

**Università di Genova**  
**Facoltà di Ingegneria**

*Telematica 2*

**1. Servizi Multimediali e  
Qualità del Servizio (QoS) su IP**  
**1.5 Voce e Video su IP**

Prof. Raffaele Bolla



R. Bolla Telematica 2, n. o.

**Servizi multimediali e VoIP**

- I servizi multimediali realizzabili tramite tecnologia IP sono in qualche modo classificabili nell'ambito di due nomenclature usate oggi in modo piuttosto diffuso, che sono
  - **VoIP (Voice over IP),**
  - **Videoconferenza.**

## VoIP (Voce over IP)

---

- In senso generale col nome di VoIP si identificano tutti i servizi di comunicazione su IP che coinvolgono lo scambio interattivo di segnali audio ed eventualmente video e dati fra due interlocutori (*unicast*).
- L'acronimo descrive e richiama il servizio più semplice fra quelli compresi in questa categoria, ossia una comunicazione telefonica corrispondente a quella realizzabile tramite la rete PSTN .
- Altre tipologie di servizi più avanzati possono essere ad esempio la video-telefonata, l'interazione con call-center con supporti video e dati.

## Video-conferenza

---

- In questo contesto (TCP-IP), sotto il nome di video-conferenza vengono raccolti in genere i servizi che prevedo l'interazione in tempo reale audio, video e dati di tre o più utenti, si tratta in pratica della "versione" multicast del VoIP.
- Anche in questo caso si parte da servizi mono-media (solo voce) per arrivare alla versione più sofisticata che coinvolge scambi voce-video-dati.
- Due esempi sono la vera e propria video-conferenza, i servizi di didattica a distanza in tempo reale.

## VoIP – Tipologie di protocolli

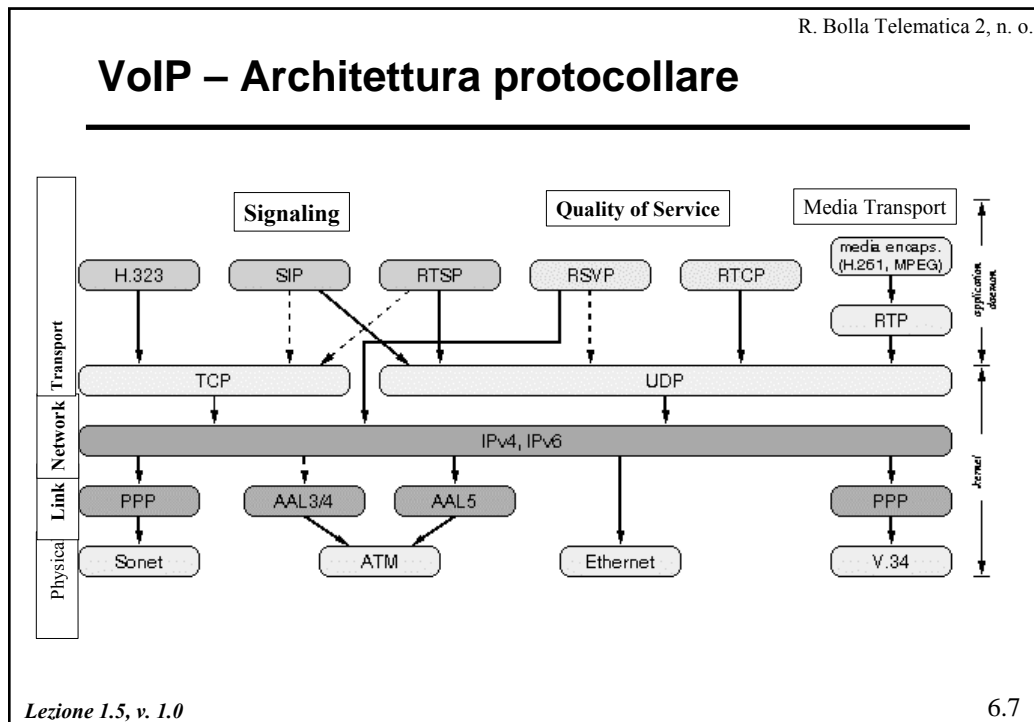
---

- Per realizzare applicazioni di VoIP sono necessari tre tipologie di protocolli:
  - **trasporto dei media:**  
servono per trasportare i veri e propri flussi multimediali (audio, video, immagini e dati) all'interno della rete;
  - **segnalazione:**  
necessari per stabilire la presenza degli utenti, localizzarli e instaurare, modificare e terminare le sessioni;
  - **supporto:**  
forniscono una serie di funzionalità "accessorie" quali traduzione degli alias in indirizzi IP, localizzazione dei gateway, QoS, AAA interdominio ...

