

## 2 Conceitos Gerais do Padrão MPEG-2

Para diminuir os requisitos de recursos para a transmissão e o armazenamento de sinais digitais de vídeo e áudio, vários padrões de codificação, compactação e compressão foram desenvolvidos. Visando estabelecer padrões internacionais para a representação e codificação de informações audiovisuais em formato digital com compressão, a ISO (*International Organization for Standardization*) e a IEC (*International Electrotechnical Commission*) estabeleceram o grupo de trabalho MPEG (*Motion Picture Coding Experts Group*), que iniciou seus trabalhos em maio de 1988. A família de padrões produzidos foi popularmente conhecida como padrões MPEG e inclui, entre outros, os conjuntos de padrões MPEG-1 e MPEG-2.

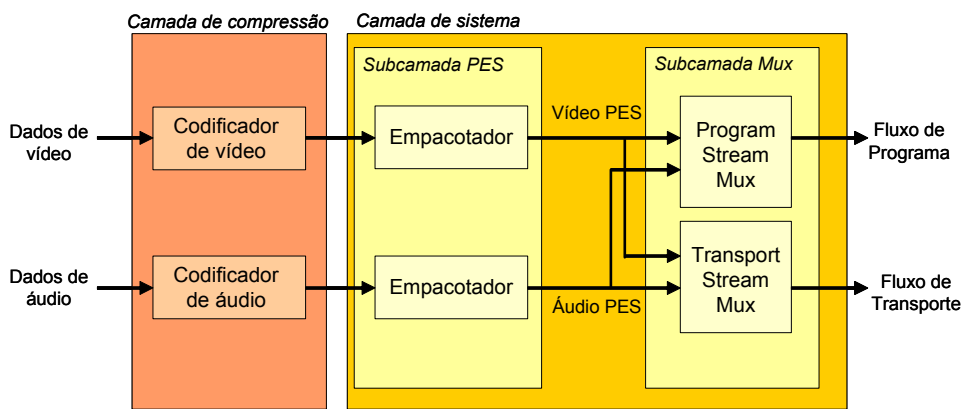
Este capítulo descreve os principais conceitos do padrão MPEG-2, salientando os parâmetros que são diretamente alterados pelas operações de ajuste elástico.

O padrão MPEG-2 foi iniciado em 1990, como uma evolução do MPEG-1, e publicado em 1995. O objetivo deste padrão é prover uma taxa de vídeo de 1,5 Mbps a 15 Mbps, adequados para sinais de televisão padrão (SDTV – *Standard Definition Television*) e taxas de 15 Mbps a 30 Mbps para sinais de televisão de alta definição (HDTV – *High Definition Television*). Para taxas inferiores a 3 Mbps, o padrão MPEG-1 pode apresentar maior eficiência que o MPEG-2.

O padrão MPEG-2 é descrito pelo conjunto de especificações ISO/IEC 13818, cujas principais estabelecem a forma de compressão para o fluxo multiplexado de sistemas (ISO, 2000a), para o vídeo (ISO, 2000b) e para o áudio (ISO, 1998). No âmbito do ITU-T (*International Telecommunication Union – Telecommunication Standardization Sector*), os padrões MPEG-2 *Systems* e MPEG-2 *Video* estão descritos nas recomendações H.222.0 (ITU, 2000a) e H.262 (ITU, 2000b), respectivamente.

**2.1.  
O Fluxo MPEG-2 de Sistemas**

A estrutura de um fluxo definida pelo padrão MPEG-2 pode ser visualizada na Figura 6 e está dividida em duas camadas: a camada de compressão e a camada de sistema. A camada de sistema, definida no padrão *MPEG-2 Systems*, é responsável pela divisão e encapsulamento de cada fluxo comprimido em pacotes; pela inserção de informações de sincronização entre fluxos de mídias diferentes; pela multiplexação dos fluxos encapsulados; e pelo transporte da informação de referência do relógio utilizado no codificador. A camada de compressão refere-se à codificação de cada um dos dados audiovisuais, conforme especificado nos padrões MPEG-2 Áudio e Vídeo.



