima di riuscire a sentire il tono più debole anche dopo che il tono forte è morto.

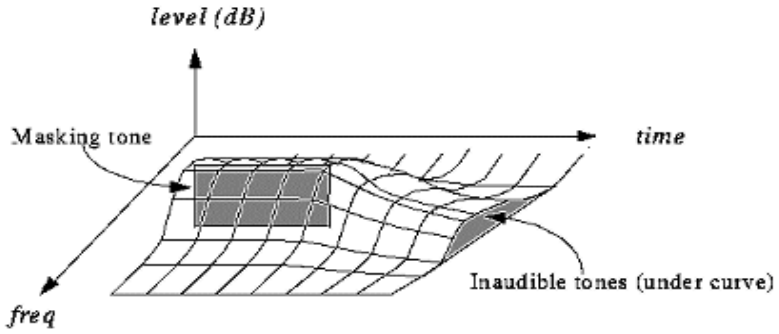
Mascheramento temporale



Il tono test, di 1KHz, viene disattivato all'istante zero: esso manteneva un valore fisso di 60dB. Se il nostro test tone ha un'ampiezza di una quarantina di dB, bastano 5 ms per avvertirlo; se è di soli 20dB, ne occorrono quasi 20 di ms perché risulti avvertibile.

A questo punto è possibile immaginare che le maschere temporali e quelle in frequenza si fondano insieme e agiscano contemporaneamente (come mostrato dalla figura seguente).

Mascheramento temporale e in frequenza

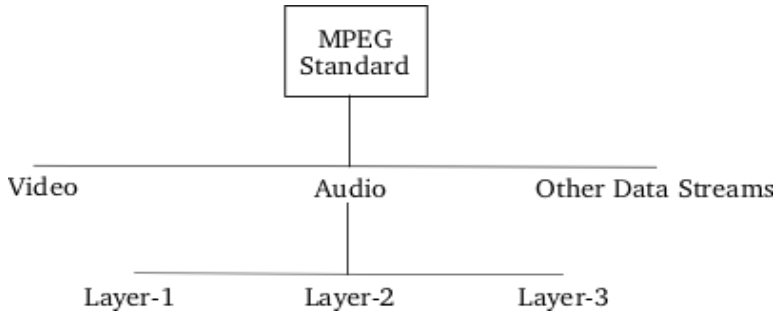


In conclusione l' effetto complessivo del mascheramento è che molti toni non sono mai udibili perché collocati nel dominio della frequenza e del tempo troppo vicino a toni forti. Tenendo conto della sensibilità dell'orecchio e del fenomeno del mascheramento è quindi possibile eliminare dallo spettro del segnale una quantità molto alta di informazioni inutili, perché non udibili dall'orecchio umano.

Questi sono i fenomeni Psico-Acustici su cui si basana i moderni algoritmi di compressione audio come MP3, AAC, ...

MPEG-AUDIO

L' Mpeg/Audio è uno standard internazionale riconosciuto dall'ISO (organizzazione internazionale preposta alla approvazione definitiva di uno standard) ed è stato varato ufficialmente nel 1992. Esso è una branca degli standard di cui si occupa il comitato Mpeg.

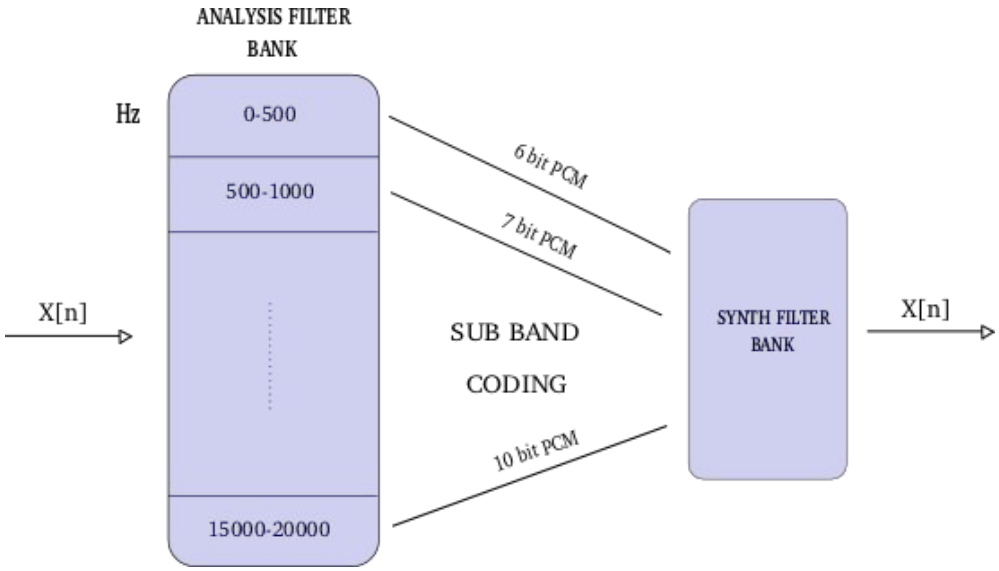


MP3

Il noto formato di compressione MP3 è appunto il Layer-3 dello standard di compressione Mpeg/Audio. Esso fu sviluppato inizialmente da una organizzazione no-profit, tuttavia la tecnologia fu brevettata e si pagano i diritti d'autore sui codificatori mentre riproduttori sono gratuiti per uso personale.

L'algoritmo di codifica MP3 è composto da diversi passi che, grosso modo, possono essere così riassunti:

- Si usano dei filtri per dividere il segnale audio che è campionato con una certa frequenza, ad esempio di 44100 campioni al secondo, in 24-32 sottobande, per ognuna delle quali sono noti i parametri di mascheramento nel tempo e in frequenza grazie ad un opportuno modello psicoacustico che abbiamo visto prima.
- Per ognuna delle sottobande, viene calcolata l'entità del mascheramento causata dalle bande adiacenti.
- Se la potenza in una sottobanda è sotto la soglia di mascheramento, allora non viene codificata in uscita l'informazione che essa trasporta, poiché non sarebbe udibile.
- Altrimenti, occorre calcolare il numero di bit necessari per rappresentare l'informazione della sottobanda facendo attenzione che in questo procedimento, per sua natura approssimante e dunque rumoroso, il rumore introdotto stia sotto la soglia (*noise floor*.)
- Infine, formare il flusso di bit (bitstream) in uscita.
-



L'MP3 utilizza il blocco dei filtri (*Analysis Filter Bank*), però a differenza dei layers 1 e 2 le sottobande non sono tutte della stessa dimensione, poiche certe frequenze contengono molta più informazione e vanno trattate con maggiore dettaglio.
Il layer 3 inoltre fa uso di una MDCT, cioè di una trasformata discreta del coseno modificata. In breve si tratta di effettuare una operazione che consenta di migliorare la risoluzione in frequenza per ognuna delle sottobande. Questa operazione consente di suddividere ognuna delle 32 sottobande in ulteriori 6(short) o 18(long) sottofrequenze, secondo un processo noto come filtraggio sottobanda (sub-band filtering).

Il modello psico-acustico lavora ulteriormente su queste sotto-sottomaschere, in particolare sui coefficienti della MDCT che le rappresentano. Il modello psico-acustico deciderà quali coefficienti devono passare in uscita e quali no, sulla base del calcolo del mascheramento temporale e sul fatto che alcuni di questi sono ridondanti giacchè magari provengono dai canali sinistro e destro che spesso portano la medesima informazione. A questo punto il tutto è quasi pronto. I coefficienti 'sopravvissuti' contengono le informazioni necessarie alle varie frequenze e devono ora essere organizzati in uscita. I coefficienti vengono ordinati passando dalla frequenza più bassa a quella più alta. Poichè la massima informazione è contenuta in bassa frequenza, i coefficienti di bassa frequenza sono più numerosi di quelli in alta frequenza.(e infatti i puristi lamentano la scarsa efficienza dell'MP3 per la riproduzione delle alte frequenze). L'intero intervallo viene diviso in tre parti (frequenze basse, medie e alte). Ognuno di questi intervalli viene codificato a parte secondo l' algoritmo di Huffmann. L'algoritmo è ottimizzato per ognuno dei tre intervalli. A questo punto i dati vengono inviati in uscita sotto forma di pacchetti che contengono un CRC(codice per la correzione dell'errore) per rendere il sistema più robusto agli eventuali errori che si possono

presentare durante il trattamento del file. Il fattore di compressione che tipicamente si ottiene è quello di 1:11 (128Kbit/s), per cui è possibile immagazzinare un minuto di musica in poco meno di un megabyte.
In termini di qualità possiamo dire che con l'MP3 otteniamo:

Stato dell'arte

L'MP3 è uno standard relativamente vecchio. È stato sviluppato nei primissimi anni '90 (venne definito nel 1991 e standardizzato dall'ISO e dalla IEC alla fine del 1992) e i suoi stessi sviluppatori riconoscono che esso ha rappresentato un primo tentativo di utilizzare efficacemente i modelli psico-acustici per limitare la ridondanza dell'informazione presente in un file campionato in modalità PCM. Il suo successo fu dovuto essenzialmente al fatto che il suo software informativo di riferimento era di ottima qualità e che il decoder MP3 potesse leggere file codificati con i layer precedenti.