## VoIP – Protocolli utilizzati

---

- Trasporto:
  - **RTP** (*Realtime Transport Protocol*) per i media (definito da IETF e adottato anche dall'ITU);
  - TCP/UDP per il trasporto all'interno della rete.
- Segnalazione:
  - **SDP** (*Session Description Protocol*) - IETF
  - **SIP** (*Session Initiation Protocol*) - IETF
  - **H.323** - ITU-T
- Supporto:
  - DNS (*Domain Name System*);
  - RSVP (*Resource Reservation Setup Protocol*);
  - COPS (*Common Open Policy Service*), definisce politiche di controllo per QoS;



R. Bolla Telematica 2, n. o.

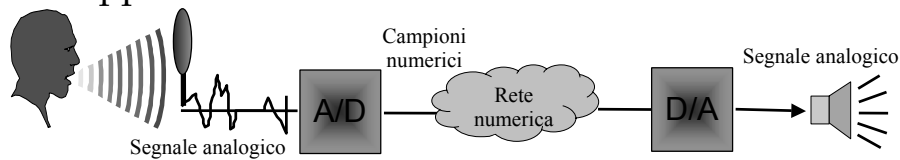
## Trasporto di flussi multimediali su IP

- La trasmissione dei diversi media richiede una serie di operazioni fondamentali
  - acquisizione dei segnali (microfoni, telecamere, tastiere);
  - **codifica dei segnali** ←
  - » conversione analogico/digitale,
  - » rappresentazione in una codifica binaria;
  - » compressione.
  - trasmissione dell'informazione
  - decodifica dei segnali
    - » conversione digitale/analogica;
  - riproduzione del media (altoparlanti, monitor, stampanti, ...).

*Lezione 1.5, v. 1.0* 6.8

## Media - Voce

- Rappresenta il media fondamentale per le applicazioni di telefonia su IP.



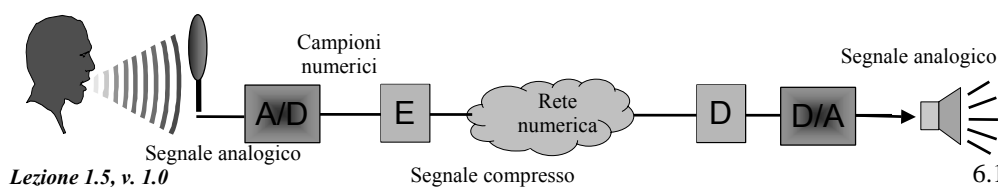
- Lo spettro del segnale vocale è di circa 4 KHz
  - in base del teorema di Nyquist è necessario campionare il segnale analogico ad almeno 8 kHz;
  - il risultato sono 8000 campioni/s.
- Il segnale analogico viene codificato tramite PCM a 16 bit
  - usare 16 bit a campione rende l'errore di quantizzazione sicuramente impercettibile all'orecchio umano.

Lezione 1.5, v. 1.0

6.9

## Media - Voce Codec

- Un'applicazione di questo tipo genera un flusso dati a 128 kbps ( $8000 \text{ camp/s} \times 16 \text{ bit/camp}$ ).
- L'utilizzo di un codec permette di ridurre il tasso trasmissivo mantenendo un livello adeguato di qualità del segnale
  - è composto da un codificatore (E) e un decodificatore (D)
    - » il codificatore comprime i campioni numerici rimuovendo l'informazione ridondante (vedi teoria dell'informazione);
    - » il decodificatore ricostruisce il segnale originale a partire da quello compresso.