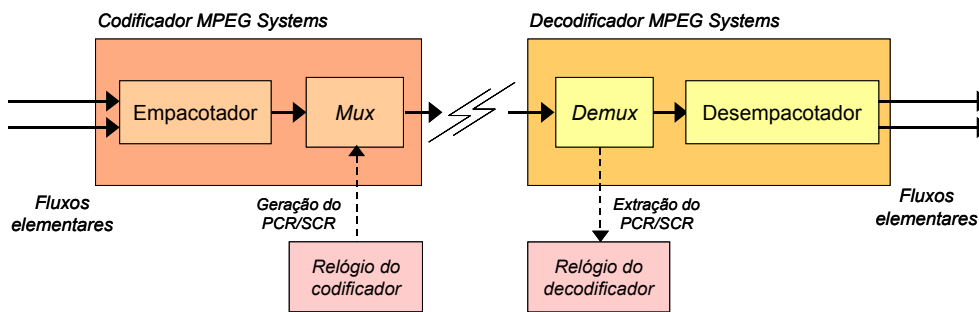
**Figura 6: Estrutura do fluxo MPEG-2.**

Os dados individuais de cada mídia, após sofrerem o processo de compressão, são denominados de fluxos elementares e são divididos em pacotes na subcamada PES (*Packetized Elementary Stream*). As principais funções desempenhadas pela subcamada PES são a identificação exclusiva de cada fluxo, realizada através do parâmetro *stream\_ID*, e a sincronização intra e intermídia, discutida a seguir. Os dados empacotados, ou seja, os fluxos de áudio, vídeo ou dados PES, são enviados à subcamada de multiplexação, onde é inserida a informação de referência de relógio do codificador.

Dois formatos para o fluxo de sistema estão definidos: o Fluxo de Transporte (TS), que contém um ou mais programas e é apropriado para a transmissão e o armazenamento em ambientes ruidosos onde a ocorrência de erros é freqüente; e o Fluxo de Programa (PS), que contém apenas um programa e é adequado para uso em ambientes com baixas taxas de erros. Cada programa é

definido como um conjunto de fluxos elementares, vídeo, áudio e dados, por exemplo, que podem ou não ter algum relacionamento temporal entre si. A codificação dos elementos sincronizados entre si utiliza uma mesma base de tempo, ou referência de relógio.

A sincronização intra e intermídia é obtida através da inserção de marcas de tempo (*time stamps*), tanto nos fluxos PES quanto no fluxo de sistemas. A marca de tempo é uma amostra do contador da respectiva base de tempo, em um determinado instante. As marcas de tempo inseridas no fluxo de sistemas, na subcamada de multiplexação, permitem, ao decodificador, a recuperação da referência do relógio utilizado pelo codificador. Elas são denominadas de *System Clock Reference (SCR)* e *Program Clock Reference (PCR)* para os fluxos TS e PS, respectivamente, e são definidas em termos de um relógio de sistema comum denominado *STC (System Time Clock)*. Os valores das marcas de tempo SCR e PCR significam o instante de tempo em que o último bit desses campos entra no decodificador. O intervalo de tempo máximo permitido entre o envio de duas marcas consecutivas é de 0,7s. O processo de geração e extração das marcas de tempo relativas ao SCR e PCR é ilustrado na Figura 7. O padrão MPEG não considera os efeitos introduzidos pela rede de comunicação.



**Figura 7: Sincronização entre o codificador e o decodificador.**

Após o empacotamento dos respectivos dados em cada PES, alguns pacotes são escolhidos para transportar marcas de tempo. Dentre essas, dois tipos são definidos: o *Presentation Time Stamp (PTS)* e o *Decoding Time Stamp (DTS)*. O PTS indica o instante de tempo em que a unidade de apresentação (figura, para o vídeo, e quadro, para o áudio) deve ser exibida. O DTS, presente apenas no fluxo de vídeo, indica o instante de tempo em que a unidade de apresentação deve ser entregue ao respectivo decodificador e é utilizado quando é necessária a reordenação de quadros, no decodificador.

## 2.2. O Fluxo MPEG-2 de Vídeo

O padrão MPEG-2 de vídeo utiliza algoritmos para eliminar ou reduzir a redundância temporal existente entre quadros consecutivos. As imagens de um vídeo são representadas por quadros de vídeo, os quais são representados por três matrizes retangulares de inteiros: uma matriz de luminância e duas matrizes de crominância. O termo quadro é utilizado tanto para imagens ainda não codificadas através do padrão MPEG-2 quanto para as imagens após a codificação. As informações de um quadro podem ser separadas em campos denominados *top field* e *bottom field*, compostos pelas linhas ímpares ou pares de cada matriz que compõe um quadro, respectivamente.

Uma figura codificada através do padrão MPEG-2 pode representar um quadro ou um campo codificado. Um sinal de vídeo que contenha figuras que representem campos é dito ser um vídeo entrelaçado. Se o fluxo de vídeo contiver apenas figuras que representem quadros, ele é dito ser progressivo.

