Capitolo 5
Reti

- **Canale**
  - Buffering
  - Correzione errori
- **Congestione**
  - FIFO, FQ, WFQ, RED
- **QoS**
  - priorità, leaky bucket, token bucket, IntServ, DiffServ
- **RTP**
- **ATM**

Canale

- Come le applicazioni multimediali sono sensibili a **RITARDI** e **PERDITE** dovute alle caratteristiche del canale?
  - Audio e video → ritardo end-to-end e jitter
  - Testo e immagini → errori sul canale

- **BUFFERING**
  - Meccanismo per rifurre il jitter
  - Al ricevitore, invece di ripodurre immediatamente i pacchetti in ingresso, li si accoda in un buffer e si procede alla loro decodifica solo quando la coda è sufficientemente piena
  - Aumenta il ritardo complessivo ma si assorbono le differenze di ritardi tra pacchetti successivi, si assorbe il jitter

Buffering

- Difficile la valutazione del ritardo tra ingresso e uscita al buffer, causa scarsa conoscenza della statistica del traffico → STIMA DI PARAMETRI tramite MISURA sul sistema
- si = istante generazione pacchetto i-esimo
- ri = istante di ricezione = istante ingresso al buffer
- pi = istante in cui viene passato al sistema di decodifica
- di = ritardo
- vi = deviazione standard del ritardo

$$\begin{aligned} d_i &= (1-u)d_{i-1} + u(r_i - s_i) \\ v_i &= (1-u)v_{i-1} + u|r_i - s_i - d_i| \end{aligned}$$

$$p_i = s_i + d_i + Kv_i$$

Buffering – caso uniforme

- Ipotesi:
  - Ritardo istantaneo distribuito uniformemente in [m-Δ,m+Δ]
    - m=media del ritardo,
    - Δ=variazione tra ritardo istantaneo e ritardo medio
  - Densità di probabilità del ritardo in ingresso al buffer = $p_{in}(z)$
    - $p_{in}(z) = 1/(2\Delta)$ se z è in [m-Δ,m+Δ]
    - $p_{in}(z) = 0$ altrimenti
- Varianza al decodificatore senza buffer

  $\sigma^2 = E[z^2] - (E[z])^2 = E[z^2] - m^2$

$$\sigma_{in}^2 = \frac{1}{2\Delta}\int_{m-\Delta}^{m+\Delta} z^2 dz - m^2 = \frac{\Delta^2}{3}$$

- Con un buffer costruito in modo da rilasciare il pacchetto all'istante ri+m+Δ il ritardo aumenta di Δ ma la densità di probabilità del ritardo risulterebbe $p_{out}(z)=\delta(z-m-\Delta)$ quindi con $\sigma^2=0$
- Il sistema riduce ed annulla il jitter
- Esempio ideale (ipotesi di z uniforme)

Buffering – caso gaussiano

- Ipotesi:
  - Modello di ritardo più realistico: $$p_{in}(z) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(m-z)^2}{2\sigma^2}}$$
  - Varianza al dec senza buffer → σ^2
  - Con buffer che fornisce pacchetto in ingresso al decodificatore all'istante ri+mi+Δ

$$p_{out}(z) = \begin{cases} \alpha\delta(z-m-\Delta) + p_{in}(z) & \text{se } z \geq m+\Delta \\ 0 & \text{altrimenti} \end{cases}$$

$$\alpha = \int_{-\infty}^{m+\Delta} p_{in}(z)dz = \frac{1}{2}\left[1 + \text{ erf } \frac{\Delta}{\sqrt{2\sigma^2}}\right]$$

$$m_{out} = \alpha(m+\Delta) + \int_{m+\Delta}^{\infty} z p_{in}(z)dz =$$

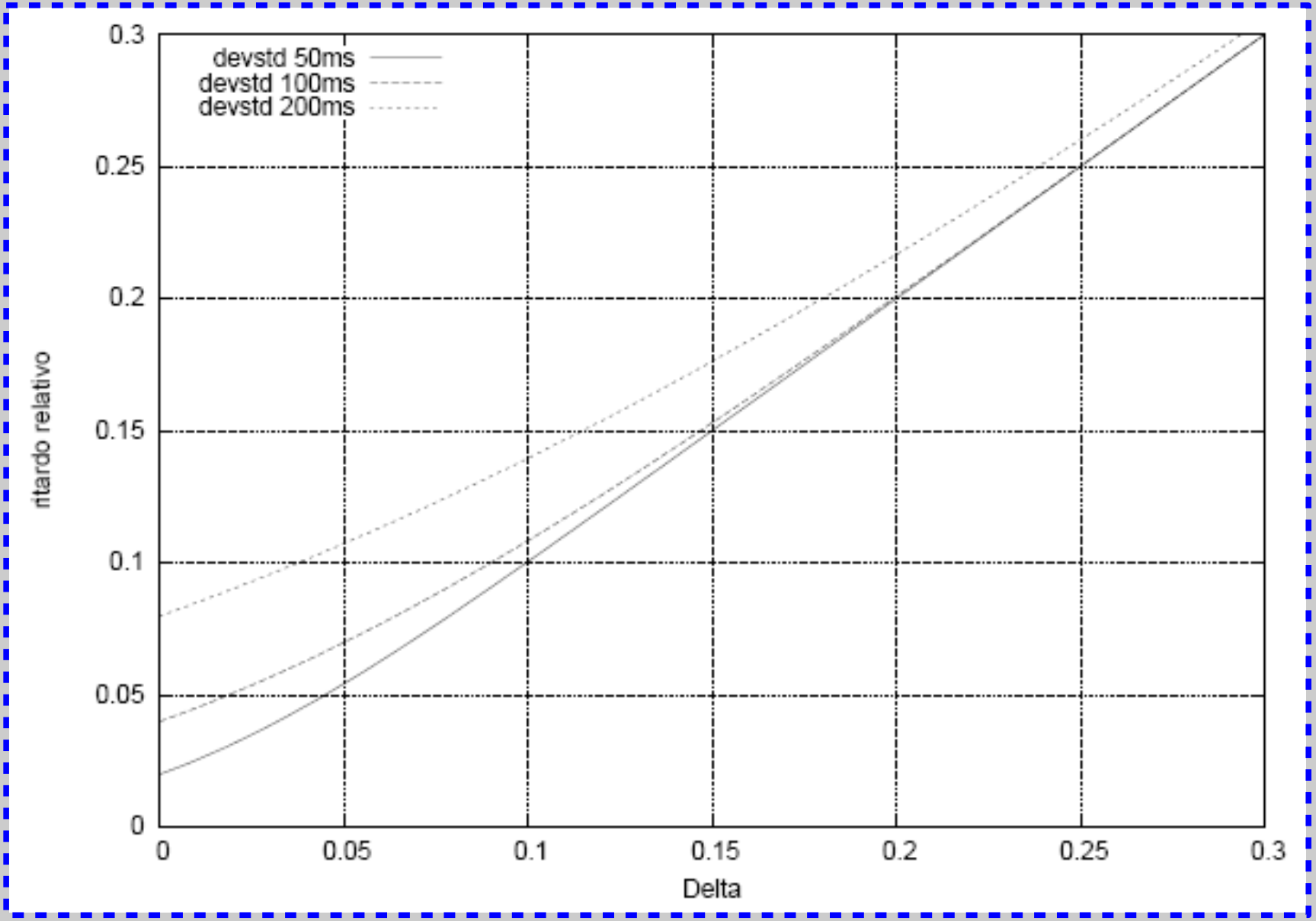
$$= m + \Delta + \frac{\sigma e^{-\frac{\Delta^2}{2\sigma^2}}}{\sqrt{2\pi}} - \frac{1}{2}\Delta \text{ erfc } \frac{\Delta}{\sqrt{2\sigma^2}}$$

$$\sigma^2_{out} = \alpha(m+\Delta)^2 + \int_{m+\Delta}^{\infty} z^2 p_{in}(z)dz - m^2_{out} =$$