Lezione 1.5, v. 1.0

6.10

## Media – Voce

### Codec

---

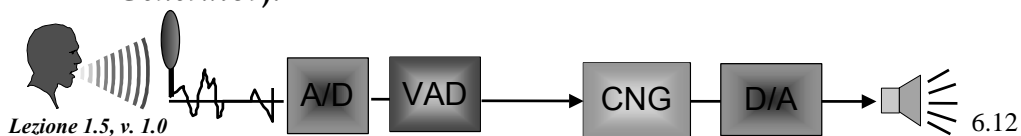
- Esistono due tipologie di codificatori:
  - **voce**
    - » sfruttano le caratteristiche della voce umana permettendo di raggiungere elevati fattori di compressione.
  - **audio**
    - » coprono una ampia gamma dello spettro e di tipologie di suoni ma raggiungono fattori di compressione minori del caso precedente;

## Media – Voce

### Codec

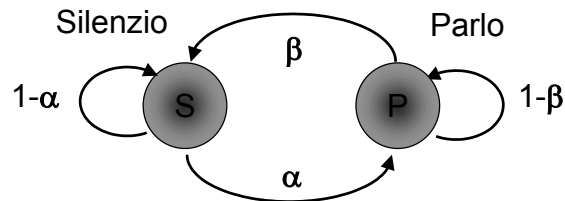
---

- Sono varie le caratteristiche della voce che contribuiscono a rendere efficiente (e quindi vantaggiosa) l'operazione di compressione.
- Una di esse è data dal fatto che una sorgente vocale alterna fasi di attività (parlato) a fasi di inattività (silenzio) in cui è inutile trasmettere. Pertanto si può prevedere
  - In trasmissione l'utilizzo di **Voice Activity Detector (VAD)** permette di riconoscere i momenti di silenzio e annullare la trasmissione durante il loro svolgersi;
  - in ricezione, dato che la completa assenza di segnale appare innaturale, l'uso di generatori di rumore **CNG (Confort Noise Generator)**.



## Media – Voce Codec

R. Bolla Telematica 2, n. o.



- Durata media del periodo attività (*talkspurt*) = 1.00 s
- Durata media del periodo di silenzio (*gap*) = 1.35 s
- Percentuale di attività  $\approx 42\%$
- Questo modello in pratica riconosce solo le pause fra frasi o al massimo parole. Modelli (e SAD) più complessi sono in grado di riconoscere e sfruttare anche le micro-pause fra parole e fra sillabe ed ottenere livelli di efficienza ancora superiori

Lezione 1.5, v. 1.0

6.13

## Media – Voce Codec

R. Bolla Telematica 2, n. o.

- Per permettere lo svolgimento di comunicazioni vocali efficaci è ovviamente necessario che le tecniche di compressione vengano inquadrare in uno standard riconosciuto.
- L'ente che ha fatto questo nel campo della voce è principalmente ITU-T, con gli standard
  - G.711
  - G.723.1
  - G.726
  - G.728
  - G.729
  - G.729A

Lezione 1.5, v. 1.0

6.14

## Media – Voce

### Codec ITU-T

---

- G.711

- Ampiamente utilizzato nelle reti telefoniche e noto come  $\mu$ -law (America e Giappone) e A-law (Europa);
- prevede la trasformazione di campioni PCM lineari a 14 ( $\mu$ -law) o 13 bit (A-law) in campioni codificati esponenzialmente a 8 bit;
- produce una bit rate di 64 kbps.

## Media – Voce

### Codec ITU-T

---

- G.723.1

- produce in uscita pacchetti di dimensione variabile:
  - » a partire da blocchi di 240 campioni (30 ms) codificati a 16 bit/camp. produce pacchetti di 24 byte, tramite l'algoritmo *Multi-Pulse Maximum Likelihood Quantization* (MP-MLQ), ottenendo una bitrate di 6.3 kbps;
  - » a partire dagli stessi ingressi produce pacchetti di 20 byte, utilizzando l'algoritmo *Algebraic Code-Excited Linear Prediction* (ACELP), ottenendo una *bitrate* di 5.3 kbps;
  - » Genera pacchetti di 4 byte nei periodi di silenzio per specificare i parametri da utilizzare per il CNG;
- è possibile variare la codifica alla fine di ogni blocco corrispondente a 30 ms di conversazione;
- lo standard definisce anche gli algoritmi per il VAD e il CNG.



## Media – Voce

### Codec ITU-T

---

- **G.726**

- utilizza la codifica *Adaptive Differential Pulse Code Modulation* (ADPCM) per comprimere i campioni a 8 bit di una codifica G.711 a 2, 3, 4 o 5 bit, ottenendo bitrate pari a 16, 24, 32 o 40 kbps.

