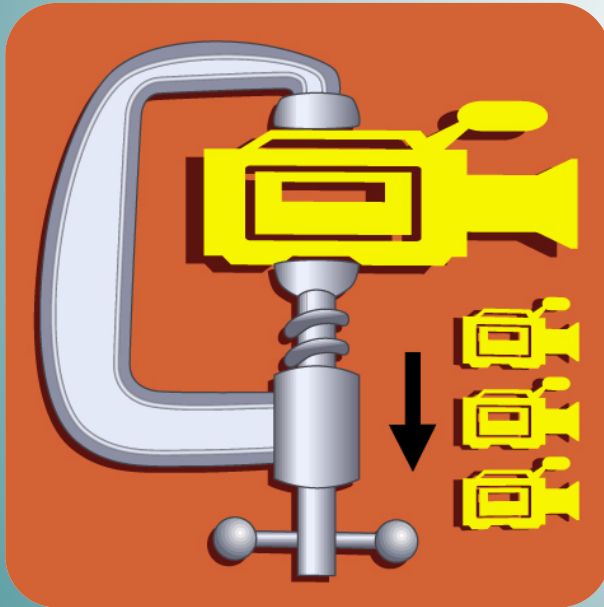


Compresión de Vídeo

Tema 2.8. Otras características



Juan A. Michell Martín
Gustavo A. Ruiz Robredo

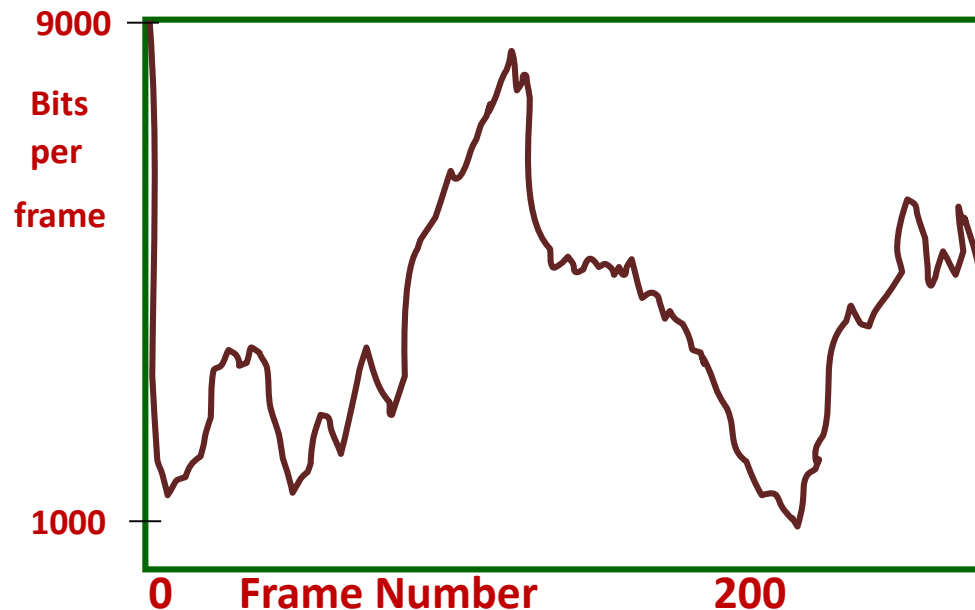
Departamento de Electrónica y Computadores

Este tema se publica bajo Licencia:

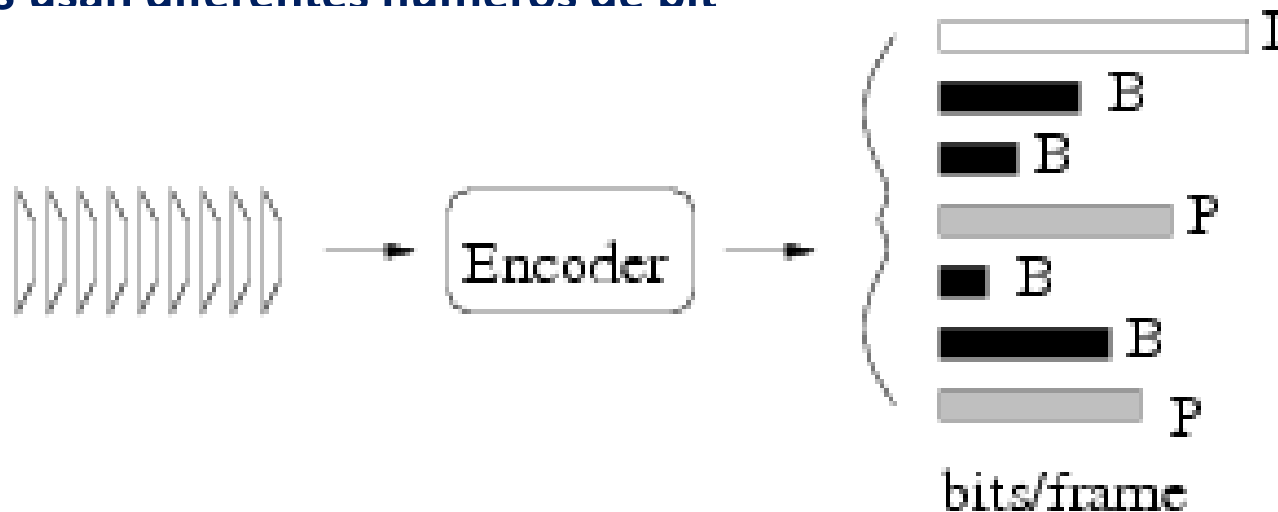
[Creative Commons BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