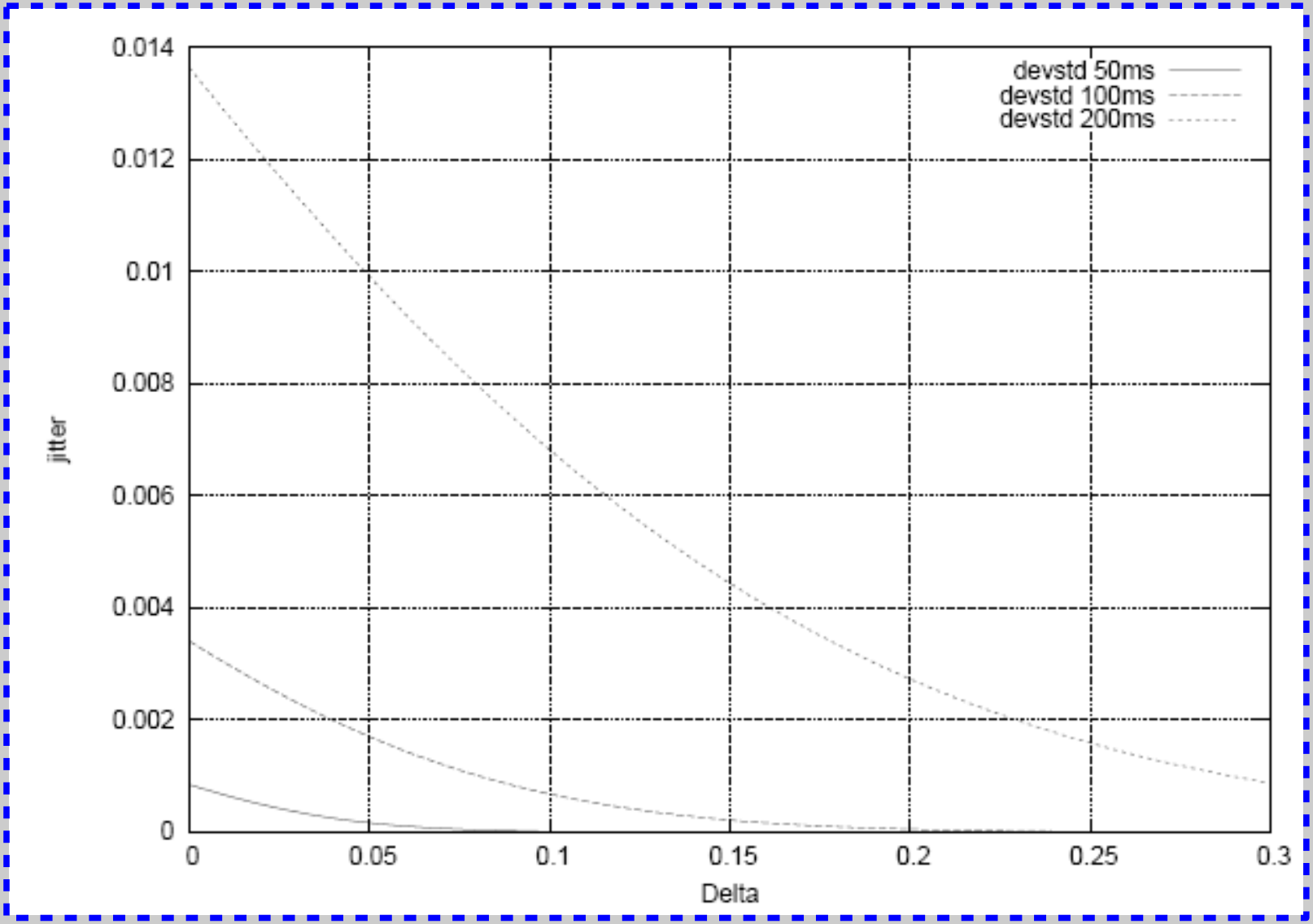
$$= \frac{1}{4}\Big[\Delta^2 + 2\sigma^2 - \frac{2\sigma^2 e^{-\frac{\Delta^2}{\sigma^2}}}{\pi} +$$

$$+ \left(-2\Delta\sigma\sqrt{\frac{2}{\pi}}e^{-\frac{\Delta^2}{2\sigma^2}} - 2\sigma^2\right) \text{ erf } \frac{\Delta}{\sqrt{2\sigma^2}} - \Delta^2 \text{ erf }^2 \frac{\Delta}{\sqrt{2\sigma^2}}\Big]$$