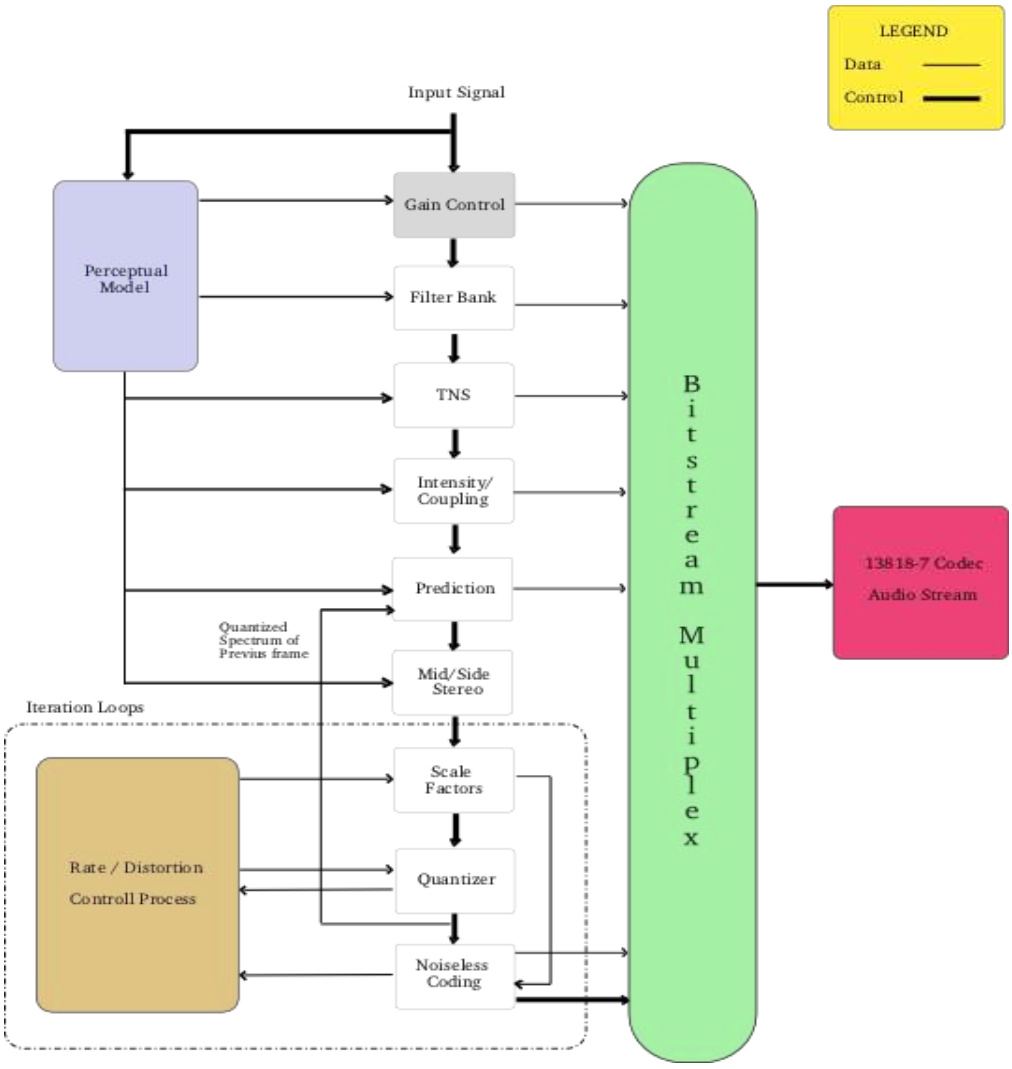
Attualmente tra le tecnologie di compressione audio digitale che vanno per la maggiore troviamo l'**MP3-pro** (l'evoluzione dell'MP3 che si basa sulla Spectral Band Replication), il **VQF**, il **G2** di Real, l'**AAC** di Dolby.

Tipo di formato	Produttore	Note	Modalità
G2	RealNetworks	Leader nel mercato dello streaming audio	streaming
TwinVQ	Yamaha	Incorporato nell'MPEG-4	download
MP3	Fraunhofer Ins.	Parte degli standard MPEG-1 e 2	download
AAC	Dolby	Parte dello standard MPEG-2	download

Cenni sull' AAC

Se l'MP3 rappresenta il passato allora la soluzione che al momento si presenta come la più promettente è senza dubbio l'AAC. AAC sta per Advanced Audio Coding ed è stata sviluppata dal **Fraunhofer Institute** (sviluppatore anche dell'MP3) e tre partner internazionali: **Dolby laboratories**, **AT&T** e **Sony** Corporation . Il fatto che tra i promotori vi sia la Dolby, che è attivamente impegnata nella ricerca di algoritmi di compressione audio che possano supportare l'effetto sorround impegnando dei bassi bit-rate,e la AT&T, che è il gigante per le telecomunicazioni americane e che quindi ha tutto l'interesse a poter usufruire di mezzi che consentano di stipare più comunicazioni sullo stesso mezzo in modo da contenere i costi delle infrastrutture, la dice lunga sulle speranze che vengono riposte nella diffusione di questo standard; standard che, non è esattamente recente, visto che è stato approvato in via definitiva dall'ISO/IEC nel lontano 1997 ma che ha faticato a decollare a causa della larga diffusione dell'MP3.

L'AAC si propone come un formato con una qualità superiore del 30% a quella dell'MP3 a parità di bit-rate. AAC è compatibile con le specifiche Dolby dell'audio multicanale, e supporta fino a 48 canali ognuno con un bit-rate pari a 96KHz.Magrado quello che uno sarebbe portato a pensare, l'AAC non presenta una riproduzione della banda a 20KHz, pari cioè alla totalità dello spettro sonoro percebibile dal nostro udito. Infatti in genere oltre i 16KHz non registra più nulla. Lo sforzo della codifica è infatti concentrato sulla **riduzione dell'errore spettrale** fra il file sorgente e il file compresso, fattore critico per mantenere elevata la qualità in sistemi multi canale.



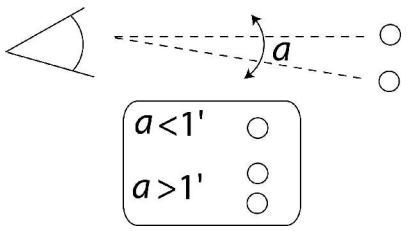
Nel soprastante diagramma a blocchi dell'AAC il percorso principale, quello dei dati processati, è evidenziato dalle freccette spesse. Le freccette sottili invece indicano i segnali di controllo. Notiamo subito che il blocco del modello psicoacustico è quello che fa da 'regista' durante tutto il percorso che i dati compiono passando dall'ingresso all'uscita. Un altro blocco importante è quello del controllo di distorsione, che supervisiona l'errore di quantizzazione.

Immagini

Immagini analogiche e digitali

Quando parliamo di immagini possiamo intendere due categorie distinte: le immagini analogiche e quelle digitali. Un' immagine si intende analogica quando è scomposta in una serie di righe (che determinano la risoluzione) di per sé continue. I fotogrammi della TV analogica sono composti da 525 righe per lo standard NTSC (Stati Uniti) e da 625 per lo standard PAL (Europa). Al contrario un' immagine si dice digitale quando è divisa in tanti puntini (pixel) ad ognuno dei quali è associata una certa intensità.
Viste da una opportuna distanza le immagini (sia analogiche che digitali) ci appaiono comunque 'continue' in quanto l'occhio umano non è in grado di distinguere un dettaglio (e quindi di discernere due pixel/righe) se questo sottende un angolo inferiore a un minuto.
Parlando di compressione la nostra attenzione sarà rivolta unicamente alle immagini digitali.

Angolo minimo per distinguere un dettaglio



RGB e YUV

Un'altra suddivisione interessante che può essere fatta per le immagini a colori è la classificazione tra immagini RGB e YUV.
L'RGB è un modello di rappresentazione basato sulla combinazione di tre onde monocromatiche: una rossa, una verde e una blu che corrispondono ad altrettanti tipi di coni presenti sulla retina dell'occhio umano.
L' YUV al contrario scompone l'immagine in luminanza (Y) e due componenti di crominanza (UV). La prima da sola da un'immagine b/n mentre le altre due aggiungono il colore. L'aver utlizzato questa tecnica nei sistemi televisivi ha garantito la compatibilità delle vecchie TV in bianco e

nero con le trasmissioni a colori (in pratica veniva elaborato solo il segnale Y di luminanza).

Alcuni metodi di compressione

Di per se già la tecnica YUV ci apre la strada per una semplice compressione del segnale se si pensa al fatto che l'occhio umano è più sensibile alle variazioni di luminosità piuttosto che a quelle di colore. In base a ciò si può quindi scegliere di ridurre la qualità delle componenti U e V lasciando inalterata quella della componente Y.
I metodi di compressione più importanti sono la codifica run length, il GIF e in particolare il JPEG.

Codifica run length

La codifica run-length si basa sul conteggio delle occorrenze successive di un dato carattere all'interno di una sequenza . Per ottenere una versione compatta della sequenza , si può sostituire ogni successione di caratteri uguali con l'indicazione del carattere stesso seguita dal valore che indica il numero di volte con cui esso compare . Appare chiaro che il metodo risulta efficiente solo se le sequenze sono di lunghezza maggiore di quella della coppia contatore-carattere.

GIF

Il formato GIF (Graphic Interchange Format) si diffuse negli anni '80 come metodo efficiente di trasmissione delle immagini su reti di dati.
Questo formato usa una forma di compressione a dizionario LZW (Lampel-Ziv-Welch**)** che mantiene inalterata la qualità dell'immagine. La profondità dei colori delle immagini GIF è di 8 bit, consentendo di usare una tavolozza di 256 colori. Meno colori si usano e maggiori saranno le possibilità di compressione, ovvero minori saranno le dimensioni del file.
Lo schema di compressione LZW è più adatto a comprimere immagini con grossi campi di colore omogeneo ed è meno efficiente nella compressione di immagini complicate con molti colori e grane complesse.
È possibile sfruttare le caratteristiche della compressione LZW per migliorarne l'efficienza e ridurre di conseguenza le dimensioni delle immagini GIF. La strategia consiste nel ridurre il numero di colori in un'immagine GIF al numero minimo necessario e nell'eliminare i colori isolati non necessari per la rappresentazione dell'immagine.
Questa compressione consente anche di salvare le immagini in un formato interllacciato. Il formato a interllaciamento produce una visualizzazione graduale di un'immagine in una serie di passate sempre più definite a mano a mano che i dati arrivano al browser. Ogni nuovo passo crea un'immagine più nitida fino al completamento dell'intera immagine.

JPEG

Sulla fine degli anni '80 gli ingegneri dell'ISO/ITU decisero di formare un gruppo di ricerca internazionale per creare un nuovo formato standard di memorizzazione delle immagini fotografiche che permettesse una forte riduzione dello spazio occupato in memoria a discapito di una più o meno marcata ma sempre accettabile degradazione della qualità. Nasce così il Joint Picture Expert Group che nel 1992 rilasciò il famoso formato di compressione JPEG che da questo gruppo prende il nome.
Il JPEG fu un decisivo passo avanti nella compressione delle immagini e si impose velocemente in particolar modo con la nascita del WWW che necessariamente aveva bisogno di formati molto snelli per la trasmissione delle immagini su internet.
Il JPEG è uno standard flessibile:vengono infatti fornite specifiche rigide per la decompressione ma solamente un guideline per la compressione.

In pratica non viene specificato come si deve fare la compressione ma solo quali regole devono essere rispettate dai dati compressi per poter ottenere poi una corretta decompressione.

Esiste comunque una procedura più o meno standard:

La procedura di compressione

1. Conversione RGB → YUV (opzionale)
L'immagine originale viene convertita dallo spazio cromatico RGB a quello YUV Questa separazione, se pur non necessaria strettamente, permette una migliore compressione sfruttando il principio già accennato per cui conviene comprimere maggiormente le componenti cromatiche e mantenere intatte le informazioni sulla luminosità.
La riduzione avviene tipicamente facendo la media a due a due tra pixel adiacenti dei piani U e V, in pratica viene dimezzata la risoluzione orizzonale di questi piani. In teoria è possibile anche dimezzare la risoluzione verticale (come ad esempio nell'MPEG1) ma non è una pratica tipicamente usata nel JPEG. Nel primo caso si parla di codifica 4:2:2 nel secondo caso di 4:1:1. Già questo accorgimento lossy (cioè con perdita di informazioni) permette una riduzione delle dimensioni del 30% o del 50% rispettivamente.
Se questa fase viene saltata, la successiva processerà l'immagine in RGB invece che in YUV.