RATE CONTROL

- Si los parámetros usados en la codificación de un video son constantes:
 - QP
 - *Motion estimation, search area, ...*
- Entonces el tamaño de la frame codificada varía (por ejemplo de 1300 a 9000 bits resultando bit rate de 32-225 kbits/sec)



➤ **Las *frames* usan diferentes números de bit**



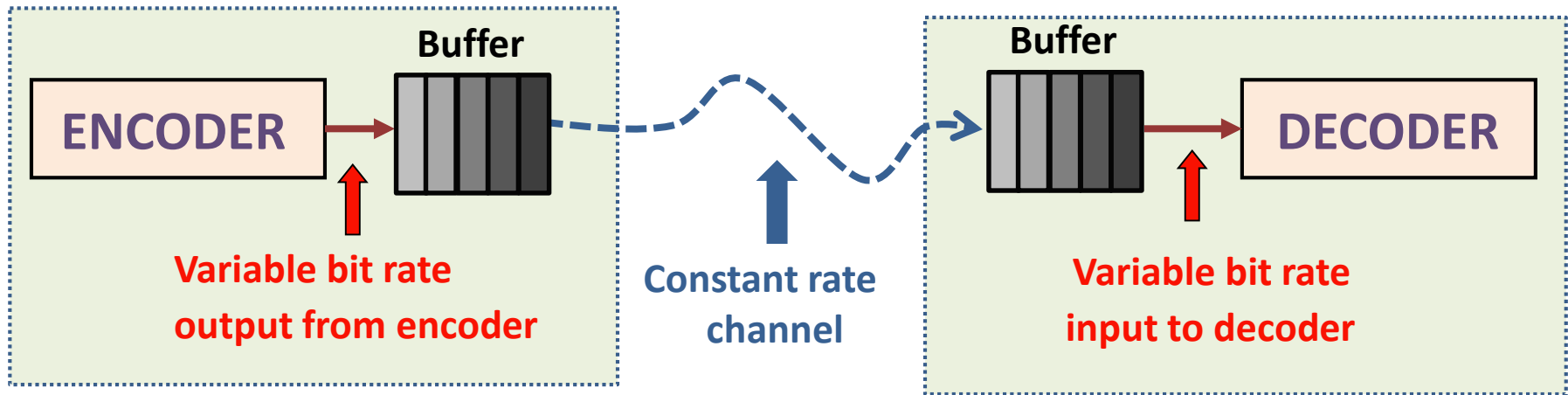
➤ **Estas variaciones son un problema:**

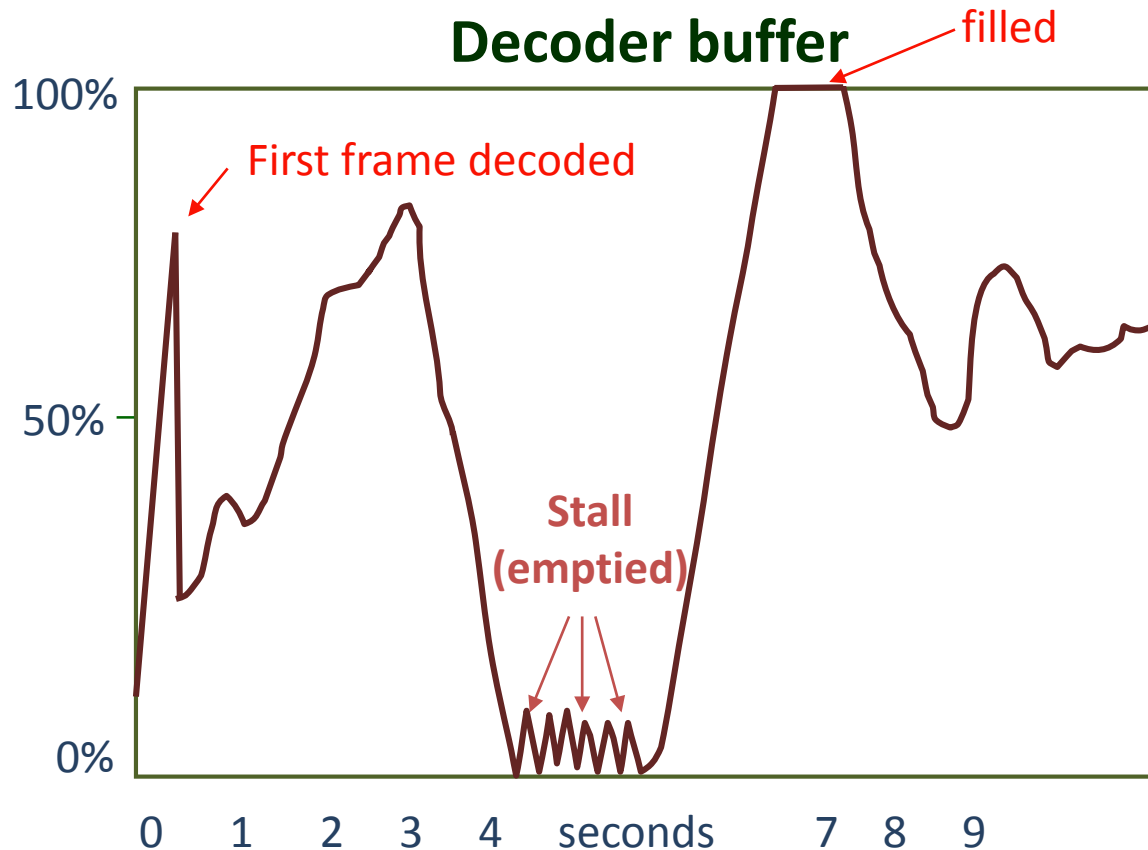
- **Canales de rate constante.** Por ejemplo sistemas de almacenamiento.
- **Canales de rate variable.** Por ejemplo, transmisión via internet, streaming, ...