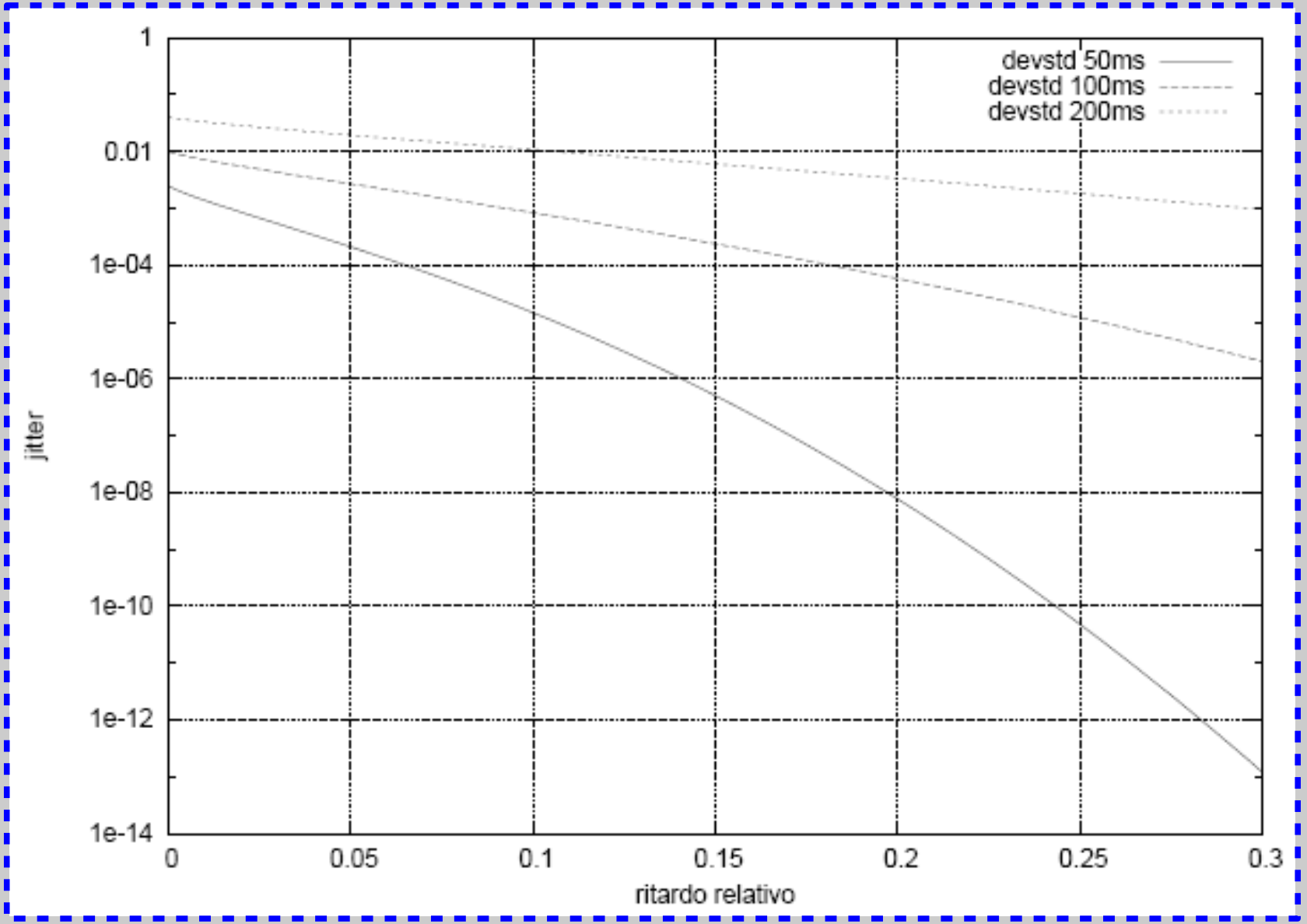
Buffering – caso gaussiano



Buffering – caso gaussiano



Buffering – caso gaussiano



Correzione errori

- Pacchetti persi
- ARQ e FEC
- FEC → ricostruzione di pacchetti completamente persi nella rete
  - **N-way Xor**
  - **Compressione a differente bit rate**
  - **Interleaving**
  - **Protezione differenziata**

N-way XOR

- Regole operatore xor
- Costruzione di una parola di parità ogni n parole di informazione
- Possibilità di recuperare un pacchetto ogni n
- Costruzione parola di parità

$$P_{\lfloor \frac{i}{n} \rfloor, j} = \oplus_{k=0}^{n-1} B_{\lfloor \frac{i}{n} \rfloor n+k, j} \quad j = 0, \cdots, L-1$$

- Ricostruzione parola persa

$$B_{v,j} = P_{\lfloor \frac{v}{n} \rfloor, j} \oplus \oplus_{k=0, \lfloor \frac{k}{n} \rfloor \neq \lfloor \frac{v}{n} \rfloor}^{n-1} B_{\lfloor \frac{v}{n} \rfloor n+k, j} \quad j = 0, \cdots, L-1$$

- Probabilità di non recupero di parola, con ipotesi di errori indipendenti e distribuzione binomiale della probabilità di perdita dei pacchetti

$$P_s = 1 - (1-p)^{n-1}[1 + p(n-1)]$$

- Aumento di bit rate

$$\phi = \frac{n+1}{n}$$

N-way XOR

- CODIFICATORE
- Costruzione di parola di parità
- L=5
- n=3

01111

11000

10101

---------- XOR

00010 parola di parità

- CANALE
- Persa 01111

- DECODIFICATORE
- Ricostruzione parola persa

00010 parità

11000

10101

---------- XOR

01111 ricostruzione parola persa

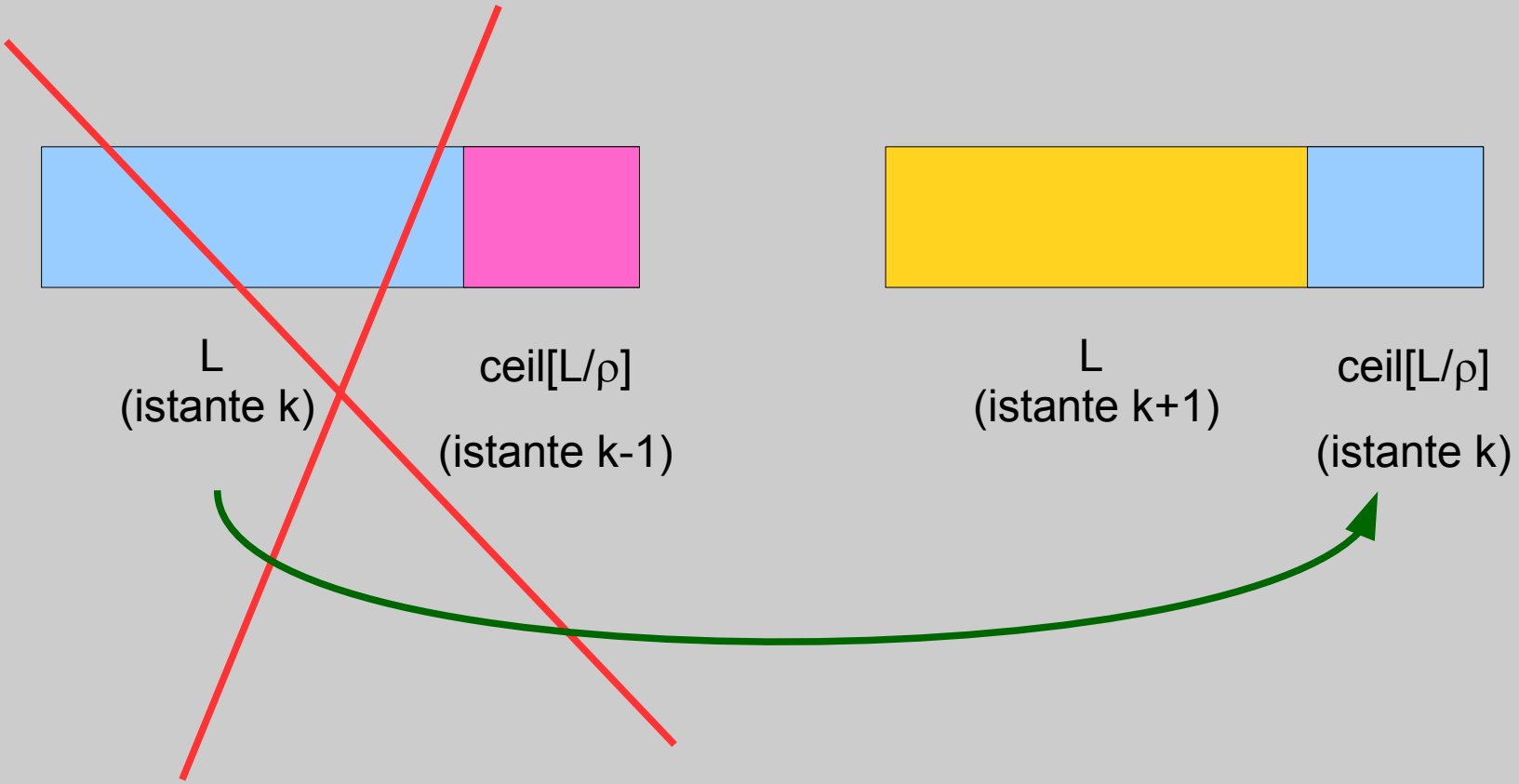
Compressione a differenti bitrate

- Sorgenti multimediali → output a diverse bitrate (diversa qualità)
- Codifica della stessa sorgente mediante differenti bitrate con associate differenti qualità
- Esempio con 2 bitrate di uscita
  - ρ=rapporto tra bitrate maggiore e bitrate minore ($\rho > 1$)
  - Pacchetti a bit rate maggiore composti da L bit
- Comporre pacchetti di L+ceil[L/ρ]
  - L bit del pacchetto a qualità maggiore all'istante k
  - ceil[L/ρ] bit del pacchetto a qualità inferiore all'istante k-1
- Decompressore normalmente decodifica i pacchetti dello stream a qualità maggiore
- In caso di perdita di un pacchetto all'istante k, decodificherà la parte a qualità inferiore contenuta nel pacchetto successivo (che appunto si riferisce a k-1, quindi a quello perso, che viene così recuperato, anche se a qualità inferiore)
- Ridondanza

$$\phi = \frac{L + \lceil \frac{L}{\rho} \rceil}{L} \approx \frac{1+\rho}{\rho}$$