- **G.728**

- utilizza la codifica *Low Delay – Code-Excited Linear Prediction* (LD-CELP) per comprimere blocchi di 5 campioni (0.625 ms) a 16 bit in campioni a 10 bit a velocità di 16 kbps.

## Media – Voce

### Codec ITU-T

---

- **G.729**

- utilizza la codifica *Conjugate Structure – Algebraic Code-Excited Linear Prediction* (CS-ACELP) per comprimere blocchi di 80 campioni (10 ms) a 16 bit in pacchetti di
  - » 10 byte (8 kbps)
  - » 2 byte, per i parametri del CNG.
- G.729 specifica anche gli algoritmi per il VAD ed il CNG.

**Media – Voce****Caratterizzazione dei Codec**

---

- **Bit rate:** un minor bitrate permette di multiplexare più flussi nella stessa banda e di ottenere una migliore QoS.
- **Complessità:** si misura in milioni di istruzioni al secondo (MIPS). La complessità di un algoritmo influenza elementi quali la memoria (ROM o RAM) necessaria, la potenza della CPU, la potenza di alimentazione necessaria.

**Media – Voce****Caratterizzazione dei Codec vocali**

---

- **Ritardo** composto da
  - *ritardo dell'algoritmo, che è legato alla:*
    - » dimensione del blocco: è necessario collezionare un certo numero di campioni prima di effettuare la codifica (es. 30 ms per G.723.1 e 10 ms per G.729);
    - » ritardo look-ahead: alcuni codificatori necessitano di conoscere anche alcuni campioni successivi al blocco in fase di compressione (es. 7.5 ms per G.723.1 e 5 ms per G.729);
  - *ritardo di elaborazione*: dipende dalla complessità dell'algoritmo e dalla potenza dell'hardware a disposizione.

**Media – Voce****Caratterizzazione dei Codec vocali**

- *Qualità del segnale*  
tipicamente misurata in termini di *Mean Opinion Score* (MOS), con un punteggio soggettivo crescente da 1 a 5.
- *Altre...*  
Ad esempio la capacità dei codec di trasportare correttamente i toni DTMF (*Dual-Tone Multi-Frequency*), particolarmente utili nei sistemi interattivi automatici (IVR, *Interactive Voice Response*), o la capacità di trasporti di segnali FAX.

**Media – Voce****Caratterizzazione dei Codec**

<i>Codec</i>	<i>Bitrate (kbps)</i>	<i>Complessità (comparata a G.726)</i>	<i>Ritardo Algoritmico (ms)</i>	<i>Qualità (MOS)</i>
G.711	64	molto bassa	0.125	4
G.723.1	5.3	8	37.500	3.9
G.723.1	6.3	8	37.500	3.6
G.726	32	1	0.125	3.9
G.728	16	15	0.625	3.6
G.729	8	10	15.000	3.9
G.729 A	8	6	15.000	3.7

**Media – Voce****Un esempio di trasmissione di voce su IP**

- Si consideri un'applicazione che genera dati con tasso pari a 64 kb/s (8000 campioni/s, ognuno di 8 bit).
- Questa significa che l'applicazione, ogni 20 ms genera un blocco di 160 Bytes ( $64 \text{ kb/s} / 8 \times 20 \text{ ms}$ ).
- Al blocco si aggiunge una intestazione ed il tutto viene inserito in un pacchetto UDP ed inviato in rete.
- Il ricevitore riceve questi pacchetti, ma non tutti, perché alcuni vanno persi, e con ritardo variabile (eventualmente anche fuori sequenza).
- Il ricevitore deve decidere quando riprodurre un blocco e cosa fare se mancano dei blocchi.

**Media – Voce****Un esempio di trasmissione di voce su IP**

- Sono due i problemi principali che vanno risolti
  - Ritardo
    - » Il ritardo *end to end* deve essere inferiore a 300 ms, altrimenti ne risente l'interattività.
    - » Il *jitter* in questo caso è dato dal fatto che i pacchetti sono trasmessi a 20 ms uno dall'altro ma possono arrivare al ricevitore sia più vicini che più lontani.
    - » Si dovrebbe comunque inserire dei *timestamp* per capire quando un blocco dati debba essere riprodotto.
  - Perdite di pacchetti:
    - » Si potrebbe usare TCP, ma la ritrasmissione introduce ritardo e il controllo di flusso limita il tasso.
    - » Se si usa UDP, in ogni modo, bisogna inserire numeri di sequenza per accorgersi di eventuali perdite.