➤ **Solución: Utilizar buffers de almacenamiento temporal y control de los parámetros de codificación.**

Canal de *rate* constante (*Constant rate channel*)

- El variable bit-rate producido en el *encoder* puede ser suavizado por un **buffer tipo FIFO** (First Input- First Output) a su salida.
- Otro buffer a la entrada del *decoder* gestiona su entrada de datos variable.





- 0.5 seg para decodificar la primera frame.
- Entonces se decodifica a 25 frames/sec...
- Alrededor de los 4 seg, el buffer se vacía. El decodificador se para por falta de datos.
 - Problema: La imagen se congela a la espera de más datos.
 - Solución parcial: Añadir un retraso deliberado de 1 seg (por ejemplo) para permitir que el buffer tenga mayor retención de datos.
- El buffer se llena. No admite más datos de entrada. Se pierden *frames*.

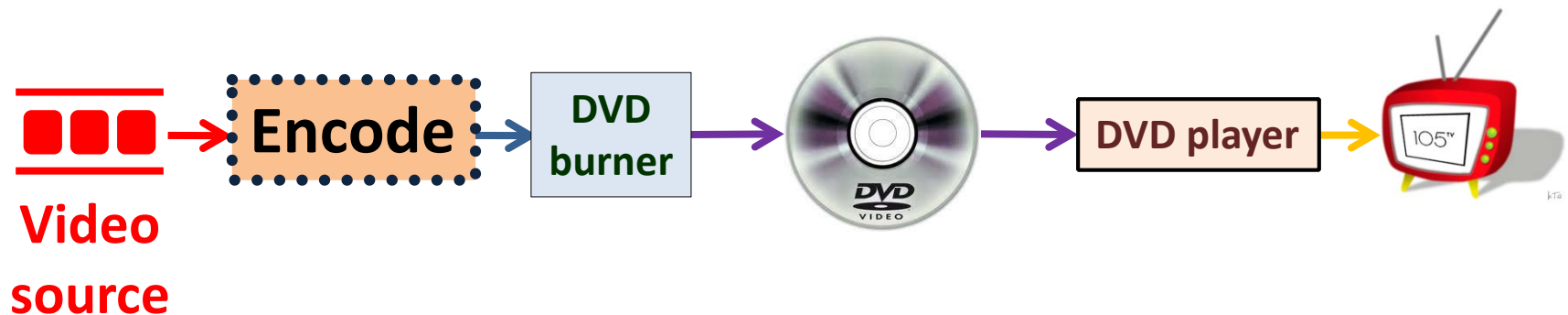
Canal de *rate* variable (*Variable rate channel*)

- Un canal de *rate* variable puede adaptarse a uno de *rate* constante usando buffers. Esto implica
 - Coste debido al buffer
 - Retraso
- No es posible enfrentarse a cualquier arbitraria variación del *bit-rate* si no es usando un buffer de gran tamaño.
- Se precisa de algún método que controle el porcentaje de ocupación del buffer.