- Estensioni a più di due sottostream

Compressione a differenti bitrate



Compressione a differenti bitrate

- Esempio audio
  - G.711 → 64kbps → codificatore alta qualità
  - G.729 → 8kbps → codificatore bassa qualità
  - ρ=8, ϕ=1.125
- Esempio video
  - H.261 → 1152kbps → alta qualità
  - H.261 → 128kbps → bassa qualità
  - ρ=9, ϕ=1.11

Interleaving

- La perdita di un pacchetto può causare la perdita di un'ampia parte del messaggio originale codificato dalla sorgente, tipicamente con informazioni in qualche modo correlate.
- Per minimizzare gli effetti di degradazione alla riproduzione si può procedere effettuando un **mescolamento** dello stream in modo che ***informazioni correlate vengano distanziate e la perdita non vada a colpire una singola parte della riproduzione ma si distribuisca con effetti minore in parti diverse.***
- Applicazione di una funzione di permutazione
- Aumento del ritardo per attendere che tutti i pacchetti (permutati) arrivino al decodificatore

Protezione differenziata

- Le informazioni presenti in un pacchetto non hanno tutte stessa importanza → maggior protezione alle parti più importanti (header, frame I ...)
- Esempio con due sole classi di importanza (classe1 più importante, classe2 meno importante)
- Ogni pacchetto di L bit contiene L1 bit più importanti e L2 bit meno importanti (L=L1+L2)
- Parola di parità L1 ogni n1 pacchetti e parola di parità L2 ogni n2 pacchetti (A=n2/n1, A>1, m=minimo comune multiplo tra n1 e n2)

$$P_{si} = 1-(1-p)^{n_i-1}[1+p(n_i-1)]$$

$$\phi = \frac{\sum_{i=1}^{C} L_i + \sum_{i=1}^{C} \frac{L_i}{n_i}}{\sum_{i=1}^{C} L_i}$$

$$P_{s1} = 1-(1-p)^{n_1-1}[1+p(n_1-1)]$$
$$P_{s2} = 1-(1-p)^{An_1-1}[1+p(An_1-1)]$$

$$\phi = \frac{m(L_1+L_2)+\frac{m}{n_1}L_1+\frac{m}{An_1}L_2}{m(L_1+L_2)} = \frac{An_1(L_1+L_2)+AL_1+L_2}{An_1(L_1+L_2)}$$

Protezione differenziata

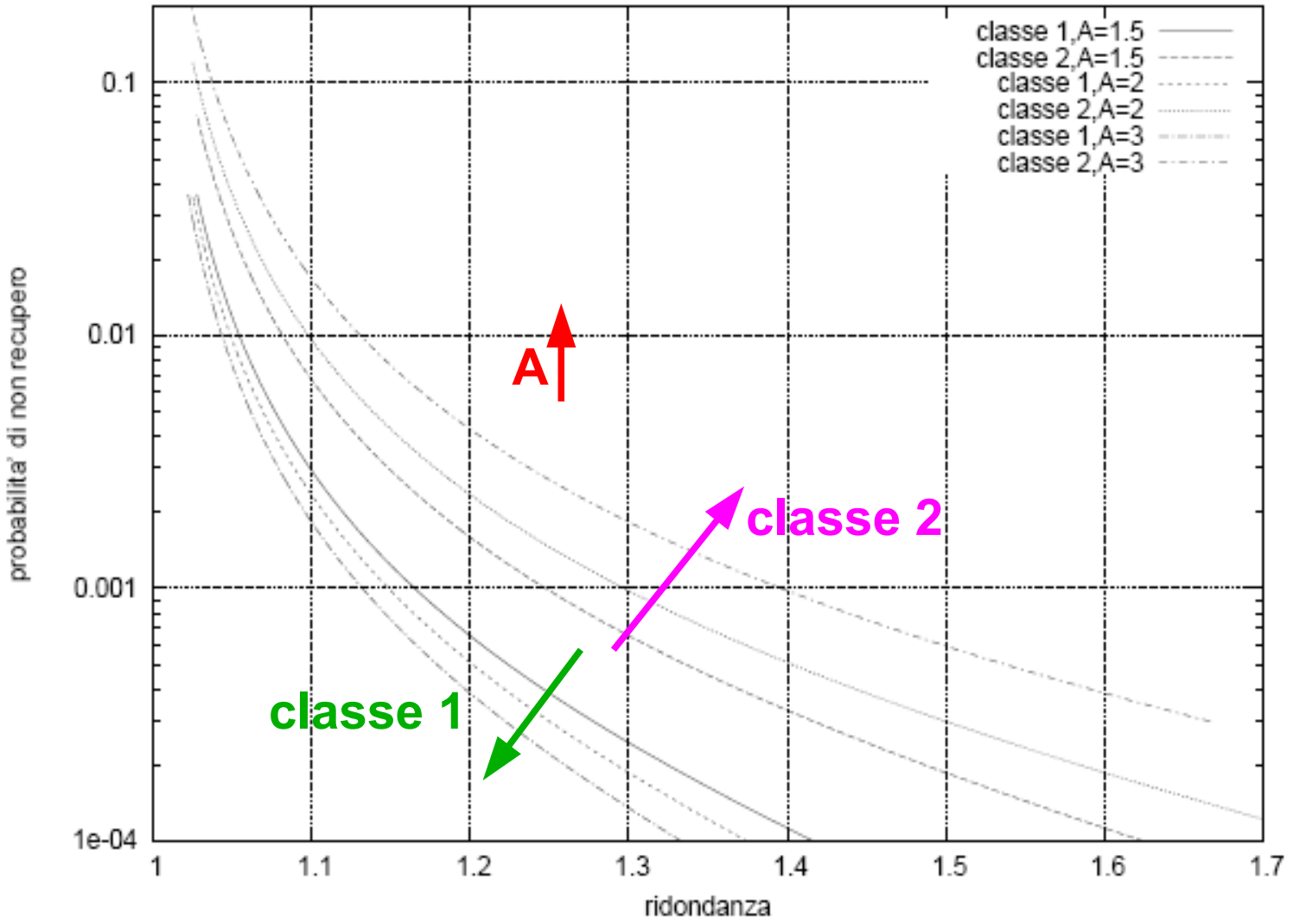


Figura 5.5: Compromesso probabilità di non recupero e ridondanza in un sistema a protezione differenziata.