R. Bolla Telematica 2, n. o.

### Media – Voce

## Un esempio di trasmissione di voce su IP

The diagram illustrates the voice transmission process over IP. It starts with a microphone (Segnale analogico) connected to an A/D converter, which outputs numerical samples (Campioni numerici). These samples are processed by an Encoder, resulting in a compressed signal (Segnale compresso). This signal is transmitted through a network (Rete numerica), incurring a transmission delay ( $d_t$ ). The signal then passes through a De-jitter buffer and a Decoder, which outputs an analog signal (Segnale analogico) to a speaker. The total delay from the microphone to the speaker is labeled as the end-to-end delay ( $d_{end-to-end}$ ).

- I ritardi del microfono, dei convertitori A/D e D/A e degli altoparlanti sono trascurabili.
- Il ritardo dei coder ( $d_{coder}$ ) è dovuto a
  - all'encoder
    - » ritardo algoritmico
    - » ritardo di elaborazione
  - al decoder
    - » ritardo di elaborazione

Lezione 1.5, v. 1.0 6.25

R. Bolla Telematica 2, n. o.

### Media – Voce

## Un esempio di trasmissione di voce su IP

- Il ritardo di trasmissione ( $d_t$ ) dipende dalla rete e dal traffico presente
  - il ritardo di trasmissione è composto dal ritardo della rete ( $d_{net}$ ) e dalle sue variazioni (*jitter*,  $j_{net}$ ):
 
$$d_t = d_{net} + j_{net}$$
- Il ritardo *end-to-end* può quindi essere caratterizzato da queste due componenti:
 
$$d_{end-to-end} = d_{coder} + d_t < 300 \text{ ms}$$

Lezione 1.5, v. 1.0 6.26

**Media – Voce****Un esempio di trasmissione di voce su IP**

- A destinazione
  - il ritardo, se limitato, non influenza la ricezione;
  - il jitter, al contrario, degrada notevolmente la qualità del segnale ricevuto.
- Per questa ragione conviene introdurre un ulteriore ritardo in modo da eliminare i jitter.

**Media – Voce****Compensazione del jitter**

- Il *Jitter* viene in genere compensato utilizzando un buffer ed inserendo un ritardo  $q$  di riproduzione, ossia:
  - Se un blocco con un *timestamp*  $t$  è ricevuto entro  $t + q$ , viene riprodotto all'istante  $t + q$
  - Altrimenti viene scartato
- Questa strategia non richiede numeri di sequenza per gestire le perdite
- Il problema è fissare  $q$ 
  - $q$  lunghi implicano meno perdite (per *jitter*);
  - $q$  corti: l'interazione è più efficace.

R. Bolla Telematica 2, n. o.

### Media – Voce

## Compensazione del jitter

---

- La sorgente genera un pacchetto ogni 20 ms.
- Il primo pacchetto è ricevuto all'istante  $t$ .
- Nel primo caso (ritardo  $q$ ) è riprodotto a partire dall'istante  $r$ .
- Nel secondo caso (ritardo  $q'$ ) è riprodotto a partire dall'istante  $r'$ .

Lezione 1.5, v. 1.0

6.29

R. Bolla Telematica 2, n. o.

### Media – Voce

## Compensazione del jitter

---

- Il meccanismo di compensazione del jitter può essere reso più efficace variando in modo adattativo il ritardo di riproduzione.
- In particolare questo si realizza
  - Stimando il ritardo introdotto dalla rete (ed eventualmente la sua varianza).
  - Variando l'istante di inizio di ogni periodo di *parlato* (quando la sorgente è nello stato *parlo*), ossia comprimendo o allungando i periodi di silenzio (quelli durante i quali la sorgente è nello stato *silenzio*).
  - Durante i periodi di *parlato* (*talk spurt*) la riproduzione avviene come nel caso di ritardo fisso.