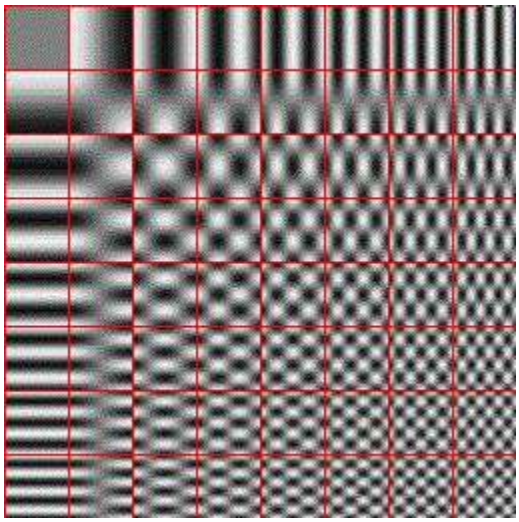
2. Analisi in frequenza (DCT)
L'elaborazione chiave del JPEG è sicuramente la DCT, **Discrete Cosine Transform** nella sua versione bidimensionale (2D). La DCT è una trasformata che in generale fa passare il segnale dal dominio del tempo al dominio della frequenza. Si tratta di una versione in campo reale della FFT che invece è in campo complesso. I coefficienti che ne derivano rappresentano le ampiezze di quei segnali armonici (coseno) che sommati ricostruiscono il segnale.
Nel JPEG viene usata la versione bidimensionale della DCT ed in questo caso non si parla di tempo e frequenza ma di spazio e frequenze spaziali. Per poter essere processata, l'immagine viene divisa in piani (tre piani cromatici R-G-B o Y-U-V a seconda dallo stadio precedente) e all'interno di ogni piano viene di nuovo suddivisa in blocchi di 8x8 pixel.
Il blocco di 8x8 pixel nel dominio dello 'spazio' viene trasformato in un blocco di 8x8 coefficienti nel dominio della frequenza spaziale. In questo blocco avremo i coefficienti in alto a sinistra che rappresentano le basse frequenze spaziali mentre quelli via via in basso a destra rappresentano le alte frequenze spaziali ossia i dettagli dell'immagine. In particolare il primo coefficiente del blocco trasformato rappresenta la media dei valori del blocco 8x8 originario (detto anche componente continua o DC).

La formula matematica della trasformata DCT in 2 dimensioni è la seguente:

$$B(k_1,k_2)=\sum_{i=0}^{N_1-1}\sum_{j=0}^{N_2-1}4\,A(i,j)\cos\left[\frac{\pi k_1}{2N_1}(2i+1)\right]\cos\left[\frac{\pi k_2}{2N_2}(2j+1)\right]$$

dove N1 e N2 sono le dimensioni in pixel dell'immagine di partenza, A(i,j) è l'intensità del pixel in posizione i,j; B(k1,k2) è il coefficiente risultante.

Matrice dei coefficienti della trasformata spaziale bidimensionale DCT



---

3. Quantizzazione

In questa fase avviene l'eliminazione delle informazioni visive meno importanti. Ciò si realizza moltiplicando la matrice 8x8 di coefficienti in frequenza per una tabella, detta *quantization table*, la quale contiene valori tra 0 e 1: quelli più bassi si trovano in corrispondenza delle alte frequenze mentre quelli più alti in corrispondeza delle basse frequenze.
I valori così ottenuti vengono arrotondati all'intero più vicino, in questo modo i coefficienti meno significativi tendono ad azzerarsi mentre rimangono i coefficienti relativi ai contributi informativi più importanti. Essendo già piccoli, i valori in alta frequenza vengono molto spesso arrotondati a 0. Il risultato è la concentrazione di pochi coefficienti diversi da 0 in alto a sinistra e 0 tutti gli altri.

Quando in un file JPEG si sceglie il fattore di compressione, in realtà si sceglie un fattore di scala sui valori della 'quantization table'. Più i valori sono bassi e maggiore è il numero di coefficienti che si azzerano con

conseguente riduzione del numero di coefficienti significativi. Questo processo ovviamente cancella informazioni via via più importanti e porta ad un progressivo deterioramento della qualità dell'immagine compressa.

Deterioramento dell'immagine in seguito all'eliminazione di troppi coefficienti della DCT

---

4. Codifica entropica

Una volta eliminati i dettagli meno importanti grazie alla DCT e alla quantizzazione, è necessario adottare una serie di tecniche entropiche per ridurre la quantità di memoria necessaria per trasmettere le restanti informazioni significative:

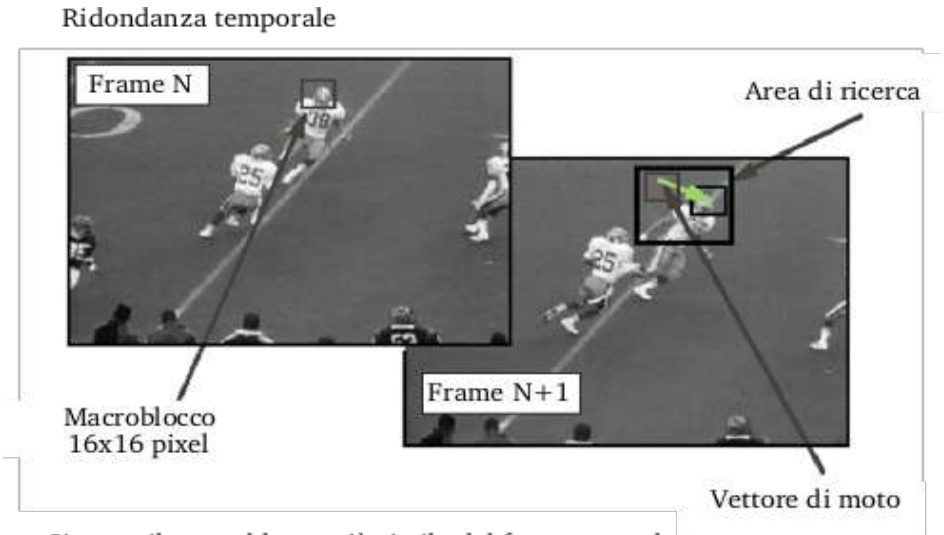
- Lettura a Zig-Zag: necessaria per rendere adiacenti il più possibile i coefficienti uguali a 0 permettendo una ottimale rappresentazione dei dati tramite codifica Run Lenght.
- Run Length Encode (RLE): Il vettore risultante dalla lettura a Zig-Zag contiene molti zeri in sequenza, per questo si rappresenta il vettore tramite coppie (skip, value), dove *skip* è il numero di valori uguali a zero e *value* è il successivo valore diverso da zero. La coppia (0,0) viene considerata come segnale di fine sequenza.
- Compressione Huffman:l'ultima codifica entropica applicata ai dati è la classica codifica a lunghezza di codice variabile. In pratica i dati vengono suddivisi in stringhe di bit, viene analizzata la frequenza statistica di ciascuna stringa e ognuna viene ricodificata con un codice a lunghezza variabile in funzione della frequenza di apparizione: codice corto per quelle che appaiono frequentemente e via via codici più lunghi per quelle meno frequenti.

JPEG è un codec simmetrico ne consegue quindi che l'elaborazione necessaria per la **decompressione** sia l'esatto inverso di quella necessaria per la compressione.

Video

Se il problema della compressione delle immagini si stava risolvendo con la definizione dello standard JPEG per la compressione digitale dei flussi video ancora nel 1990 non si era trovata una soluzione soddisfacente.
Un filmato full-pal è costituito da 25 fotogrammi al secondo ampi 704x576 pixel. In formato RGB a 24bit ciò significa 30Mbyte/s di flusso dati.
Essendo un flusso video composto da una successione di singoli fotogrammi, la cosa più naturale da fare in prima istanza è comprimere singolarmente i singoli fotogrammi. Utilizzando ad esempio il JPEG si ottiene il noto formato MotionJPEG, spesso utilizzato in schede di montaggio video ed acquisitori che porta il flusso dati a 3MByte/s. Questi valori sono chiaramente ancora eccessivi per ambiti quali l'archiviazione su supporti ottici e del tutto impossibili da sostenere per effettuare video comunicazione uno a uno (video conferenza) o uno a molti (trasmissioni broadcast).

Macroblocchi e ridondanza temporale



Si cerca il macroblocco più simile del frame precedente
e si invia solamente il vettore di spostamento.
Solamente i macroblocchi intra sono inviati come se fossero nuovi

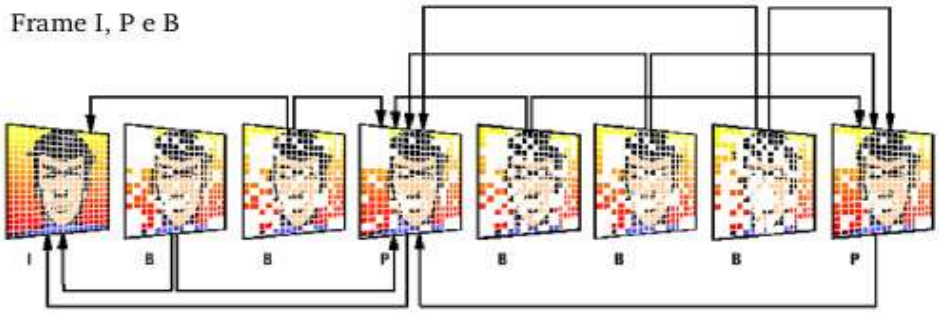
Compensazione del moto

Nella compressione video, a causa della forte correlazione temporale, tra un frame ed il successivo si può notare che molti oggetti ovvero parti dell'immagine si spostano lentamente (questo è fondamentalmente vero per filmati di scene reali). Da questo si può dedurre la possibilità di approssimare il frame successivo facendo (almeno in parte) ricorso allo spostamento di frammenti del precedente (*motion vector*). Il modo di procedere consiste nel suddividere una immagine in blocchi e nel tentare di stimare all'interno del frame precedente (od un altro qualsiasi dei circostanti) quali siano i blocchi che presentano una maggiore somiglianza con quello che si considera (stima del moto). La ricostruzione di una immagine in base a segmenti di altre prende il nome di compensazione del moto.