Congestione

- **Condizione di congestione per una rete** → quando la richiesta di risorse supera quelle disponibili
- Linea out di un router congestionata → tasso di arrivo dei pacchetti superiore al tasso di smaltimento
- **Controllo di congestione**
  - End-to-end → limitare # pacchetti immessi sulla rete da ogni singola applicazione
  - Hop-by-hop → massimizzare throughput della rete su ogni singolo router
  - Tecniche di gestione delle code e di eliminazione mirata di pacchetti
    - **FIFO**
      - **Politiche di classificazione**
    - **FQ**
    - **WFQ**
    - **Politiche di drop**
    - **RED**

Code FIFO

- FIFO (First In First Out)
  - pacchetti processati nell'ordine esatto con cui sono entrati nel sistema
  - Classificatore, misuratore del riempimento attuale della coda e politiche di ammissione
  - Ammissione → sulla base della classe del pacchetto e del riempimento della coda
  - Politiche di classificazione
    - Combinazioni caratteristiche layer 3 e 4 (IP sorgente, IP destinatario, porta sorgente, porta destinatario, Type Of Service del TCP)

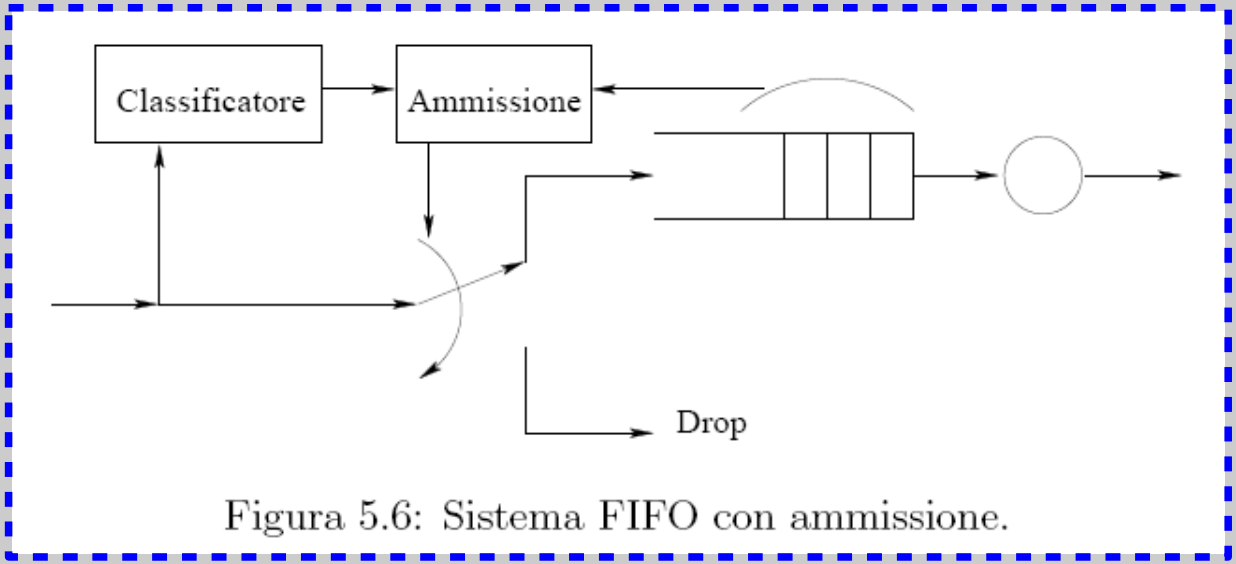


Figura 5.6: Sistema FIFO con ammissione.

Fair Queuing (FQ)

- Meccanismo di classificazione → flussi tra sorgente e destinazione
- Gestione separata per ogni flusso con associazione di ogni flusso ad una differente coda, indipendente da tutte le altre.
- Ad ogni flusso viene erogato uno stesso ammontare di servizio.
- **Si** = ammontare di servizio che il flusso i ha ricevuto sino all'istante attuale
- L'algoritmo procede assegnando servizio al flusso con il minor valore di Si che ha pacchetti pendenti
  - viene servito il flusso **k** tale che **Sk < Si** con **qi > 0**, (gi pacchetti nella coda i-esima)
- Dopo il servizio di un pacchetto di lunghezza L sul flusso i, l'ammontare di servizio viene incrementato secondo la regola: **Si = Si + L**
  - Servitore mai inattivo
- Per evitare che una coda mantenga il servizio troppo a lungo si calcola

  Smin = min{Si|qi > 0} e periodicamente si pone il valore del servizio delle code vuote pari a tale valore.

Weighted Fair Queuing (WFQ)

- Una versione avanzata del FQ.
- Consente di pesare in modo differente ogni singolo flusso.
- Wi = coefficiente assegnato al flusso i
- Meccanismo di selezione del flusso k come WkSk < WiSi con qi > 0

Politiche di drop

- Eliminazione dei pacchetti dalla coda del sistema a coda
  - Pacchetti in coda sarebbero ultimi ad essere serviti e arriverebbero a destinazione con troppo ritardo e sarebbero quindi inutilizzabili
- Eliminazione dei pacchetti dalla testa del sistema a coda
  - In condizione di congestione sono i pacchetti più vecchi presenti nel sistema e sarebbero inutilizzabili
- Eliminazione da posizione casuale
- Svuotamento totale della coda

Random Early Detection (RED)

- Meccanismo che tenta di prevenire la congestione
- Mantiene traccia della dimensione pesata della coda (L_W)
- Due soglie (minima T_m e massima T_M)
- Tre zone di lavoro:
  - $L_W < T_m \rightarrow$ pacchetti accodati
  - $T_m < L_W < T_M \rightarrow$ pacchetti droppati con probabilità P
  - $L_W > T_M \rightarrow$ pacchetti droppati

$$\beta = \gamma \frac{L_W - T_m}{T_M - T_m}$$

$$P = \frac{\beta}{1 - C\beta}$$

- Quanto più la coda si riempie, tanto più è probabile che un pacchetto venga scartato.
- La probabilità di drop aumenta anche con aumentare di C ossia del pacchetti arrivati dall'ultimo evento di drop. Il contatore C viene resettato al drop di un pacchetto o quando $L_W > T_M$

Quality of Service (QoS)

- Definizione → **garantire un certo livello di servizio ad un sistema o ad un applicativo**
- Misura → set parametri (ritardo, jitter, banda, perdita pacchetti)
- Internet (Best effort)
  - Supportare QoS occorre differenziare i servizi offerti e assicurare diversi livelli di prestazione
    - **Classificazione** pacchetti
    - **Isolamento** (assegnare caratteristiche di QoS ad una classe indipendentemente da ciò che accade alla altre classi)
    - Utilizzo di **canali a banda larga** (capacità di controllare le classi)
    - **Politiche di ammissione** (negoziazione di risorse)
- **Code con priorità**
- **Leaky e token bucket**
- **IntServ e DiffServ**

Code con priorità

- Diversamente dal FQ che serviva tutte le code in modo equo, questo sistema riserva maggiori risorse alle code a maggiore priorità.
- Processa tutti i pacchetti presenti sulla coda a maggiore priorità e passa a processare i pacchetti in una coda a priorità inferiore solo se tutte le code a priorità superiore risultano vuote
- **Esempio**
- due code (priorità 1 alta, priorità 2 bassa)
- M/G/1
- Tasso di arrivo ad ogni coda λi
- Fattore di utilizzazione di ogni coda ρi
- Condizione di stabilità $\rho = \sum_{i=1}^{K} \rho_i < 1$
- E[v] = tempo medio necessario a terminare il servizio attualmente in corso dal servitore