Lezione 1.5, v. 1.0

6.30

## Media – Voce

### Compensazione del *jitter*

---

- Indicando con
  - $t_i$  = il *timestamp* dell'*i*-esimo pacchetto
  - $r_i$  = l'istante in cui l'*i*-esimo pacchetto viene ricevuto
  - $p_i$  = l'istante in cui l'*i*-esimo pacchetto viene riprodotto
  - $d_i$  = il ritardo medio di rete stimato dopo la ricezione dell'*i*-esimo pacchetto
  - $v_i$  = la deviazione media del ritardo dopo la ricezione dell'*i*-esimo pacchetto
- Si può scrivere:
 
$$d_i = (1 - u)d_{i-1} + u(r_i - t_i)$$

$$v_i = (1 - u)v_{i-1} + u | r_i - t_i - d_i |$$

↙ Costante < 1 (ad es. 0,01)
- I valori  $d_i$  e  $v_i$  sono calcolati ogni pacchetto ricevuto, anche se vengono usati solo all'inizio di ogni periodo di *parlato*.

## Media – Voce

### Compensazione del *jitter*

---

- Usando questa quantità si può iniziare a riprodurre un *talk-spurt* all'istante:
 
$$p_i = t_i + d_i + Kv_i$$
- Per ogni singolo pacchetto il ritardo dovrà essere:
 
$$q_i = p_i - t_i = d_i + Kv_i$$
- E quindi l'istante di riproduzione di un pacchetto *r* all'interno del *talk-spurt* diventa
 
$$p_r = t_r + q_i$$
- Per determinare l'inizio di un *talk-spurt* si deve verificare se la differenza fra due successivi *timestamp* con il numero di sequenza (che qui diventa necessario per non farsi ingannare dalle perdite di pacchetti) in ordine corretto è > di 20 msec.



**Media – Voce****Compensazione del jitter**

- La compensazione del jitter ha comunque dei limiti

$$d_{\text{end-to-end}} < 300 \text{ ms} \quad (1)$$

- Si ricordi che:

$$d_{\text{end-to-end}} = d_{\text{coder}} + d_t = d_{\text{coder}} + d_{\text{net}} + j_{\text{net}} \quad (2)$$

- In ricezione, la riproduzione di un *frame* attenua l'effetto dei jitter:

$$\text{i. } \max(j_{\text{net}}) \leq fs \Rightarrow j_{\text{buf}} = 0$$

$$\text{ii. } \max(j_{\text{net}}) > fs \Rightarrow j_{\text{buf}} = \max(j_{\text{net}}) - fs$$

**Media – Voce****Compensazione del jitter**

- Le precedenti equazioni consentono di ricavare un limite alla compensazione dei jitter in ricezione:

$$d_{\text{end-to-end}} = d_{\text{coder}} + d_{\text{net}} + j_{\text{net}} < 300 \text{ ms}$$

$$d_{\text{coder}} + d_{\text{net}} + j_{\text{buf}} + fs < 300 \text{ ms}$$

$$j_{\text{buf}} < 300 \text{ ms} - d_{\text{coder}} - d_{\text{net}} - fs$$

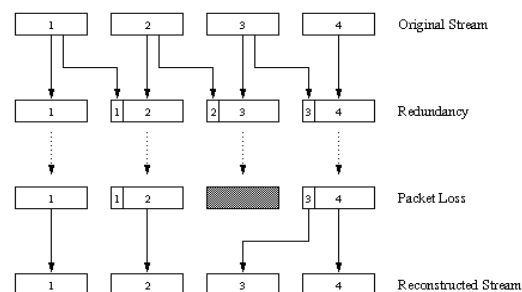
- Considerando un coder G.723.1,  $d_{\text{coder}} \approx 42.5$  ms,  $fs = 30$  ms e  $d_{\text{net}} = 60$  ms, è possibile assorbire i jitter della rete fino a 167.5 ms.