MPEG

Nello standard MPEG esistono tre tipi di fotogrammi : fotogrammi che vengono codificati singolarmente senza nessun riferimento ad altri (Intraframes o I frames), fotogrammi che vengono predetti sulla base di un frame di tipo I (Forward predicted frames o P frames), e fotogrammi che vengono ottenuti interpolando fra un frame I ed un frame P (Bidirectional frames o B frames). In MPEG quindi la predizione di un fotogramma puo' essere fatta considerando sia la storia passata (I frames) che quella futura (P frames).

Frame I P e B



I-frames: contengono le informazioni dell'intera immagine
P-frames: calcolati in base alle differenze con un frame I o P precedente
B-frames: interpolati con frames I e P

[Un frame I è trasmesso almeno ogni 1/2 secondo]

In sostanza come primo passo viene generato un frame I considerandolo come una singola immagine fissa. Per il calcolo del motion vector e la predizione del frame P si considerano i punti all'interno di blocchi 16x16 (macroblock) nel canale di luminanza Y e nei corrispondenti blocchi 8x8 nei canali di crominanza U e V. Per ognuno di questi blocchi si cerca quello che ad esso si avvicina di piu' nell'ultimo frame I o P inviato, il verso e la direzione fra questi due blocchi identificano il motion vector. Se si riesce ad individuare il motion vector, per specificare il blocco nel frame P che stiamo codificando bastera' indicare, oltre ovviamente al motion vector stesso, la differenza fra i punti dei due blocchi in esame. Una volta codificato un frame I ed uno P si possono codificare i frames B compresi fra essi.
Allora si esaminano i macroblocchi dei fotogrammi compresi fra il frame I e quello P cercando per ogni blocco quello a lui più simile nel frame I (quindi

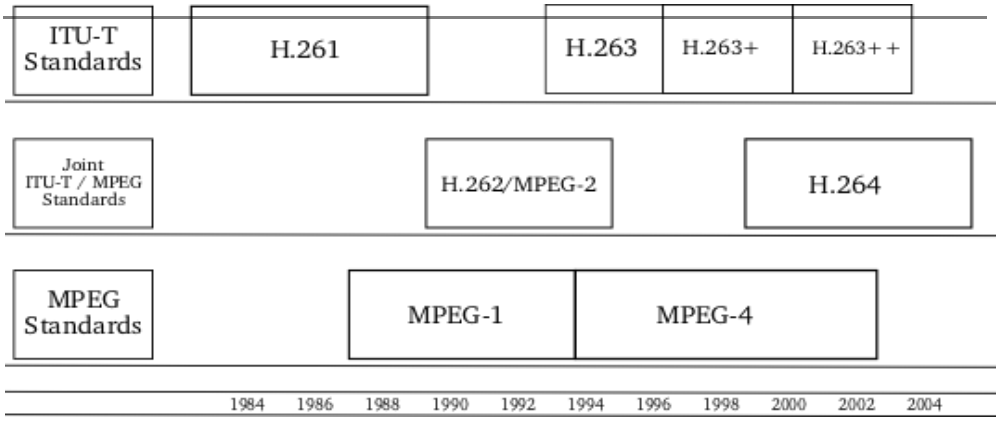
indietro nel tempo), quello piu' simile nel frame P (quindi avanti nel tempo) oppure cercando di fare una media fra il blocco più simile nel frame I e quello piu' simile nel frame P e sottraendo a questa il blocco da codificare.
Se con nessuno di questi tre procedimenti si ottiene un risultato soddisfacente si puo' sempre codificare il blocco come se facesse parte di un frame I ovvero senza riferimenti ai blocchi precedenti o futuri.

Quindi otteniamo logicamente una sequenza di frames del tipo:

I B B P B B P B B P B B I B B P B B P B B P B B I ...

(tipica sequenza di codifica MPEG) in cui ci devono essere al massimo 12 frames fra un frame di tipo I ed il successivo (codifica SIF US), mentre la successione di frames P e B è libera.
Questo permette di avere un accesso casuale al flusso di immagini ogni **1/30*12=0,4 sec** e di fare in modo che le immagini codificate non divergano troppe da quelle reali.
Naturalmente, visto che per poter decodificare i frames B occorre conoscere gia' il frame P successivo, la sequenza dei frames che vengono inviati dopo la codifica e' diversa da quella logica.

L'**ITU-T**, *International Telecommunication Union* è una delle due organizzazioni in gradi di sviluppare ed approvare standard di codifica video digitale. L'altro organismo è l'**ISO/IEC** JTC1.
Gli standard emanati dall'ITU sono spesso chiamati raccomandazioni e sono denotate dalla sigla H.26x (H.261,H.262,H.263,H.264), mentre gli standard ISO/IEC vengono denotati con la sigla MPEG-x (MPEG-1,MPEG-2,MPEG-4,etc…).
Le due organizzazioni hanno lavorato per lo più separatamente eccetto per l'H.262 alias MPEG2 e più recentemente per la messa a punto dell'H.264 noto anche come MPEG4 part 10 o MPEG4 AVC (Advanced Video Codec).



MPEG 4

Nell'ambito dello scenario wireless e in quello di Internet, grazie alle sue caratteristiche, si sta affermando sempre di più il formato MPEG4.

L'MPEG-4 supporta le seguenti funzionalità:

- ***Interattività basata sul contenuto*** : riguarda l'interazione tra l'utente e i dati

1. Strumenti per l'accesso basato sul contenuto ai dati multimediali
2. Manipolazione del bitstream basata sul contenuto
3. Codifica ibrida naturale e sintetica
4. Accesso casuale ai dati

La struttura dell'MPEG-4 supporta la composizione di scene ibride, contenente oggetti sia naturali che sintetici. Ogni singolo oggetto può essere diviso in sotto-oggetti i quali a loro volta possono essere riordinati per formare un oggetto composto. Ogni oggetto può essere manipolato dall'utente in ogni sua caratteristica (colore, forma, dimensione ... etc) specificando le proprietà della trasformazione. Gli oggetti comprendono dati video e dati audio indipendenti tra di loro in modo da permettere la manipolazione da parte dell'utente. La composizione gerarchica di oggetti è supportata tramite multiplazione gerarchica del bitstream.

- ***Compressione dati***: riguarda l'uso di metodi efficienti per l'immagazzinamento e la trasmissione di dati audiovisivi

1. Miglioramento dell'efficienza di codifica
2. Codifica di più flussi di dati concorrenti

Ogni singola immagine può essere suddivisa in frammenti che possono essere traslati e posizionati per ricreare l'immagine; questi frammenti sono funzioni particolari come la DCT (variante della trasformata di Fourier) oppure blocchi parzialmente o completamente ricostruiti. In questo modo un oggetto anche complesso può essere descritto efficacemente da linee di codice.

- ***Accesso universale***: rende i dati codificati accessibili a decodificatori di diversa qualità

1. Robustezza agli errori (Punti di sincronia a codici entropici, Data partinioning, codifica entropica robusta)
2. Scalabilità basata sul contenuto

La qualità dell'immagine può essere suddivisa su diversi livelli, costruiti assegnando una priorità ad ogni oggetto ed effettuando un'interpretazione a partire dagli oggetti più importanti a quelli meno importanti. Questa suddivisione viene definita scalabilità della banda e della risoluzione. Questo metodo permette ai decodificatori di diversa qualità di visualizzare gli stessi dati.

Lo standard MPEG4 è stato sviluppato per permettere la manipolazione del video utilizzando la ***codifica ad oggetti*** con lo scopo di codificare sia dati immagine e sia dati audio in un modo altamente flessibile.
Lo standard video MPEG4 introduce il concetto di VOP (Video Objet Planes) , in cui ogni parte della sequenza video di ingresso viene divisa in un numero di regioni con immagini di forma arbitraria. Ogni regione può coprire una parte dell immagine o il contenuto d interesse ossia definire fisicamente oggetti o contenuti all interno della scena, in cui l informazione video non è più vincolata ad essere di forma rettangolare ma può rappresentare ad esempio una persona sola e non un intera scena. Quindi il singolo oggetto audiovisivo può essere sia sintetico che naturale.

L'MPEG4 offre:

- *Video:* Naturale, Texture, modelli 3D basati su Computer Graphics, talking heads
- *Audio*: - AAC MPEG4
    1. 96 canali
    2. Fc a 8 Khz (molto bassa)
    3. Bitrate alti 192 kbit/s per canale

    - Audio Sintetico (midi like)

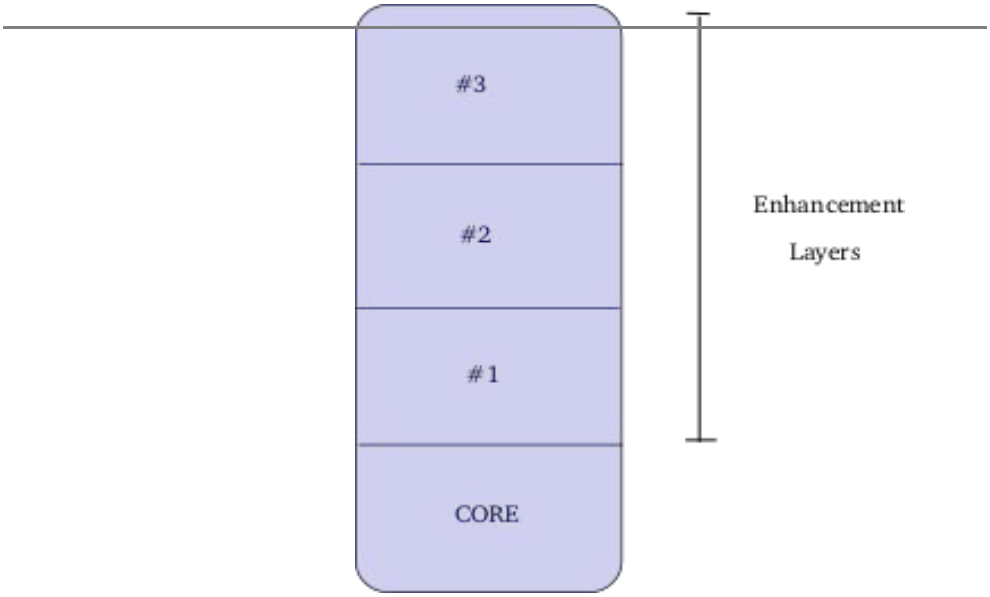
    - Audio naturale a bassissimo rate (16 kbit/s TwinVQ)

    - Voce naturale
    4. Vocoder LPC (HVCX Sony) 2 Kbit/s
    5. CELP (NEC) 4-16 Kbit/s

    -Voce sintetica: text-to-speech <1 Kbit/s

- *Scalabilità:*

Il bit stream compresso non è monolitico ma formato da strati



E' composto da un CORE fondametale ai fini della decodifica e da altri livelli che migliorano la qualità del primo. Questo permette una maggiore flessibilità in decodifica.
Infatti nel caso dello streaming mi basta ricevere il CORE per iniziare la decodifica mentre se ricevo tutti livelli ho una qualità pari a Q_{MAX}.