Code con priorità

$$E[T_{q_1}] = E[v] + E[x]\lambda_1 E[T_{q_1}]$$

$$E[T_{q_2}] = E[v] + E[x]\lambda_1 E[T_{q_1}] + E[x]\lambda_2 E[T_{q_2}] + E[x]\lambda_1 E[T_{q_2}]$$

$$E[T_{q_i}] = E[v] + \sum_{j=1}^{i} E[x]\lambda_j E[T_{q_j}] + \sum_{j=1}^{i-1} E[x]\lambda_j E[T_{q_i}]$$

$$E[T_{q_i}] = \frac{E[v] + \sum_{j=1}^{i-1} \rho_j E[T_{q_j}]}{1 - \sum_{j=1}^{i} \rho_j}$$

Code con priorità

$$E[T_{q_i}] = \frac{E[v]}{\left(1 - \sum_{j=1}^{i} \rho_j\right)\left(1 - \sum_{j=1}^{i-1} \rho_j\right)}$$

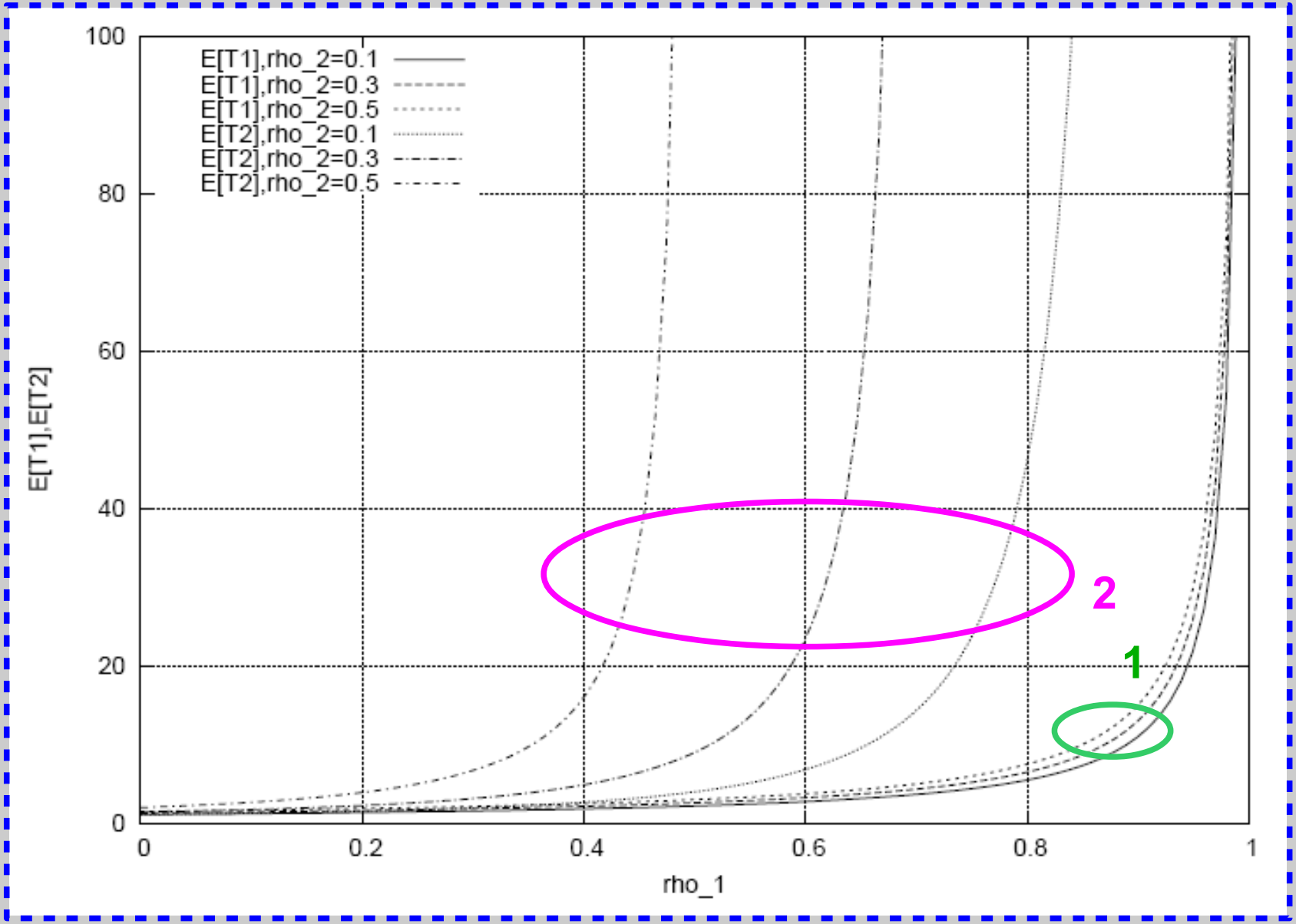
$$E[v] = \frac{1}{2}\sum_{j=1}^{K} \lambda_j E[x^2]$$

$$E[T_1] = \frac{1+\rho_2}{1-\rho_1}E[x]$$

$$E[T_2] = \frac{1-\rho_1(1-\rho_1-\rho_2)}{(1-\rho_1)(1-\rho_1-\rho_2)}E[x]$$

- Si ricade nelle formule dell M/M/1 in caso di $\rho 1=0$ o di $\rho 2=0$
- E[T1] risente poco del traffico presente sulla coda 2
- E[T2] ha una considerevole dipendenza da $\rho 1$ e l'impatto è tanto maggiore quanto più è grande $\rho 1$