Elección del bit-rate control

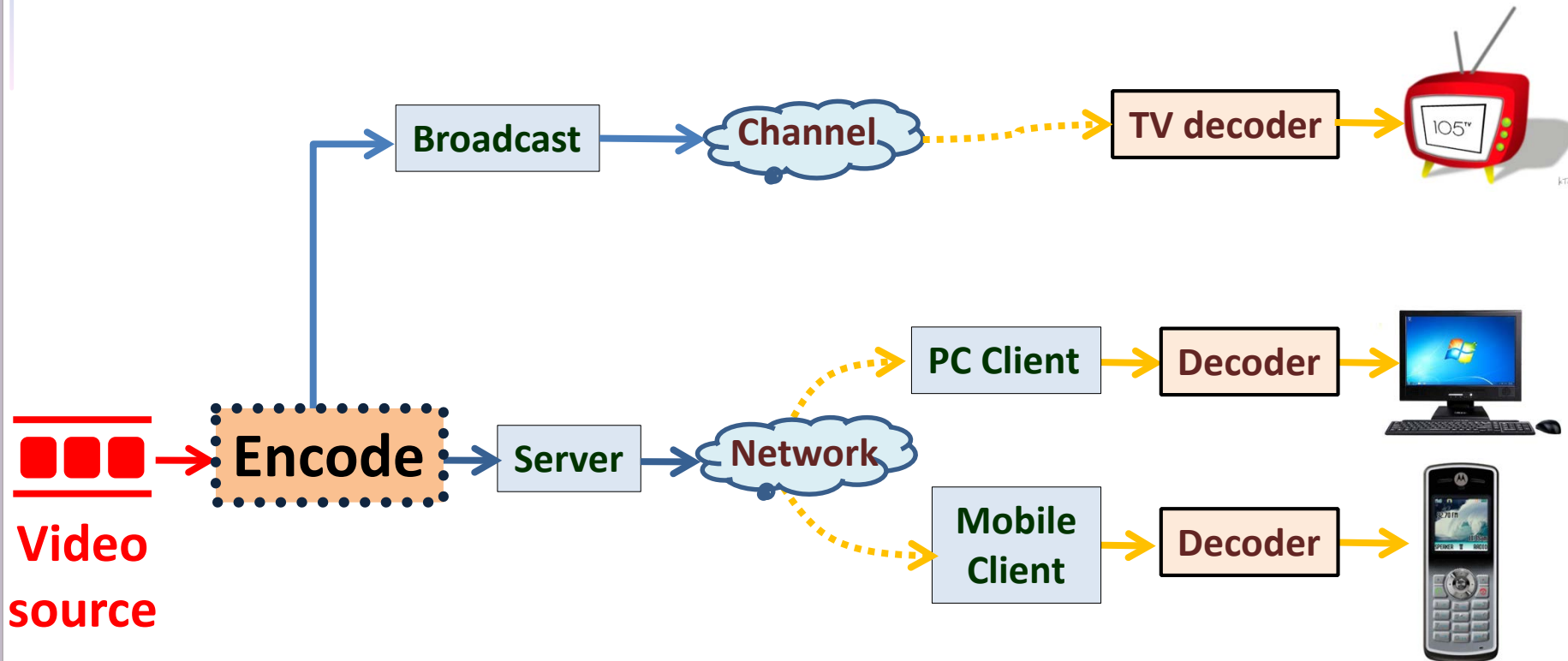
➤ *Offline-encoding* de video para DVD's:

- Tiempo de procesado no es una limitación.
- Se pueden emplear algoritmos complejos.
- Se realizan dos pasadas (*Two-pass*) en el proceso de codificación:
 - *Encoder* extrae datos estadísticas del video en la 1ª pasada.
 - *Encoder* codifica el video en la 2ª pasada.
- Se busca adaptar el tamaño de video al espacio del DVD:
 - Maximizando la calidad del video
 - Preveniendo vaciar (underflow) o llenar (overflow) el buffer durante el proceso de decodificación.



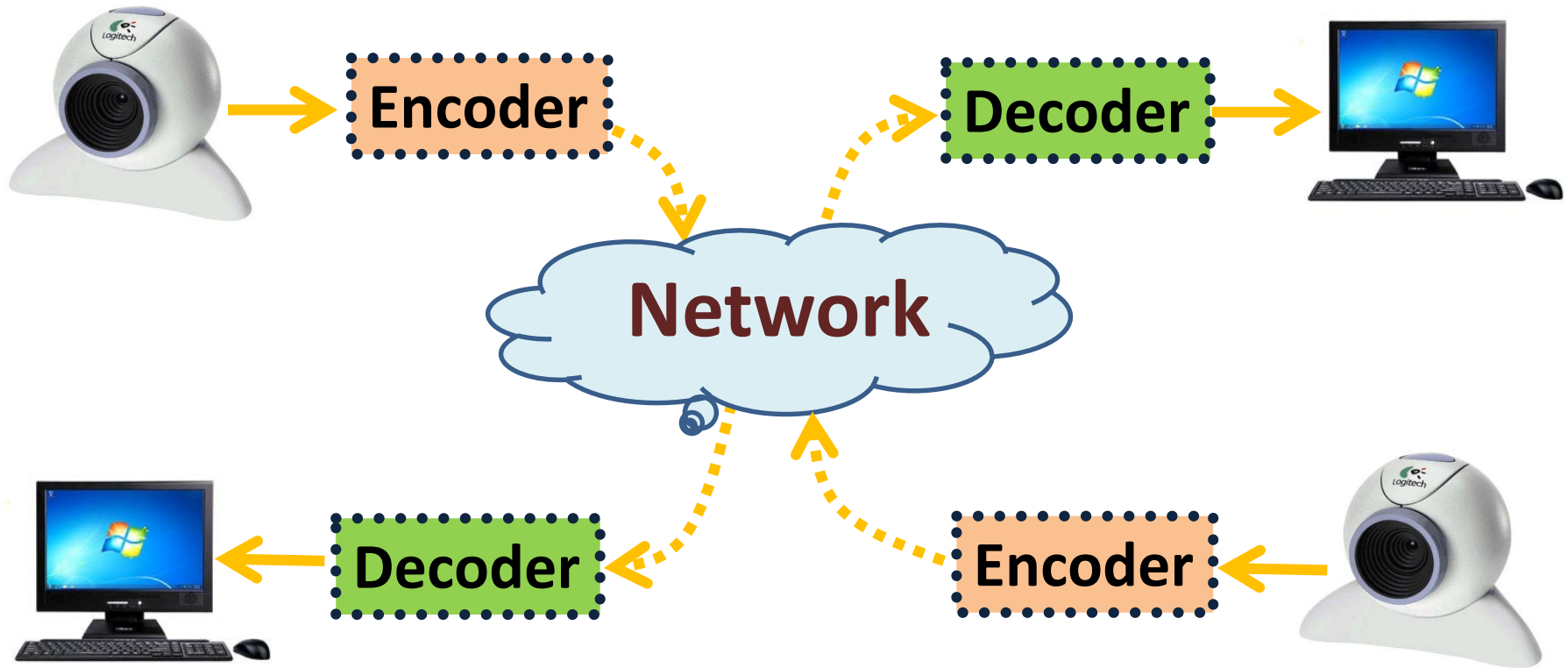
➤ Codificación de video para transmisión (*broadcast*).