Multimedia Over IP

Trasmissione su canale con rumore
(impatto di perdite sulle diverse tecniche)

- Vocoder LPC [perdita di 1 frame]: impatto sul frame corrente
- DPCM [errori sul bit]: impattto campione corrente e n campioni futuri
- Video [perdita di 1 frame]: frame corrente :ricostruito. Frame futuri impatto variabile a seconda del tipo: B (nessun impatto) P (su frame B e P che lo referenziano) I (su B e P che lo refernziano anche indirettamente)
- Mp3 [errori sul bit] : impatto su uno dei segnali passa-banda. Impatto su campioni successivi se DPCM.

Caratteristiche del protocollo IP

Per parlare di multimedia over IP è bene partire dalle caratteristiche principali di questo protocollo in quanto le scelte fatte per l'invio di flussi multimediali sono state condizionate fondamentalmente dalla sua natura.

- nessuna ottimizzazione poichè si intende garantire apertura al futuro
- nessuna discriminazione tra applicazioni
- l'intelligenza è ai bordi della rete (*stupid network – smart application*)
- *best effort* (nessuna garanzia su loss e delay e jitter)
- soggetta a packet loss rate: il router può decidere di scartare dei pacchetti in caso di congestione. (Parlando di multimedia si intendono pacchetti persi anche quelli arrivati troppo tardi – *late packets*)
- rete aperta (le innovazioni possono essere messe subito in campo – j*ust plug in*)

L'utilizzo combinato di TCP/IP è inadeguato poichè non garantisce agli utenti la trasmissione di un certo numero di dati in un preciso periodo di tempo. Le prestazioni della rete possono fluttuare di momento in momento: a volte i dati sono trasmessi immediatamente, a volte subiscono ritardi o non sono inviati affatto.

RTP/RTCP

Si è scelto di utilizzare un protocllo più vecchio di TCP chiamato UDP (*User Datagram Protocol*) soprattutto per l'identificazione delle porte su host (caratteristica necessaria per i flussi multimediali) e sopra si sono costruiti i protocolli RTP(*Real Time Protocol*) e RTCP(*Real Time Control Protocol*).

I pacchetti RTP sono caratterizzati da:

- sequence number
- payload type: 8 bit (di cui 7 utilizzati – 128 tipi)
- timestamp (utile per identificare i late packets, il tempo di playout e calcolare ritardi e jitter)

- source identifier: ID unico per flusso RTP

Il sequence number e il source identifier sono inizializzati con numeri casuali per evitare attacchi di tipo *known-text*.

Il protocollo di controllo RTCP invece si occupa dei Sender Reports e dei Receiver Reports utili per:
- gestione della connessione
- descrizione del flusso
- monitorare l'andamento della trasmissione (PLR, delay e jitter)

L'RFC raccomanda che questi pacchetti siano inviati ogni 5 sec. In generale comunque il traffico RTCP non deve essere superiore al 5% del traffico RTP poichè si tratta di traffico di overhead.

Applicazioni del Multimedia Over IP

Caratteristiche delle applicazioni MMOIP

Le applicazioni del multimedia over IP è possibile classificarle in base ad alcune caratteristiche principali:

- interattive / unidirezionali
- live / non live (*on demand*)
- audio / video

Riguardo le applicazioni interattive ci interessa in particolare il VoIP mentre per le applicazioni unidirezionali parleremo dello streaming.

VoIP

Per **VoIp** si intende la trasmissione di comunicazioni voce su rete IP.
I vantaggi nel sostituire le vecchie reti telefoniche con un architettura VoIP sono:

- avere un'unica rete per dati e voce
- passare da una tecnologia proprietaria a una IP
- facilità nell'intordurre nuovi servizi e nuove funzionalità (gestione segreteria, *unified messaging*)
- sicurezza grazie all'ausilio di crittografia e certificazione del telefono e/o del parlatore.

Le reti che è possibile utilizzare per il VoIP sono:
- reti IP wired (LAN)
- UMTS (3G)
- WLAN 802.11 (cordless, hotspot)

Trattandosi di comunicazione interattiva bisogna dare molto peso al ritardo introdotto che in questo caso è inteso come ritardo bocca-orecchio. Dai test è risultato che con un ritardo inferiore ai 150 ms si ha la cosiddetta tall quality, con valori superiori ai 400 ms la qualità diventa inaccettabile mentre con tutti i valori intermedi si ha una qualità adatta solo per servizi a basso costo.Come in tutte le comunicazioni multimediali il ritardo viene gestito come compromesso tra l'inserimento di più frame nello stesso pacchetto

(maggiore ritardo), l'overhead introdotto dagli header inserendone meno e le dimensioni del playout buffer.

Riguardo agli errori abbiamo:

- per errori sul bit una probabilità inferiore a 0,001 è in media percettivamente trasparente.
- per errori sul frame una probabilità del 1% è in media percettivamente trasparente.

Per il packet lost rate una probabilità minore del 1% è tall quality, con una probabilità intorno al 3-5% abbiamo già un impatto sul MOS significativo (anche maggiore di 0,5) mentre per valori superiori la qualità diventa inaccettabile.

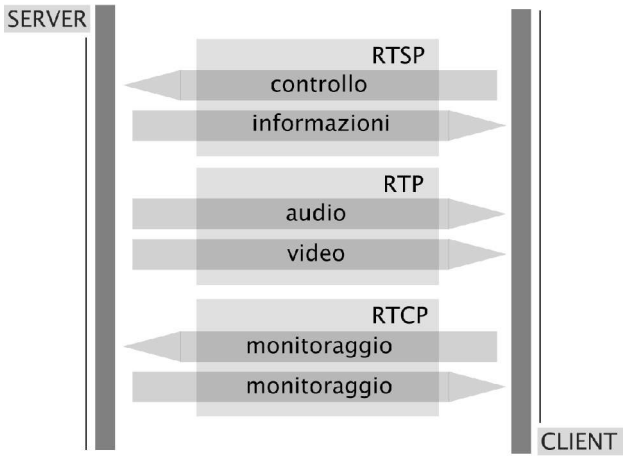
Altre importanti caratteristiche introdotte nel VoIP sono:

- cancellazione dell'eco
- Supporto DTMF (Dual Tone Multi Frequency)
- Voice Activity Detector (permette di recuperare fino al 50 % della banda non inviando alcun segnale, o inviando pochi pacchetti, quando l'utilizzatore è in ascolto.

Streaming

Lo **streaming** è una modalità di accesso a contenuti che permette il playback dei dati mano a mano che arrivano.
Le princiapli caratteristiche di questa modalità sono : un utilizzo efficiente delle risorse di rete e di memoria su disco(utile soprattuto per applicazioni su palmari e telefoni cellulari),la fruizione immediata, la flessibilità di accesso (nel caso *non-live*), l'intuitività (utilizza comandi comuni come play,stop,pause,...).
I protocolli utlizzati sono:

- IP/UDP/RTP : per il trasporto dei dati (come nel caso del VoIP)
- RTSP (*Real Time Streaming Protocol*): per la gestione e il controllo



Dal punto di vista software lo streaming è una architettura client/server.
Non avendo bisogno dell'interattività (come accadeva per il VoIP) dal lato client è possibile utilizzare un buffer di playout anche molto ampio (secondi) con il vantaggio di recuperare i *late packets* e di poter richiedere anche più volte i pacchetti che sono andati persi.

RTSP

Vediamo ora nel dettaglio come funziona il protocollo di gestione controllo.
Il Real Time Streaming Protocol (RTSP), è un protocollo di livello applicativo realizzato e sviluppato da RealNetworks, Netscape e dalla Columbia University, il cui obiettivo è essenzialmente quello di offrire supporto affidabile allo streaming di dati multimediali in Internet, basandosi su comunicazioni unicast e multicast. L'idea innovativa di tale protocollo è che esso nasce come "telecomando di rete" per controllare l'erogazione di contenuti multimediali allocati su servers opportuni.
Innanzitutto è un protocollo di tipo "text-based", molto simile al HTTP, i cui messaggi, composti di headers e body, implementano praticamente delle richieste e delle risposte necessarie alla comunicazione tra client e server ma a differenza dell'HTTP che non definisce precisamente il genere di file che vengono trasferiti il RTSP controlla i dati con funzionalità simili concettualmente a quelle di un videoregistratore.

In un messaggio di richiesta, si distinguerà nella prima linea l'indicazione del metodo, la URI della richiesta e l'indicazione della versione del protocollo (RTSP/1.0).
I possibili metodi sono 11 di cui 6 obbligatori e 5 facoltativi.

Metodi obbligatori

I metodi obbligatori sono:

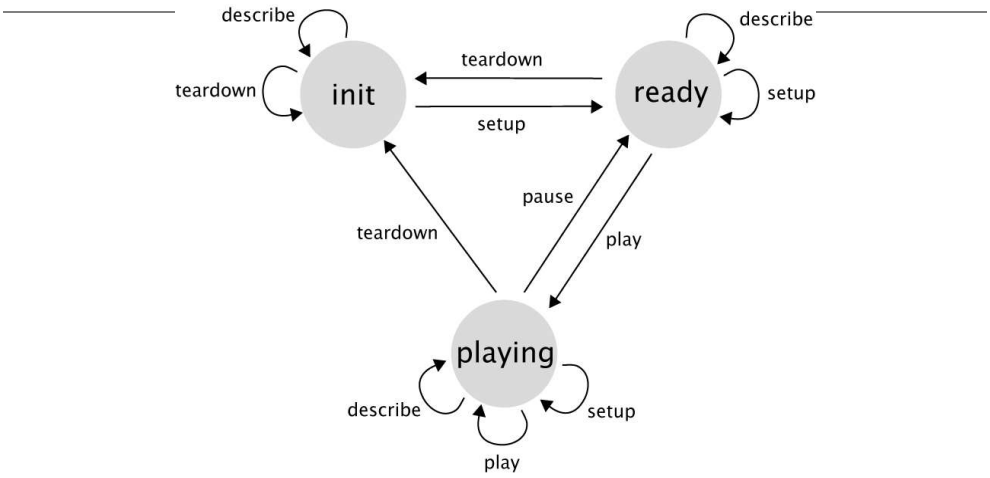
- Describe : con tale metodo un client comunica la volontà al server di ricevere la descrizione della presentazione identificata dalla URL data. In genere con tale richiesta viene fornito nel header anche il campo "Accept" con cui il client specifica quali sono i formati di descrizione che esso è in grado di comprendere
- Options : utilizzato in genere dal client per chiedere al server quali metodi esso supporta
- Setup : tale richiesta specifica il meccanismo di trasporto che deve essere utilizzato per lo streaming; addirittura un client potrebbe

richiedere di cambiare i parametri di trasmissione per un flusso multimediale che è già in riproduzione
- Play : con tale metodo, un client chiede al server di iniziare il trasferimento dei dati del media tramite il meccanismo specificato nel metodo 'setup'
- Pause : tale richiesta causa la temporanea interruzione della trasmissione dei dati e quindi della riproduzione; è naturale che qualora fosse indicata con tale metodo la URL di una sola traccia del flusso multimediale, solo essa verrebbe stoppata. Nel caso ad esempio di una traccia audio ciò equivarrebbe ad un "mute"
- Teardown : tale richiesta termina la trasmissione dei dati per quel dato URI, rilasciando tutte le risorse allocate precedentemente per lo streaming