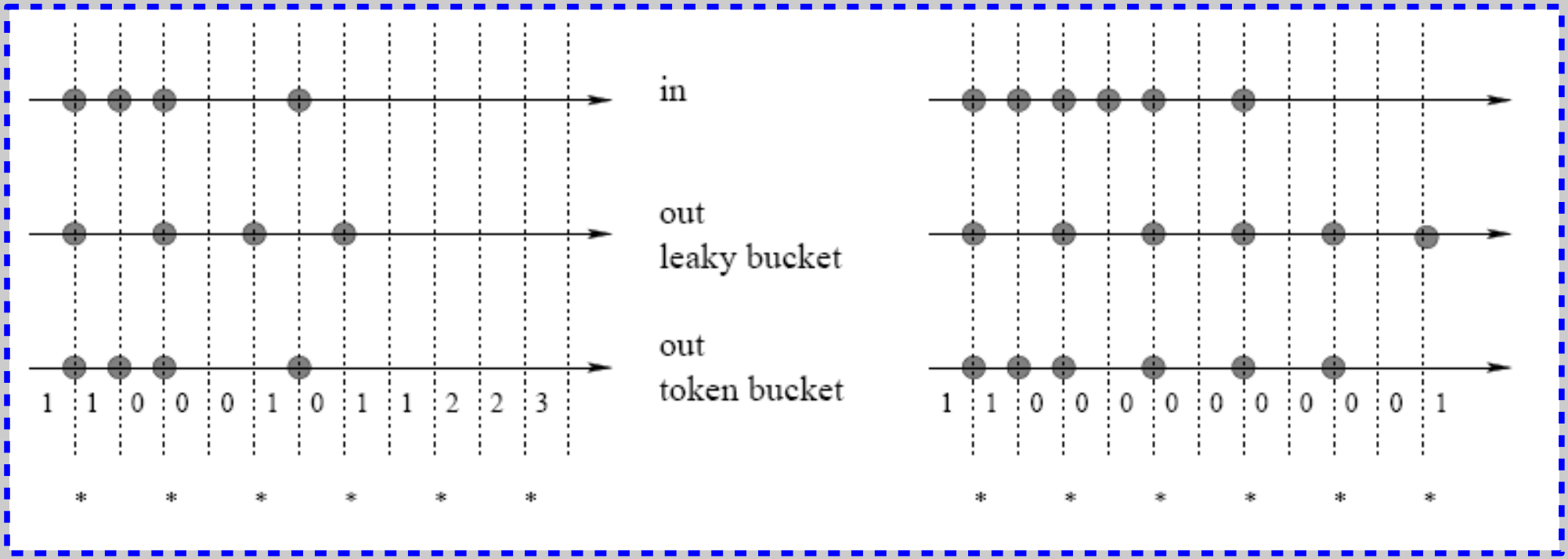
Code con priorità



Traffic Shaping

- Strategie usate per variare la statistica di traffico di una sorgente
  - controllare il traffico in rete in modo da garantire e ottimizzare le performance
  - controllare la velocità con cui i dati vengono immessi in rete
  - generare pacchetti temporalmente equispaziati, indipendentemente dalle caratteristiche dell'ingresso
  - ingresso (traffico con caratteristiche variabili) → sistema di traffic shaping → uscita con traffico smooth, piatto
- Obiettivo → rendere traffico maggiormente predicibile e quindi gestibile
- **Leaky Bucket**
- **Token Bucket**

Leaky Bucket & Token Bucket



Integrated Services (IntServ)

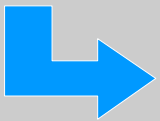
- Architettura di rete per fornire Qos alle sessioni delle applicazioni in una rete IP, tramite un meccanismo di **riservazione delle risorse**
- Ogni router nel percorso tra S e D deve mantenere informazioni riguardo le risorse allocate su diversi circuiti virtuali legati alle sessioni tra applicativi ed effettua politiche di ammissione per instaurazione di circuiti virtuali
- Supporta due classi di servizio: **QoS garantito** (ritardo e jitter) e **controllo di carico** (approx come se la rete fosse scarica)
- Protocollo di segnalazione **RSVP**, usa due specifiche:
  - R-spec → per definire requisiti di QoS (classi)
  - T-spec → per definire caratteristiche di traffico
  - Messaggi multicast per segnalazione
- Problemi: complessità, segnalazione complessa RSVP, scalabilità limitata, poche classi, diffusione (ogni router deve implementarlo)

Differentiated Services (DiffServ)

- Le risorse vengono allocate per **classi** invece che per flussi individuali tra applicativi
- Complessità viene portata ai soli router di bordo (marchiano i pacchetti su classi differenti)
- I router della rete processano tali pacchetti su **code a differenti priorità**
- Tre classi: **best-effort, premium service, assured service**

Real-time Transport Protocol (RTP)

- ***Come inviare uno stream audio/video su una rete Internet?***
- **Requisiti** per tx sorgente multimediale
  - sincronizzazione flussi
  - controllo compressori
  - gestione QoS
  - controllo applicazioni
  - controllo affidabilità trasmissione.
- **Diffusione** dello stream → unicast, multicast
- **Architettura di rete**
  - media gateway (riceve, manipola e ritrasmette il flusso)
  - attivo (transcodifica e mixing) o passivo (filtraggio e registrazione)



RTP

RTP

- **Componenti**
  - RTP → gestione unicast e multicast dello streaming
  - RTCP → Real-time Transport Control Protocol per feedback qualità e per gestione conferenze
  - RTP media gateway → mixer, monitor, transcodifica
  - Formato e semantica messaggi RTP e RTCP
  - Rappresentazione payload RTP
- RTP e RTCP si appoggiano a **UDP**, lavorano su due porte diverse
- **SSRC** (Synchronization Source) identificatore univoco di sessione di una sorgente
- **CSRC** (Contributing Source) identificatore di un media gateway attivo

  (gateway passivo emette pacchetti con lo stesso ssrc)

RTP - pacchetto

- V → versione
- P → padding
- CC → # di CSRC presenti nell'header
- Sequence number → incrementato per ogni pacchetto emesso dalla sorgente
- Timestamp → per sincronizzazione dei flussi
- SSRC → identificativo della sorgente

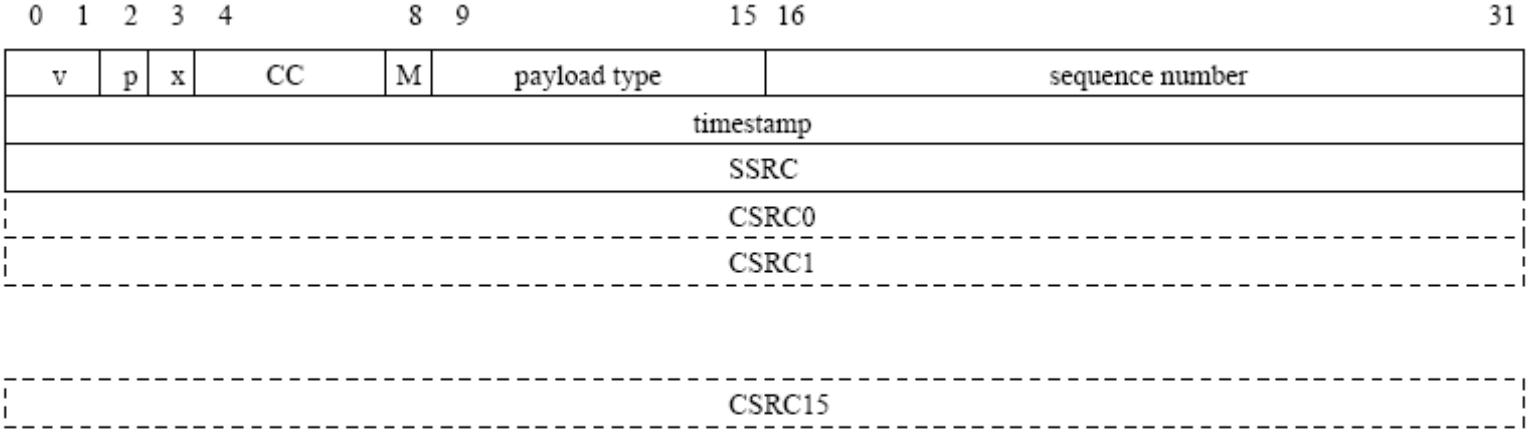


Figura 5.9: Header pacchetto RTP.