- Un *encoder* → Múltiples decodificadores.
- Proceso de decodificación y *buffering* son limitados.
- *Encoder* puede usar hardware de altas prestaciones.
- El retraso de unos segundos puede ser admisible.
- El algoritmo de *rate-control* tiene media complejidad.
- En algunos casos admite dos pasadas (*two-pass*) para cada *frame*.



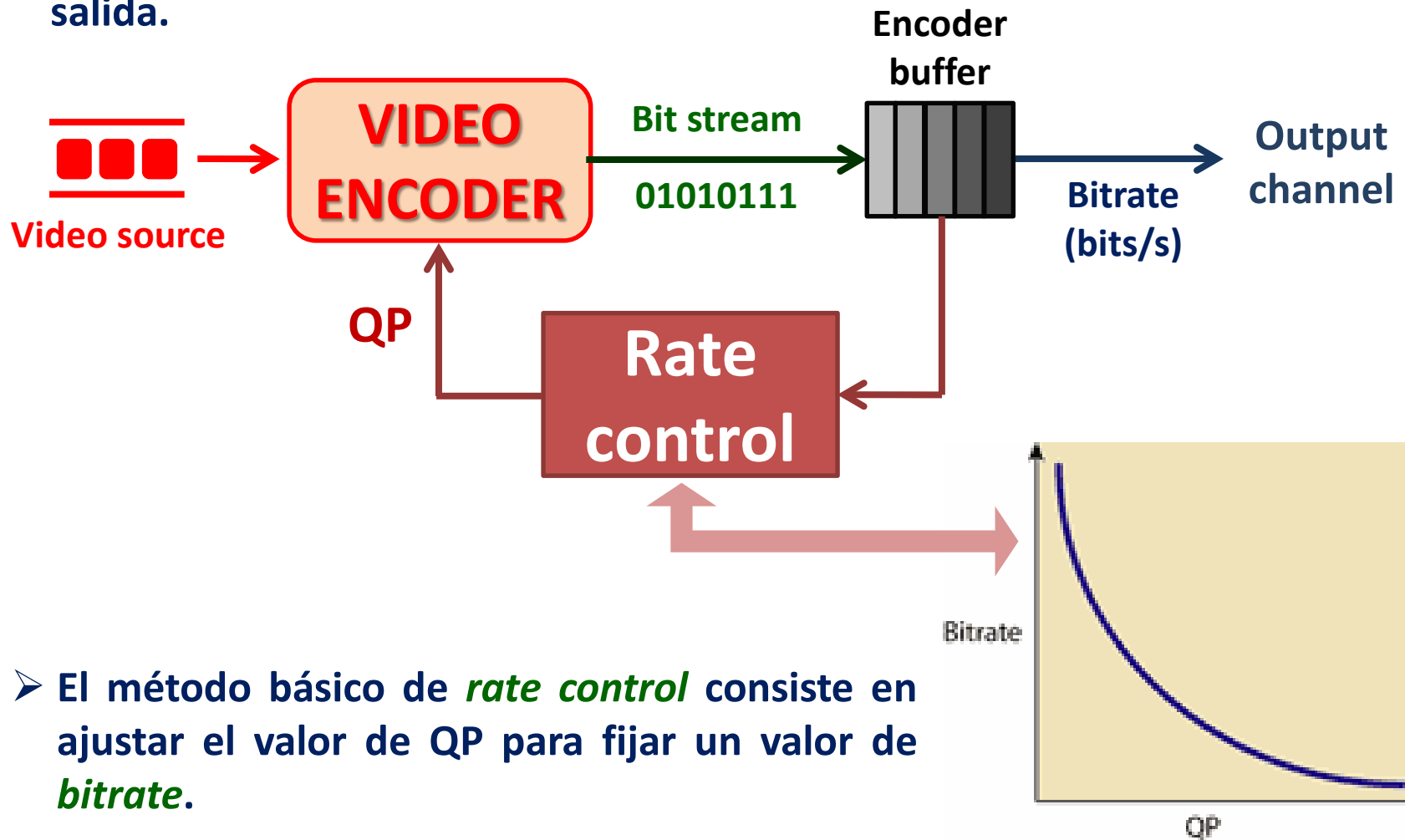
➤ Codificación para video-conferencia (*Two-way*)

- Cada terminal debe hacer simultáneamente codificación y decodificación.
- El retraso debe ser mantenido al mínimo (< 0.5 seg).
- Baja complejidad del *rate-control*.
- Tamaño del buffer se reduce al mínimo para mantener un retraso pequeño.
- El codificador debe tener un total control del output-rate.
- Consecuencias: La calidad del video puede variar significativamente, sobretodo en escenas de gran movimiento.