I metodi facoltativi sono:

Metodi facoltativi

- Announce : inviato da un client verso il server consegna la descrizione di una risorsa multimediale al server, inviato dal server verso il client serve per aggiornare la precedente descrizione in tempo reale
- Get_parameter : utilizzato per ottenere il valore di uno o più parametri della risorsa multimediale specificata nel URI
- Set_parameter : con tale richiesta si setta il valore di un parametro per una dato risorsa
- Record : con tale metodo, si richiede la registrazione di un opportuno range di dati facenti parte di un dato flusso
- Redirect : una tale richiesta informa il client che si deve connettere ad un altro server, deve quindi contenere nel header obbligatoriamente il campo Location, in cui sarà indicata la nuova URL verso cui si dovranno inoltrare le richieste



Nel messaggio di risposta, la prima linea è la "Status-Line", consistente nella versione del protocollo seguito da un codice di stato numerico; questultimo, è un intero di tre cifre necessario per codificare il tipo di risposta nel messaggio e che usualmente è seguito da una frase dove si esprime, in un formato più intuibile dall'uomo, il significato di tale risposta (es. "200 OK" o "404 Not Found").

Quality of service: tecniche best effort

Il problema fondamentale che sta alla base delle comunicazioni multimediali su una rete come Internet è che la gestione dei pacchetti da parte del protocollo IP destinatario comporta una ricomposizione di questi nell'ordine originale. Se alcuni subiscono errori di trasmissione o ritardi vari si capisce come il processo di ricomposizione possa risultare rallentato. Per applicazioni classiche, come ad esempio la posta elettronica, tale ritardo non acquista particolare significato, ma per applicazioni real-time, dove l'interazione tra le parti è fondamentale, questo può provocare un degradamento notevole nella qualità del servizio.

Ci sono due possibili famiglie di tecniche per garantire la qualità del servizio quando si parla di Multimedia Over IP:

Tipi di tecniche

- Tecniche best effort
- Soluzioni architetturali

Obiettivi

Gli obiettivi che si vogliono raggiungere con queste tecniche sono:

- riduzione/controllo del packet losses
- se possibile riduzione/controllo del ritardo (tipicamente in maniera indiretta)

Usando il protocollo IP il progettista ha un ampio grado di libertà potendo utilizzare tecniche di cross-layer design (o joint source channel coding).

Tecniche best effort

Vediamo ora nel dettaglio quali sono i fattori modificabili in fase di progetto rimanendo nell'ambito delle tecniche best effort.

1.Configurazione dell'encoder multimediale

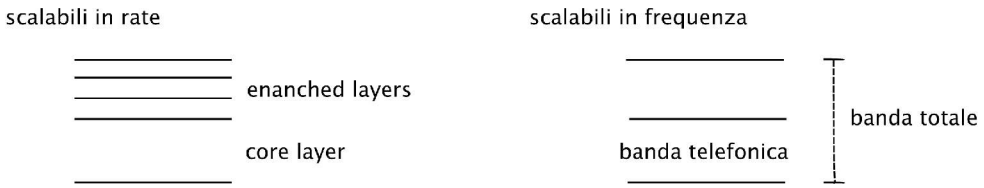
Parlando di trasmissione di segnale **voce/audio** le tecniche possibili sono:

- codec VBR netowork driven (il bitrate è 'pilotato' dalla stato della rete). Per quanto riguarda la voce questa tecnica è utilizzata dal GSM-AMR
- codec che producono bitstream adatti per garantire la robostezza:

1. scalabili in rate

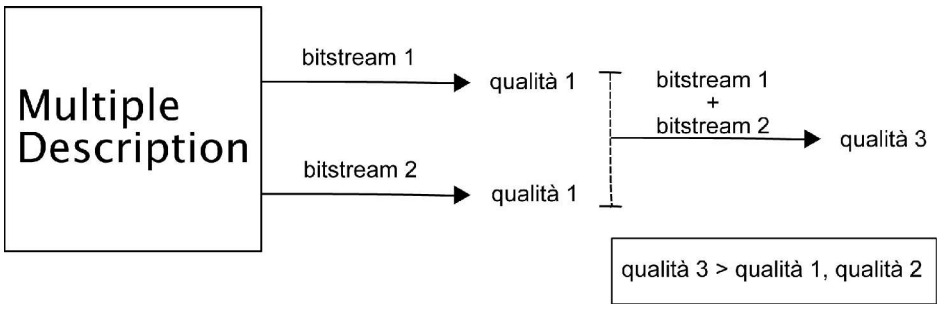
2. scalabili in frequenza

Codec scalabili



- multiple description coding: il segnale codificato è diviso in più parti ciascuna delle quali decodificabili indipendentemente dalle altre garantiscono una qualità accettabile. La qualità migliore si ha quando si ricevono tutte le parti.

Multiple Description



Per quanto riguarda la trasmissione di **immagini** le tecniche sono:

- DCT (Discrete Cosine Transform) based (ad esempio JPEG)
- livello di compressione (numero di bit per il coefficiente in continua e matrice di quantizzazione)
- codec scalabili (in rate o con tecniche Multiple Description)

Vediamo infine le tecniche per la trasmissione del **video**:

- livello di compressione (VBR network driven)
- struttra del GOP: struttura I-P-B modificabile ad ogni GOP
- slicing: facendo attenzione all' overhead introdotto dagli header
- frame drop
- codec scalabili: in rate oppure con tecniche Multiple Description (ad esempio inviando due griglie di pixel codificabili indipendentemente)
- rate control: controllo in uscita per garantire un flusso il più possibile costante

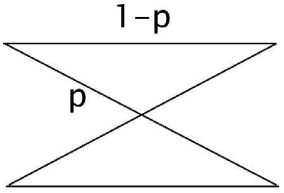
2.Pacchettizzazione

Tipicamente la cosa migliore da fare è inserire in un pacchetto N unità indipendentemente codificabili del bitstream compresso. Il significato di "unità" cambia però a seconda del tipo di stream che stiamo considerando: si tratta di frame nel caso di voce e audio, macroblocchi nel caso delle immagini, slice nel caso del video. Ovviamente occorre fare attenzione alle codifiche differenziali per le quali non è facile ottenere una vera è propria indipendenza tra le varie unità.
I fattori influenzati dalla pacchettizzazione sono:

- ritardo: per la voce e per l'audio più frame per pacchetto significano più ritardo, nel caso del video più slice non necessariamente implicano un maggiore ritardo.
- overhead: introdotto dai 40 byte per header protocollari RTP/UDP/IP. E' un incentivo ad aumentare la dimensione dei pacchetti.
- robustezza: cioè la variazione di percezione in caso di perdita di un pacchetto. E' difficile da quantificare ma in linea generale possiamo dire che:

1. per voce e audio: pacchetti grossi sono più difficili da mascherare
2. per immagini e video: calcolando il PSNR (rapporto segnale/rumore di picco) per ogni frame e facendo la media si ottiene che è meglio avere pacchetti grossi (percettivamente è meno fastidioso un fotogramma molto rovinato che più fotogrammi poco rovinati).

$$PSNR = 10 \log_{10} \frac{255^2}{\sum (X[n] - \hat{x}[n])^2}$$



$$p(\text{pacchetto non perfetto } >1 \text{ errori}) = 1-(1-p)^N$$

p.errore = p

802.11 (Standard IEEE per Wi-Fi)

La dimensione del pacchetto influenza in maniera significativa la probabilità ritrasmissione. Un pacchetto deve essere ritrasmesso nel caso contenga uno o più bit corrotti oppure se è avvenuta una collisione: entrambe le cause sono direttamente legate alla dimensione del pacchetto ma in modo opposto.

- errori sul bit: maggiore è la dimensione del pacchetto maggiore è la probabilità che contenga un errore.

Quindi se il canale è molto rumoroso conviene avere pacchetti piccoli

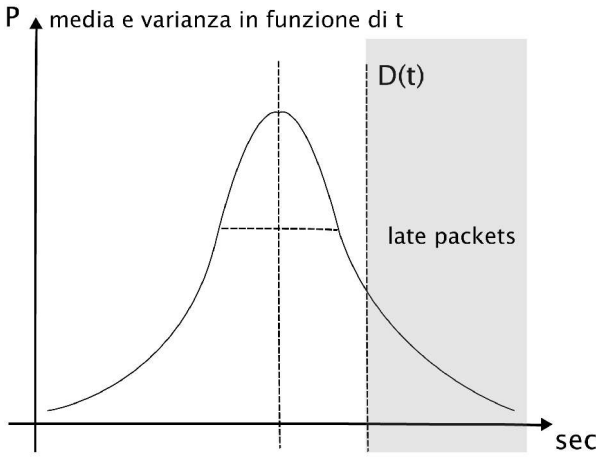
- collisioni: maggiore è il numero di pacchetti maggiore è la probabilità di collisione. Quindi se il canale è congestionato i pacchetti dovrebbero essere grandi.

Occore quindi trovare un compromesso per la dimensione dei pacchetti.

3.Buffer di playout

Anche in questo caso ci troviamo ad affrontare un compromesso tra ritardo e perdita di pacchetti. Dal momento che la curva di probabilità che modellizza l'arrivo dei pacchetti non è costante la cosa più conveniente da fare è utilizzare un buffer di playout anch'esso dinamico.