**Media – Voce****Compensazione delle perdite**

- ***Forward Error Correction (FEC)***
  - Siccome la perdita significa il non arrivo del pacchetto, le tecniche FEC devono coinvolgere più di un pacchetto.
  - Possiamo fare due esempi di possibili strategie:
    - **Strategia 1**
      - » Ogni n blocchi, si costruisce un blocco ridondante ottenuto come XOR degli n precedenti e lo si invia in coda al gruppo
      - » Se perdo un blocco solo degli n+1, all'istante di ricezione del blocco n+1 lo posso ricostruire
      - » La banda aumenta di un fattore 1/n (ad es. 1/10 se n= 10)
      - » Il ritardo fisso iniziale d deve essere tale da permettere la ricezione di almeno n+1 pacchetti (se n = 10, d > 200 msec).
      - » Quindi aumentando n aumenta l'efficienza ma aumenta anche il ritardo e la probabilità di avere più di una perdita in una sequenza
- Lezione 1.5, v. 1.0 (situazione in cui questo meccanismo diventa inefficace).* 6.35

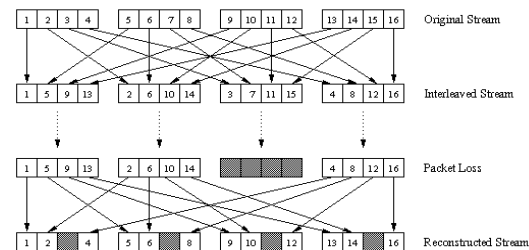
**Media – Voce****Compensazione delle perdite**

- **Strategia 2**
  - » Si comprime i dati usando due codifiche a risoluzioni diverse: una bassa (B) ed una alta (A) (ad es. un PCM a 64 Kbit/s e un GSM a 13 Kbit/s).
  - » Si costruisce il pacchetto n prendendo il blocco n del flusso codificato con A (160 Byte) e il blocco n-1 del flusso codificato con B (32,5 byte)
  - » Quando un pacchetto viene perso può essere ricostruito (con minore qualità) all'arrivo de pacchetto successivo
  - » Si può proteggere anche nei confronti di due perdite consecutive aggiungendo anche il blocco n-2 del flusso B.



**Media – Voce****Compensazione delle perdite**● ***Interleaving***

- I blocchi sono spezzato in unità più piccole (ad esempio in 4 unità da 40 byte).
- Le unità vengono mescolate in pacchetti successivi che quindi contengono parti di più blocchi.
- I blocchi originali vengono ricostruiti al ricevitore.
- Se un pacchetto viene perso la perdita viene distribuita su n blocchi e quindi è più facilmente compensabile.



- Ricostruzione pacchetto/unità persa
  - Si usano dei dati che assomigliano ai precedenti (ripetizione o interpolazione)
  - Funziona abbastanza bene per perdite rare e pacchetti/unità piccole (4-40 msec)

**Media - Testo**

- Oggi lo scambio di testo è molto diffuso (es. chat)
  - il testo può essere generato da una tastiera o da appositi dispositivi convertitori della voce o riconoscitori della scrittura.
- Un esempio di coder del testo è T.140 dell'ITU-T
  - utilizza la codifica ISO-IEC 10646-1 level 3 e ISO-IEC 6429 per rappresentare i caratteri alfabetici e le funzioni di controllo;
  - non prevede la compressione;
  - richiede al protocollo di trasporto di mantenere l'ordine dei dati inviati.

## Media - Video

---

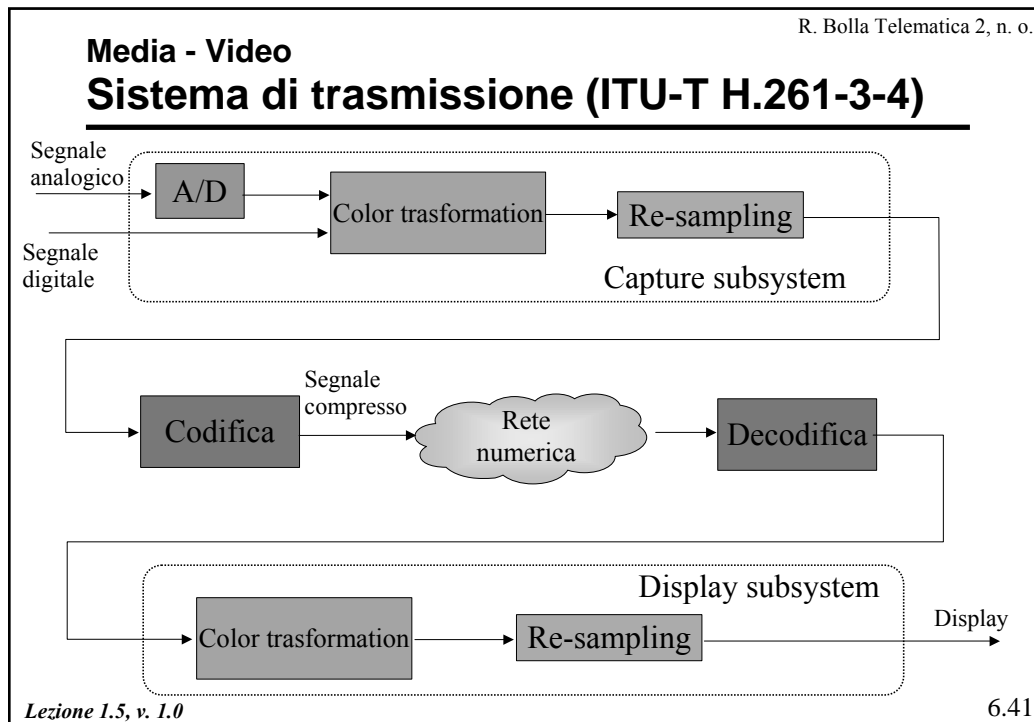
- La trasmissione di video in rete richiede di acquisire, codificare e trasportare sulla rete il segnale video.
- Le fasi richieste sono analoghe a quelle viste per il segnale vocale:
  - acquisizione;
  - **codifica** ←
  - trasporto;
  - decodifica;
  - visualizzazione.