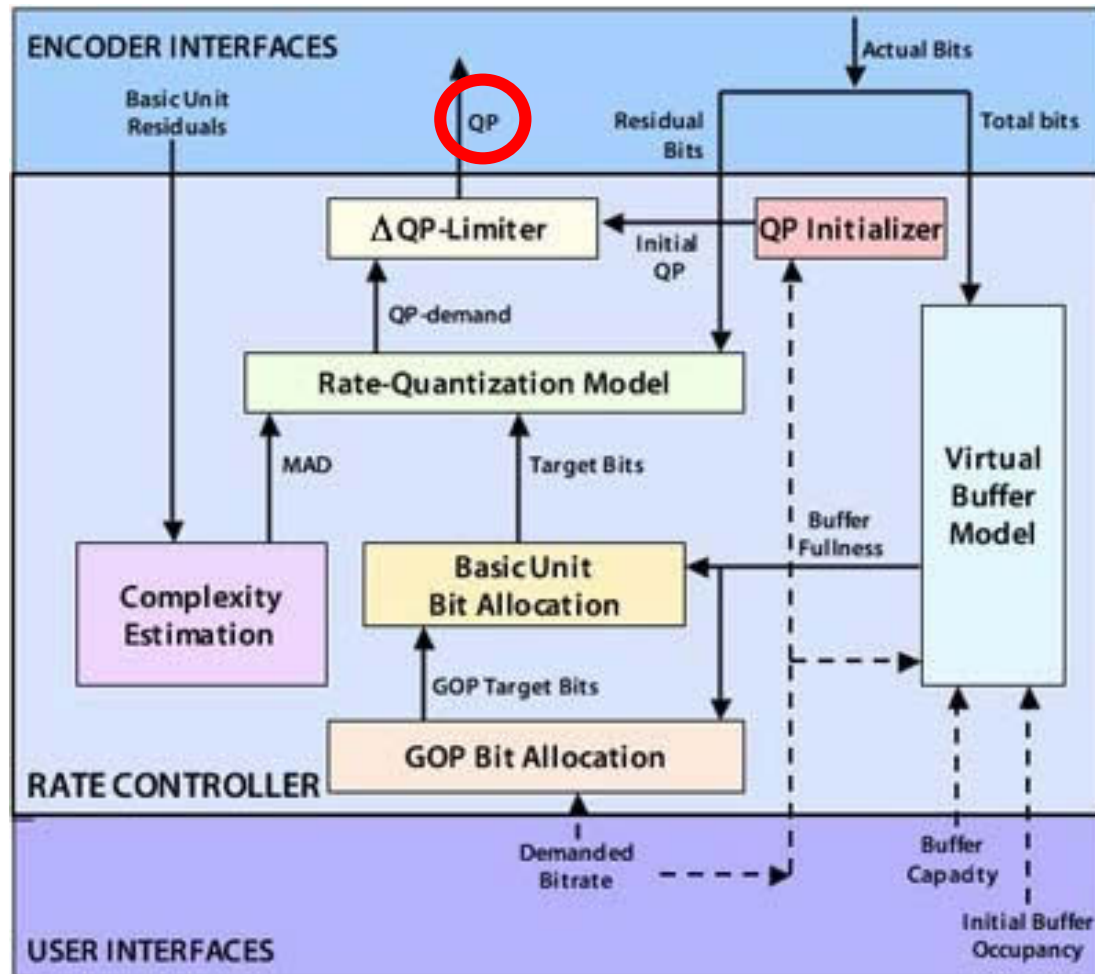
RATE CONTROL EN EL H.264

- *Rate control* en sistemas de codificación de video: limita el *bitrate* de salida.



- El método básico de *rate control* consiste en ajustar el valor de QP para fijar un valor de *bitrate*.

- El H.264 usa un modelo de *buffer* virtual: Simula el efecto de *buffer* de un decodificador real.
- EL *rate control* se aplica a diferentes unidades: *frame*, *slice* y MB.



```
cmd) EncodeH264MSWin.exe -f configuracion.cfg
```

```
FrameRate      = 10.0 # Frame Rate per second (0.1-100.0)
```

```
.....
```

```
RDOptimization = 1 # rd-optimized mode decision  
                # 0: RD-off (Low complexity mode)  
                # 1: RD-on (High complexity mode)  
                # 2: RD-on (Fast high complexity mode - not work in FREX Profiles)  
                # 3: with losses
```

OCULTAMIENTO DE ERRORES (ERROR CONCEALMENT)

- El proceso de ocultamiento de errores (*error concealment*) busca reconstruir la información perdida de un video durante el proceso de transmisión o almacenamiento.