Buffer di playout dinamico



4.Mascheramento dell'errore (error concealment)

Per mascheramento dell'errore si intendono tecniche passive *receiver side* atte a riempire (in modo più o meno elaborato) i buchi formatisi in seguito alla perdita di pacchetti sul canale. L' informazione fondamentale che sfruttano queste tecniche è la conoscenza esatta del punto di flusso multimediale che bisogna ricostruire.
Il mascheramento è importante poichè migliora notevolemente l'impatto percettivo e inoltre non costa nulla in termini di rate (tutto viene eseguito dal ricevitore).
Per voce/audio erano state inizialmente proposte tenciche che prevedevano la sostituzione dei frame persi con silenzio oppure con rumore bianco. Entrambe danno però risultati assolutamente inadeguati.
Una tenica che da invece buoni risultati è l'**estrapolazione** che può essere esguita sia nel dominio della forma d'onda sia in quello dei paramtri (vocoder LPC). Essa consiste nel ripetere rispettivamente la forma d'onda o i parametri del frame precedente per tutta la durata del frame perso applicando nel caso di perdite multiple un guadagno negativo ad ogni ripetizione in modo che avvenga un azzeramento dopo 6-7 frame. Nel caso di estrapolazione nel dominio della forma d'onda devo anche tenere conto del salto di forma d'onda tra la parte vera e quella ricostruita provvedendo quindi nella prima porzione del frame ricostruito al congiungimento delle due parti per evitare fastidiosi *glitch* del suono. Questo non avviene nel dominio dei parametri.
Per quanto riguarda immagini e video si può sfruttare rispettivamente la ridondanza spaziale e la ridondanza temporale del flusso trasmesso oppure per le tecniche a compensazione del moto le dipendenze che ci sono tra vettori di moto relativi a blocchi adiacenti.

Perchè il concealment funziona?

Fondamentalmente possiamo affermare che le teniche di mascheramento dell'errore funzionano principalmente poichè il compressore a monte non è perfetto e quindi il flusso di dati che arriva al ricevitore contiene ancora delle ridondanze al suo interno. Se così non fosse il segnale inviato sarebbe rumore bianco e la perdita anche di un solo bit non sarebbe assolutamente mascherabile (poichè tutti i bit sarebbero ugualmente fondamentali).
Se si volesse quindi realizzare un encoder perfetto ci si troverebbe nella condizione di dover trovare altre tecniche per garantire l'integrità di ciascun bit inviato. Si è notato però che la robustezza introdotta mediante la ridondanza da un encoder reale è migliore rispetto a quella che si avrebbe proteggendo con una FEC (*Forward Error Correction*) il bitstream di un encoder perfetto.

Quality of service: tecniche attive di robustezza

Finora abbiamo analizzato solamente tecniche di robustezza cosiddette “passive” in quanto vengono attuate dal ricevitore in modo assolutamente trasparente al trasmettitore. Parlando invece di tecniche attive si intendono quelle tecniche in cui il flusso di dati è modificato ad hoc per garantire la robustezza e basano il loro funzionamento principalmente su:

- invio di dati aggiuntivi
- modifica dell'ordine naturale dei dati
- algoritmi adattativi che pilotano una comunicazione

Vediamo ora nel dettaglio alcune tecniche:

1.Interleaving dei pacchetti

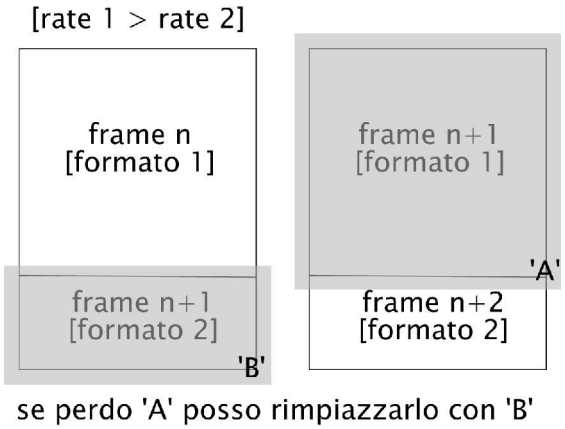
Dal momento che spesso gli errori sono presenti in burst conviene se il flusso multimediale è di tipo audio eseguire uno *scrambling* dei pacchetti in modo da rendere distribuito un errore che sarebbe invece concentrato. Se il flusso è di tipo video questa tecnica non è conveniente in quanto come abbiamo visto è meglio un frame molto danneggiato piuttosto che più frame poco danneggiati. Il prezzo da pagare per utilizzare questa tecncica è l'introduzione di un ritardo di trasmissione dipendente dall'estensione dello *scrambling*.

Interleaving dei pacchetti

2.Interleaving dei formati

In ogni pacchetto si invia oltre che il frame corrente anche uno o più successivi in un altro formato ad un bitrate inferiore in modo che se un frame viene perso è possibilie rimpiazzarlo con il suo corrispettivo a bassa qualità. Se si utilizza questa tecnica bisogna accettare l'ipotesi che sia percettivamente meglio il frame nel secondo formato piuttosto che lo stesso frame ricostruito sulla base dei precedenti e/o successivi.
I problemi nell'utilizzo dell'interleaving dei formati sono il ritardo, l'overhead (invio tutto il flusso in due formati), la complessità e infine il fatto che se si utilizzano tecniche parametriche (LPC) per il formato ausiliario il filtro di decodifica a valle partirà ogni qual volta si renderà necessario senza alcuna memoria dei campioni precedenti che sono stati decodificati con il decoder del formato principale.

Interleaving dei formati



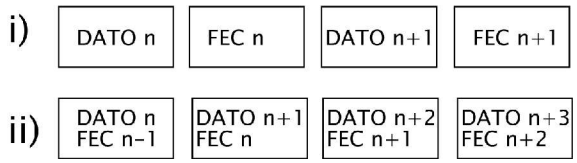
Alcuni possibili miglioramenti per questa tecnica sono:

- analisi per sintesi per scegliere tra la ricostruzione del frame perso e l'utilizzo del secondo formato (nel caso sia meglio il frame ricostruito il trasmettitore non invia nemmeno il secondo formato risparmiando banda).
- tecnica ibrida: la ricostruzione avviene sulla base del secondo formato.

3.Forward Error Correction (FEC)

Aggiunta di codici per la correzione degli errori per recuperare i pacchetti persi. Per essere efficace dati e relativo codice di protezione devono viaggiare su pacchetti differenti: un pacchetto di dati può però contenere il codice di protezione di altri dati. Tipicamente vengono utilizzati i codici di Reed-Solomon. Questa tecnica introduce del ritardo e dell'overhead nella trasmissione.

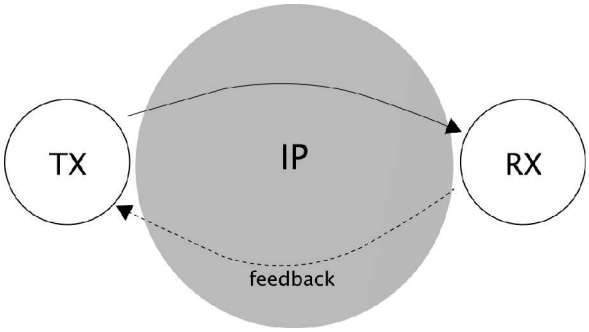
Forward Error Correction



4.Ritrasmissione selettiva

E' un tipo di *error correction* non *forward* nel senso che vengono inviati nuovamente solo i pacchetti che non sono arrivati integri (A*utomatic Repeat reQuest*).
Per utlizzare questa tecnica bisogna prevedere un canale aggiuntivo di ritorno dal ricevitore verso il trasmettitore detto *feedback* sul quale viaggerano i segnali i ACK/NACK tramite i quali il trasmettitore saprà se deve ritrasmttere qualche pacchetto.

ARQ – Automatic Repeat reQuest



I vantaggi nell'utilizzare questa tecnica sono:
- viene ritrasmesso solo quello che serve (usando la FEC i codici vengono inviati comunque anche se non servono)
- è relativamente semplice da implementare
- se abbiamo la possibilità di ritrasmettere almeno 3-4 volte è molto robusto (è la tecnica migliore per lo streaming dove grazie al playout buffer abbiamo un buon margine di ritardo).

I contro d'altra parte sono:
- overhead (per dati e segnalazione)
- bisogna utilizzare delle tecniche per proteggere ACK e NACK
- viene introdotto un ritardo di 1,5 round trip time per ogni ritrasmissione(non è una tecnica accettabile per il VoIP e la videoconferenza)

I parametri chiave di questa tecnica sono:
- il numero di ritrasmissioni (dipende dal round trip time)
- la frequenza di trasmissione dei segnali di ACK e NACK
- la robustezza della trasmissione di ACK e NACK tramite:
- duplicazione delle informazioni in pacchetti diversi
- segnalazione periodica al ricevitore dell'ultimo ACKed packet.
- ...

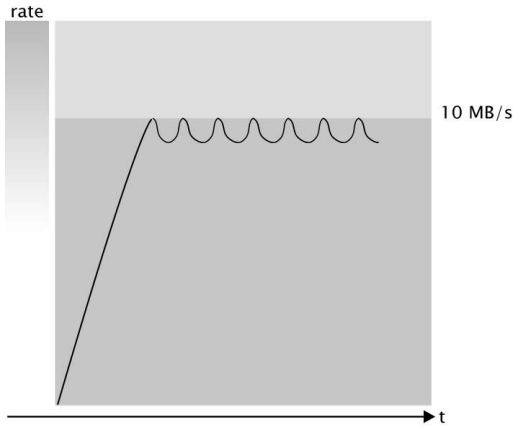
5.Rate-Adaptive Transmission

Come TCP...

Si cerca in pratica di introdurre un funzionamento TCP-like all'interno dei protocolli per i flussi multimediali. In particolare si vorrebbero ottenere dei flussi VBR (*variable bit-rate*) network driven :

- che automaticamente si dividano tra loro le risorse di rete (*fair use*)
- che automaticamente aiutino a decongestionare la rete quando serve
- Per questi scopi avremo due comportamenti del flusso a seconda dello stato della rete in quel momento:
- *additive rate increase* se la rete è scarica
- *moltiplicative rate decrease* se rilevo delle perdite di pacchetti.

Comportamento di una trasmissione rate-adaptive



...ma con alcune differenze

Il protocollo che si vuole progettare per i flussi multimediali deve avere anche delle grosse differenze rispetto a TCP:

- si devono evitare i casi di ritardo estremo di TCP
- TCP può andare anche a 0b/s se la rete è carica mentre se questa è scarica cerca di occuparla tutta (in pratica 0 < R(t) < capacità rete). Un protocollo per il multimedia dovrebbe avere un rate minimo diverso da 0 e un rate massimo dipendente dall'area di saturazione del codec(oltre non conviene più a meno di cambiare tipo di codec).