## Media - Video

### Codec

---

- In linea di massima gli standard attuali di codifica video sono:
  - H.261 (obsoleto)
  - H.263
  - H.263 versione 2
  - H.264 (MPEG4 part 10- AVC, Advanced Video Coding)



**Media - Video****Sistema di trasmissione (ITU-T H.261-3-4)**

- L'occhio umano è meno sensibile alle variazioni di colore piuttosto che alla luminanza, per cui i codificatori H.261/H.263 richiedono un sottocampionamento 4:2:0 (le due componenti di crominanza sono sottocampionate della metà, in verticale ed orizzontale, rispetto alla luminanza)
  - i livelli di crominanza sono dunque  $360 \times 240$  ciascuno.
- I codificatori H.261/H.263 accettano in ingresso solo formati multipli o frazionari di un *Common Intermediate Format (CIF)*
  - $352 \times 288$  [pixel/line  $\times$  linee] per la luminanza,
  - $176 \times 144$  [pixel/line  $\times$  linee] per la crominanza.

Lezione 1.5, v. 1.0

6.43

**Media - Video****Sistema di trasmissione (ITU-T H.261-3-4)**

- Inoltre è richiesto che questi pixel siano rappresentati su una superficie 4:3.
- H.261 accetta:
  - CIF, QCIF (Quarter CIF,  $176 \times 144$ ).
- H.263/4 accetta:
  - CIF, SQCIF ( $128 \times 96$ ), QCIF, 4CIF ( $704 \times 576$ ) e 16CIF ( $1408 \times 1152$ ) per la prima versione,
  - qualsiasi dimensione multipla di 4, a partire da  $4 \times 4$  fino a  $2048 \times 1152$  nella seconda (H.263+ e H.264).
- Tale formato non è ancora compresso
  - un video a 10 frame/s, QCIF e 12 bit/pixel richiede una banda di 4.5 Mbps.
- I tassi trasmissivi ottenibili dopo la compressione sono compresi fra 10 Kbps e 2 Mbps con H.263.

Lezione 1.5, v. 1.0

6.44

**Media - Video****Sistema di trasmissione (ITU-T H.261-3-4)**

---

- In ricezione, il sottosistema di visualizzazione sovra-campiona le componenti di crominanza (in modo da avere lo stesso numero di pixel della luminanza) e convertono la rappresentazione dei colori e la risoluzione dei quadri in un formato compatibile con l'hardware a disposizione.

**Media - Video****Sistema di trasmissione (ITU-T H.261-3-4)**

---

- Per questa ragione i codificatori H.261 e H.263 prevedono la compressione del segnale prima della trasmissione.
  - Il flusso generato prevede la compressione dei *frame* in diversi modi:
    - » I-frame (Intra-frame): se la codifica avviene in modo indipendente dagli altri frame;
    - » P-frame (Predictive) e B-frame (Bidirectional) se la codifica prende in considerazione più *frame*, operando una compensazione del movimento.