Secuencia original



Secuencia distorsionada



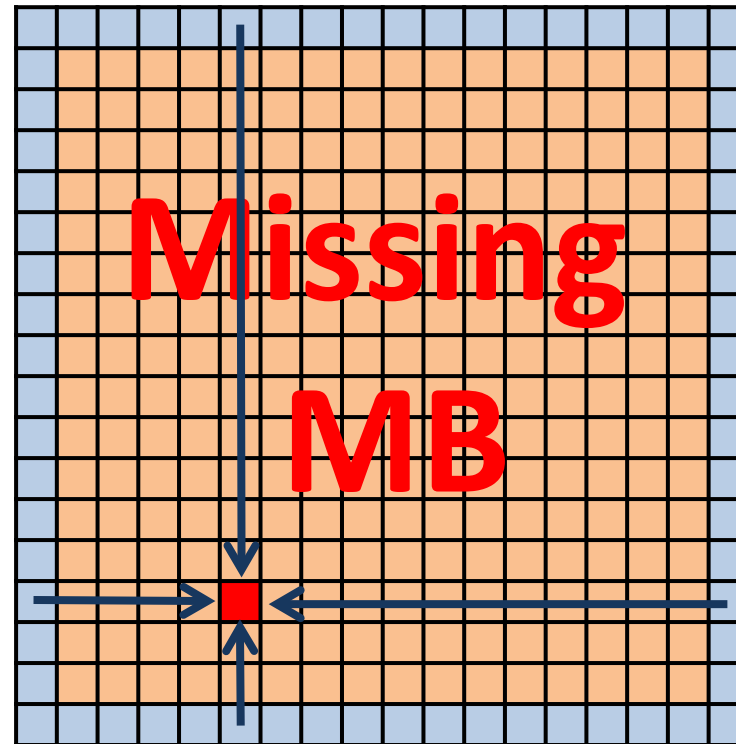
Secuencia reconstruida

- **Ocultamiento de errores espacial (*spatial concealment*):**
 - Estimación de los valores de pixel perdidos a partir de una suave extrapolación de los pixels de alrededor válidos.

- **Ocultamiento de errores temporal (*temporal Concealment*):**
 - Copia los pixels en la misma posición de la frame anterior.
 - Este procedimiento es efectivo cuando no hay movimiento, pero induce a potenciales problemas con movimiento.

- **Movimiento temporal compensado (*concealment–motion vector interpolation*):**
 - Estimar los bloques desaparecidos como bloques compensados en movimiento de la *frame* anterior.
 - Opciones posibles:
 - usar el vector de movimiento codificado,
 - usar vector de movimiento de bloques contiguos, o
 - computar un nuevo vector de movimiento.

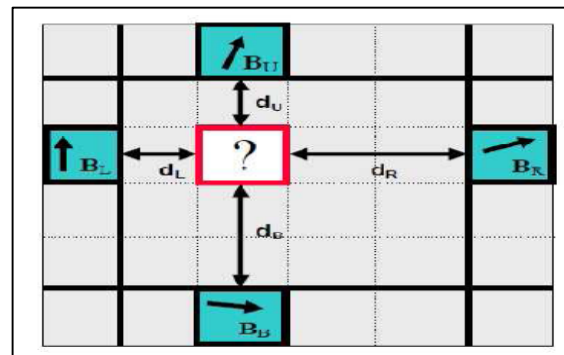
➤ Ocultamiento de errores espacial (*spatial concealment*)



Neighbor Pixels



➤ Ocultamiento de errores temporales (*temporal concealment*)



Frame t-1

Frame t

Frame t+1



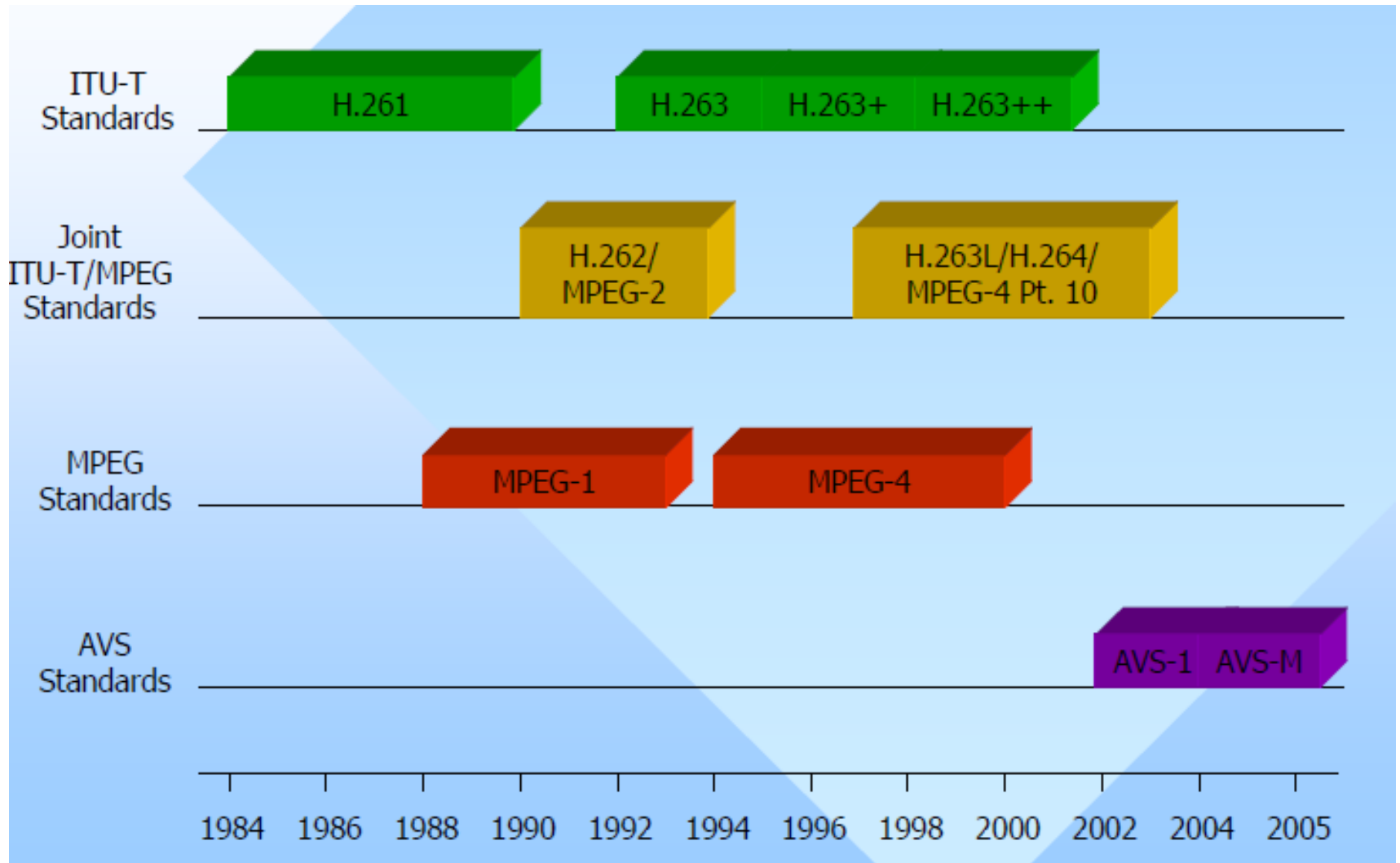
- Frame t reconstruida a partir de los vectores de movimiento de la frame t-1.
- Frame t+1 posee errores debido a que la frame t no es la frame recuperada