RTCP

- I sistemi coinvolti in una sessione si scambiano periodicamente messaggi RTCP
  - **Sincronizzazione** (orologi dei vari sistemi)
  - **QoS** (#pacchetti persi, ritardo medio, livello jitter)
  - scambio **informazioni relative ai partecipanti** coinvolti in conferenze

Overhead

- IP (20B) + UDP (8B) + RTP(12B) = 40 Byte
- Livello 2 = 30Byte circa
- **Totale 70Byte**
- *Esempio G.729*
  - *Specifiche nominale bit rate 8kbps, con frame di 10ms*
  - *Ogni frame contiene 10byte di dati e 70byte di overhead*
  - *Bitrate effettiva 64kbps*

Asynchronous Transfer Mode (ATM)

- Pacchetti piccoli (**CELLE**) di dimensioni fisse (53Bytes)
- Commutazione a pacchetto con **circuito virtuale**
- Percorso tra S e D definito con procedura di segnalazione
- Pacchetti identificati dal numero del circuito virtuale (#bit necessari a identificare un circuito sono minori del #bit necessari a identificare un destinatario)
- Circuito virtuale → ordine pacchetti viene rispettato
- Circuito virtuale → supporto nativo a QoS e riservazione risorse
- Pacchetti a dimensione fissata semplifica la gestione code (a parità di complessità hw dei sistemi, ATM può supportare bitrate più alte)
- Formato cella
  - **53bytes = 48 paylod + 5 header**
  - Header → generic flow control, **virtual path identifier (VPI), virtual channel identifier (VCI)**, cell loss priority

ATM

- Circuito virtuale identificato da due campi dell'header → VCI e VPI
- **Commutazione realizzata su VPI**
  - VCI rimangono inalterati, ogni chiamata è distinta dal proprio VPI
  - Usata quando le chiamate sono originate nello stesso punto di ingresso alla rete e sono destinate allo stesso punto di uscita
- **Commutazione realizzata su VCI**
  - Vengono cambiati sia VPI che VCI ad ogni switch (coppia di valori costituisce il PCI, Protocol Connection Identifier)
  - Si utilizza quando i flussi VCI sono diretti a destinazioni diverse

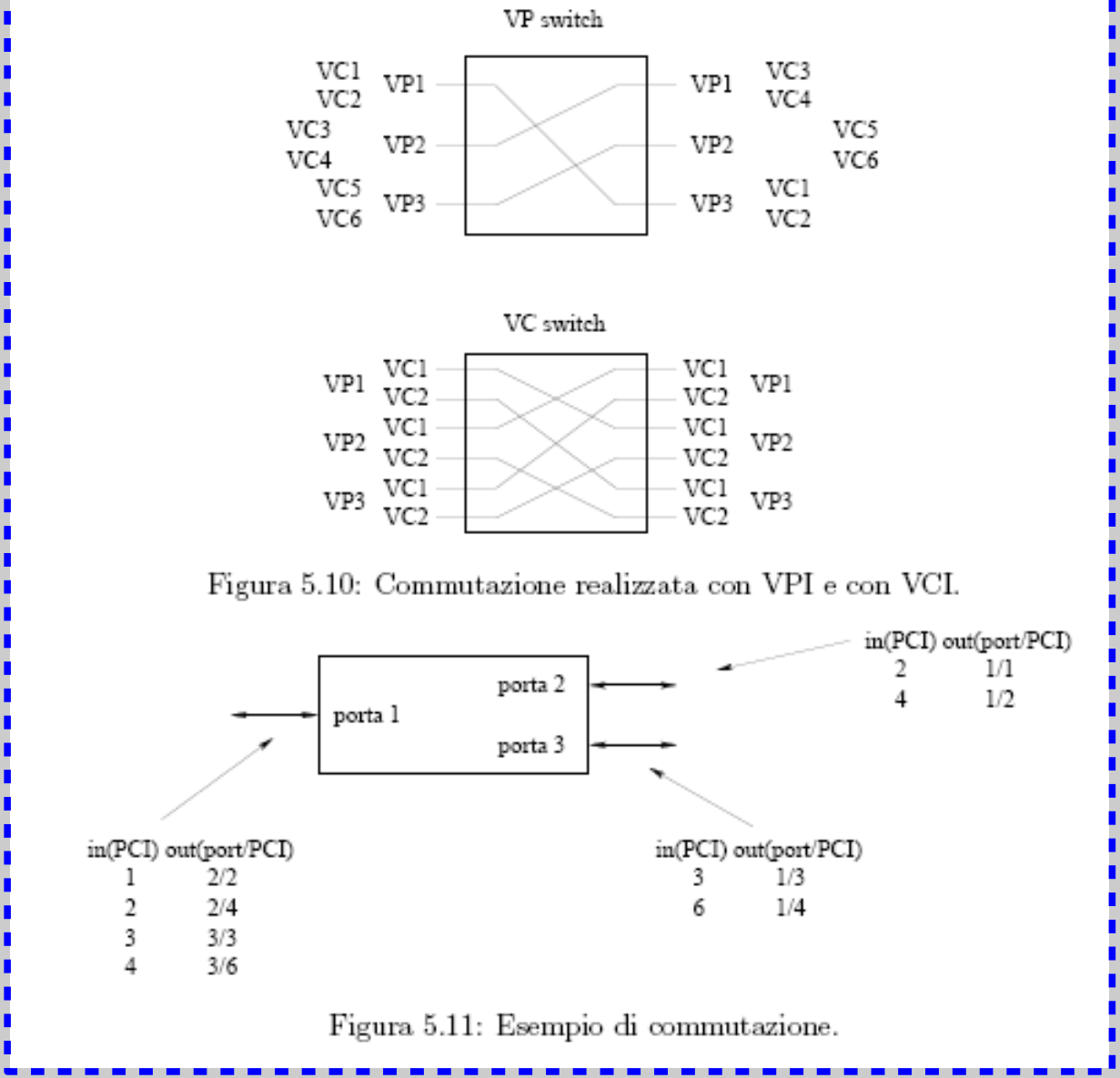


Figura 5.10: Commutazione realizzata con VPI e con VCI.

Figura 5.11: Esempio di commutazione.

ATM

- **Architettura** basata su tre funzioni
  - *Piano di controllo*
    - Protocolli di segnalazione per creazione ed eliminazione di circuiti virtuali
  - *Piano utente*
    - Protocolli che dipendono dall'applicazione e che comunicano end-to-end
  - *Piano di gestione*
    - Funzioni di organizzazione della stazione (es: errori nel funzionamento)
- Utilizzo delle celle è trasparente alle applicazioni grazie all'uso degli **ATM Adaptation Layer (AAL)**
  - Stack → strato fisico, ATM, AAL
  - **Cinque tipi di servizio (AAL1, AAL2, AAL3, AAL4, AAL5)** in funzione di ritardo, bitrate costante o variabile, tipo di connessione (con o senza connessione)

ATM

- Fornisce **classi di servizio** relative alla **bitrate**
  - **CBR:** Constant Bit Rate, audio e video non compressi
  - **VBR/RT:** Variable Bit Rate/Real Time, trasmissione real time di video compresso
  - **VBR/NRT:** Variable Bit Rate/Non Real Time, trasferimento video tra server
  - **ABR:** Available Bit Rate, specifica bit rate minima, media e massima e la media viene garantita tramite segnalazioni con le applicazioni
  - **UBR:** Unspecified Bit Rate, non viene assicurata alcuna garanzia di banda.

ATM

- Associata ad ogni classe di servizio è possibile specificare una QoS (banda, ritardo, jitter) che costituisce la base del contratto tra gestore e utente.
- Parametri caratteristici per una sorgente/destinazione (utente)
  - **Peak Cell Rate**: massima velocità con cui la sorgente eroga le celle
  - **Susteined Cell Rate**: velocità media con cui la sorgente invia le celle
  - **Minimum Cell Rate**: velocità minima delle celle che è accettabile per una sorgente
  - **Cell Delay Variation Tolerance**: massimo jitter tollerato
- Parametri relativi alla rete
  - **Cell Loss Ratio**: percentuale di celle non consegnate
  - **Cell Transfer Delay**: ritardo medio di trasferimento delle celle attraverso al rete
  - **Cell Delay Variation**: jitter medio