Vi sono principalmente due strade possibili per implementare questo comportamento a seconda di dove è posto l'algoritmo che sceglie momento per momento il rate a cui andare(sulla base dei dati contenuti nei Receiver Reports – *delay, jitter e packet losses*):

- algoritmo posto al ricevitore il quale può inviare dei comandi di rete al trasmettitore (se perdo un comando devo aspettare un lungo lasso di tempo prima che ne arrivi un altro e il rate rimane invariato). Con questa tecnica (utilizzata dal GSM-AMR):

1. i comandi sono più fragili (e quindi vanno protetti)
2. i dati sono perfetti

- algoritmo posto al trasmettitore che varia il rate sulla base dei Receiver Reports. In questo caso dal momento che sono i dati ad essere trasmessi e non i comandi:

1. i comandi sono perfetti
2. i dati possono essere potenzialmente corrotti

L'algoritmo può decidere di variare il bitrate seguendo due politiche differenti:

- rate in funzione del Packet Loss Rate: in pratica si utilizzano due soglie PLR1, sotto la quale il rate inizia a crescere e PLR2 sopra la quale inizia a decrescere.Il problema di questa tecnica è che il PLR potrebbe arrivare quando ormai la congestione è terminata
- rate in funzione del ritardo: il bitrate viene diminuito se si nota un aumento del ritardo (imputabile all'aumento delle code nei router). In questo modo è anche possibile prevenire ed evitare potenziali congestioni.

Quality of service: IntServ e DiffServ

Limiti del Best Effort

Nell'architettura Best Effort ogni pacchetto viene gestito in modo indipendente rispetto agli altri appartenenti alla stessa connessione e non esiste alcuna distinzione di priorità tra pacchetti appartenenti a connessioni diverse. Addirittura, le connessioni non esistono "logicamente" all'interno della rete, in quanto i router, per svolgere i loro compiti, non hanno bisogno di conoscerle. Quindi ogni connessione esiste solo per i due host alle sue estremità, che identificano tutti i pacchetti che si scambiano come appartenenti alla connessione stessa. Tali pacchetti, una volta usciti dall'host sorgente e prima di entrare in quello di destinazione, perdono la loro "reciproca parentela" e diventano entità indipendenti.
Di conseguenza, le risorse della rete sono assorbite in modo del tutto incontrollato dai vari flussi di pacchetti e le prestazioni ottenute variano in modo quasi casuale a seconda del livello momentaneo di congestione. Alcune connessioni sono involontariamente favorite dalla rete, mentre altre sono penalizzate.

Le architetture di tipo IntServ e DiffServ sono stati introdotte proprio per cambiare questa situazione e per dare la possibilità agli utenti (eventualmente paganti) di richiedere alla rete determinate **prestazioni garantite**.

IntServ

In una architettura IntServ (*Integrated Services over the Internet*), la conoscenza della singola connessione non è più limitata ai due host in comunicazione, ma viene estesa anche a tutti i router attraversati all'interno della rete. Grazie a un apposito protocollo di "segnalazione" (chiamato RSVP) gli utenti che desiderino usufruire di una connessione con qualità garantita possono farne richiesta alla rete; quest'ultima, se ha risorse disponibili, accetta la connessione e garantisce ai pacchetti che ne fanno parte un trattamento privilegiato rispetto a quelli che passano in modalità Best Effort. Secondo il modello IntServ, dunque, parte delle risorse della rete è usata per il traffico Best Effort (per servire chi non vuole pagare o non ha particolari requisiti in termini di risorse) ed il resto viene allocato a chi ne fa richiesta – ovviamente previa autorizzazione, autenticazione ed eventuale tariffazione.
Ovviamente l'introduzione dell'architettura IntServ richiede la modifica del software a bordo dei router per permettere loro di riconoscere le connessioni ed interpretare il protocollo di prenotazione.

Un grosso difetto di questo approccio è la mancanza di scalabilità: la gestione delle connessioni all'interno della rete, infatti, risulta essere eccessivamente onerosa per i router, che possono sopportare qualche decina di connessioni ma non le centinaia che presumibilmente si verrebbero a trovare su una dorsale di transito Internet.
Quindi, su una rete di piccole dimensioni con poche decine di utenti l'approccio IntServ è percorribile e, anzi, auspicabile: la tipica LAN aziendale, per esempio, beneficerebbe sicuramente della possibilità di dare priorità diverse a traffici diversi. In questo caso, inoltre, il numero di router da aggiornare e configurare sarebbe limitato. Per contro, in una rete di grandi dimensioni o su una dorsale difficilmente i router potrebbero sostenere la gestione di tutte le connessioni.

DiffServ

Per indirizzare il problema della scalabilità, IETF ha introdotto un nuovo modello di QoS chiamato DiffServ (*Differentiated Services over the Internet*).
Una prima sostanziale differenza rispetto e IntServ e la mancanza di connessioni all'interno della rete. Ogni pacchetto è nuovamente svincolato dagli altri facenti parte della stessa connessione e viaggia sulla rete in modo indipendente. Tuttavia, ad un livello di astrazione più alto, pacchetti appartenenti a diverse connessioni e caratterizzati da requisiti simili in termini di qualità richiesta vengono riconosciuti dai router come un unico gruppo e trattati di conseguenza.
Anche nel caso DiffServ il software di bordo dei router deve essere modificato, ma l'assenza delle singole connessioni alleggerisce di molto l'onere computazionale dovuto alla gestione della qualità. Inoltre la responsabilità sulla gestione dei pacchetti con qualità garantita è in qualche modo distribuita tra i diversi domini amministrativi attraversati. Nel caso degli IntServ, invece, l'uso del protocollo di prenotazione e l'esistenza della connessione da un host all'altro richiede che tutti i router attraversati siano configurati in modo "omogeneo", anche se nel caso di connessioni non locali essi appartengono ad entità amministrative diverse.

Nell'architettura prevista dai DiffServ, si distingue una regione della rete "ai margini" (edge) e una "al centro" (core). Il traffico proveniente dagli utenti arriva ai margini della rete, dove viene "trattato" e convogliato verso il centro. L'idea alla base dei DiffServ è l'aggregazione del traffico ai bordi della rete: i pacchetti che arrivano dagli host, indipendentemente dalla connessione cui appartengono ma tenuto conto dei loro requisiti in termini di risorse di rete (banda e ritardo), vengono raggruppati e "marcati" con un identificatore comune, che permetterà in seguito ai router nella parte centrale di applicare a tali pacchetti le più appropriate politiche di gestione.
In pratica, anziché far gestire alla parte interna della rete le singole connessioni (obbligando i router a riconoscere ogni singolo "micro flusso" di pacchetti), si gestiscono aggregati di connessioni (o "macro flussi") aventi caratteristiche simili. A grandi linee, si può dire che il principio con il quale il traffico viene gestito dalla rete è lo stesso nel caso IntServ e nel caso DiffServ, con la differenza che nel primo caso il controllo è fatto sulla singola connessione, mentre nel secondo caso è fatto su più connessioni considerate insieme.

La marcatura dei pacchetti e la distinzione tra flussi aggregati diversi è effettuata, rispettivamente, scrivendo ed esaminando un codice nel campo

TOS (Type Of Service), contenuto nell'header di ogni pacchetto IP. L'uso di questo campo, pensato originariamente proprio per distinguere tra pacchetti di natura diversa, è attualmente piuttosto limitato; alcuni router commerciali consentono di ottenere 8 diversi livelli di priorità in base al suo contenuto (una tecnica chiamata IP Precedence), senza tuttavia offrire il livello di flessibilità e funzionalità possibile con i DiffServ. I codici assegnati dai DiffServ sono però compatibili con IP Precedence e questo dovrebbe favorirne ulteriormente l'introduzione graduale.

Secondo il modello DiffServ, la rete nel suo complesso è formata da più sotto-reti, amministrativamente indipendenti, chiamate domini. Ciascun dominio fornisce un servizio ai propri clienti (gli utenti finali oppure altri domini) e richiede a sua volta ai domini adiacenti un servizio, che consiste nel mettere a disposizione una certa quantità di risorse per la gestione del proprio traffico. Gli accordi tra i domini adiacenti consentono agli aggregati di attraversare la rete usufruendo della qualità desiderata.

La scalabilità, punto debole degli IntServ, è certamente garantita con l'approccio DiffServ, grazie al limitato numero di aggregati che possono attraversare la rete di core. I router di core, infatti, possono operare a velocità molto maggiori dovendo gestire solo pochi flussi di traffico diversi. Maggiore carico è posto sui router di edge, ai quali è lasciato il compito di classificare i pacchetti e aggregarli marcandoli con il codice opportuno nel campo TOS. Tuttavia, il numero di connessioni che arrivano ad un router di edge è certamente limitato e molto inferiore a quello di connessioni gestite da un router di core. L'architettura appare pertanto ben bilanciata.

Il punto debole dei DiffServ è l'affidabilità con la quale le garanzie di QoS possono essere garantite, mancano infatti la segnalazione e la prenotazione dinamica delle risorse, che negli IntServ permettono una gestione molto più accurata della rete stessa.

In un certo senso il modello IntServ tende ad avvicinare il caotico mondo a commutazione di pacchetto, caratteristico di Internet, al più ordinato ed efficiente mondo delle comunicazioni a commutazione di circuito, tipico per esempio delle reti ATM e della rete telefonica. La complessità e il costo del secondo tipo di rete, ovviamente, è molto maggiore rispetto al primo.
Il punto debole del modello IntServ, in ultima analisi, è stato proprio cercare di ottenere i benefici della commutazione di circuito (dove il "circuito" è la "connessione" di IntServ) con apparecchiature nate per la commutazione di pacchetto, non abbastanza potenti per gestire su larga scala la rete così strutturata.
In questo contesto, i DiffServ si posizionano a metà strada tra i due approcci, in quanto tentano ancora di gestire dei "circuiti commutati", in questo caso rappresentati dagli aggregati di pacchetti, ma in modo meno complesso.

Metodologie di ricerca

Se occorre testare un nuovo software di streaming multimediale si deve tener conto di alcuni fattori in modo da avere una visione abbastanza realistica di come si comporterà in seguito su casi reali:

- Informazione Multimediale:

1. Approccio retistico:
    1. modello di sorgente (non segnali reali)
    2. modelli statistici di packet size e tempi di spedizione
    3. pro: semplice e potenzialmente accurato
    4. contro: non posso mappare PLR e *delay* in maniera percettiva.
2. Segnali reali presi da database noti codificati secondo standard recenti.

- Perdite e ritardi: si seguono i seguenti passi:

1. Error Insertion
2. Error Concealment
3. Decodifica
4. Calcolo della qualità
5. Il buffer di playout si immagina inizialmente infinito in modo da poterne simulare la dimensione in seguito in più modi differenti

- Modelli per la rete :

1. modelli analitici :
    1. grosso sforzo di progetto e validazione
    2. molto rapido l'uso
2. simulazioni:
    1. topologie standard di rete (ad esempio '*collo di bottiglia'*)
3. esperimenti:
    1. rete reale
    2. accesso a teconologia non standard
    3. non riproducibilità (nella maggior parte dei casi)

Indice