COMPARACIÓN DE LOS ESTANDARS

Feature/Standard	MPEG-1	MPEG-2	MPEG-4 part 2 (visual)	H.264/MPEG-4 part 10
Macroblock size	16x16	16x16 (frame mode) 16x8 (field mode)	16x16	16x16
Block Size	8x8	8x8	16x16, 16x8, 8x8	16x16, 8x16, 16x8, 8x8, 4x8, 8x4, 4x4
Transform	8x8 DCT	8x8 DCT	8x8 DCT/Wavelet	4x4, 8x8 Int DCT 4x4, 2x2 Hadamard
Quantization	Scalar quantization with step size of constant increment	Scalar quantization with step size of constant increment	Vector quantization	Scalar quantization with step size of increase at the rate of 12.5%
Entropy coding	VLC	VLC	VLC	VLC, CAVLC, CABAC
Motion Estimation & Compensation	Yes	Yes	Yes	Yes, more flexible Up to 16 MVs per MB
Playback & Random Access	Yes	Yes	Yes	Yes

Feature/Standard	MPEG-1	MPEG-2	MPEG-4 part 2 (visual)	H.264/MPEG-4 part 10
Pel accuracy	Integer, 1/2-pel	Integer, 1/2-pel	Integer, 1/2-pel, 1/4-pel	Integer, 1/2-pel, 1/4-pel
Profiles	No	5	8	3
Reference picture	one	one	one	Multiple (16)
Bidirectional prediction mode	forward/backward	forward/backward	forward/backward	forward/forward forward/backward backward/backward
Picture Types	I, P, B, D	I, P, B	I, P, B	I, P, B, SP, SI
Error robustness	Synchronization & concealment	Data partitioning, FEC for important packet transmission	Synchronization, Data partitioning, Header extension, Reversible VLCs	Data partitioning, Parameter setting, Flexible macroblock ordering, Redundant slice, Switched slice
Transmission rate	Up to 1.5Mbps	2-15Mbps	64kbps - 2Mbps	64kbps -150Mbps

AUDIO VIDEO CODING STANDARD OF (AVS) CHINA



Comparison	AVS Part-2	H.264/AVC
Number of profiles	2	4
Picture type	I, P and B	I, P, B, SI and SP
Interlace handling	PAFF and MBAFF	PAFF and MBAFF
Color sampling format	4:2:0 and 4:2:2	4:2:0, 4:2:2 and 4:4:4
Bits per sample	8	up to 12
Block size for intra prediction	8x8 for luma and chroma	4x4, 8x8 and 16x16 for luma 8x8 for chroma
Intra prediction mode	5 modes for luma 4 modes for chroma	13 modes for luma 4 modes for chroma
Number of reference frame	at most 2	depends on the level, can be much more than 2
Bi-directional prediction	at most 1 MV transmitted	at most 2 MVs transmitted
MV prediction	Geometrical median prediction	Median prediction

Comparison	AVS Part-2	H.264/AVC
Interpolation	4-tap filters for 1/2- and 1/4-pixel	6-tap filter for 1/2-pixel 2-tap filter for 1/4-pixel
MC minimum block size	8x8	4x4
Transform	8x8 Pre-scaled ICT	4x4 and 8x8 ICT
Quantization	64 QPs nearly 8-order exponential periodic	52 QPs 6-order exponential periodic
Perceptual-based quant.	No	Yes
Adaptive scan	Yes	No
Entropy coding	CA-2D-VLC and CABAC	CAVLC and CABAC
Deblocking filter	8x8-based, 3 BS levels	4x4-based, 4 BS levels
Error robustness	Weak	Strong
Supporting lossless coding	No	Yes