A estrutura do fluxo codificado através do MPEG-2 de vídeo é hierárquica e contém seis camadas: seqüência, grupo de figuras (GOP), figura, *slice*, macrobloco e bloco, conforme ilustrado na Figura 8.

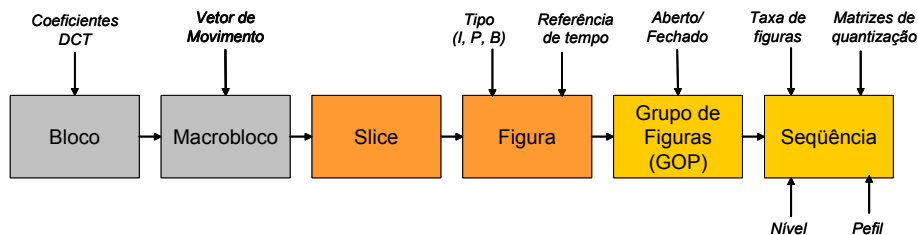


Figura 8: Estrutura do fluxo MPEG Vídeo.

### 2.2.1. Estruturas de um fluxo de vídeo MPEG-2

Um fluxo de vídeo é composto por um conjunto de seqüências, que são utilizadas para transportar informações sobre: as dimensões das figuras e a sua relação de aspecto; as taxas de quadro e de bit; e as matrizes de quantização utilizadas na codificação dos macroblocos, caso os valores-padrão não sejam utilizados. O parâmetro *progressive\_sequence*, presente no elemento *Sequence*

*Extension*, especifica o tipo de sinal de vídeo utilizado, ou seja, se o fluxo de vídeo contém quadros entrelaçados ou não.

A taxa de quadro é fixa, sendo determinada através dos parâmetros *frame\_rate\_code*, da estrutura *Sequence Header* e que indica o valor base (*frame\_rate\_value*) da taxa de quadros, e os parâmetros *frame\_rate\_extension\_n* e *frame\_rate\_extension\_d*, presentes na estrutura *Sequence Extension*. Ela é calculada conforme a equação abaixo:

$$\text{(Eq. 1)} \quad \textit{frame\_rate} = \textit{frame\_rate\_value} * \left[ \frac{\textit{frame\_rate\_extension\_n} + 1}{\textit{frame\_rate\_extension\_d} + 1} \right]$$

Através do parâmetro *frame\_rate\_code*, é possível escolher a taxa de quadros entre 8 valores pré-determinados. Os parâmetros *frame\_rate\_extension\_n* e *frame\_rate\_extension\_d* permitem que a taxa de quadros a ser utilizada seja diferente dos valores pré-determinados.

A camada grupo de figuras (GOP), representada pelo *GOP Header*, é uma estrutura opcional. Quando presente, informa o instante de tempo referente à primeira figura contida no GOP, de acordo com o valor do relógio de referência utilizado pelo codificador e é utilizado para prover acesso aleatório ao fluxo de vídeo. Um GOP também indica a dependência temporal entre figuras pertencentes a GOP diferentes. GOP fechados são aqueles formados apenas por figuras cujas referências localizam-se no próprio GOP. Em um GOP aberto, a decodificação de uma de suas figura pode requerer a decodificação anterior de uma figura localizada em outro GOP, ou seja, pode haver dependências entre GOP distintos.

A camada de figura especifica o tipo de predição utilizado na codificação de uma imagem e fornece a informação de temporização de cada figura. A quantidade de bits utilizada para a codificação de cada figura é variável e é influenciada pelo valor do *quantum* adotado no processo de quantização da imagem.

O método de predição interquadro por compensação de movimento, utilizado para reduzir a redundância temporal entre quadros consecutivos, realiza a comparação de cada macrobloco de uma figura com macroblocos pertencentes a figuras vizinhas. O macrobloco da figura vizinha, escolhida para servir de referência na operação de predição, será aquele que menos se diferenciar do macrobloco a ser codificado. Um vetor de movimento é definido para indicar a

diferença entre as localizações espaciais do macrobloco a ser codificado e o de referência, sendo transmitido junto ao macrobloco codificado. Cada macrobloco especifica sua posição em relação ao macrobloco anterior, a indicação do método de predição utilizado e quais os blocos de luminância e crominância estão codificados. Também contém um fator de escala para permitir o controle do *quantum* do processo de quantização.

A especificação da estrutura de uma figura, estabelecendo se esta representa um campo ou um quadro, é determinada, em um fluxo de vídeo codificado, através do parâmetro *picture\_structure*, presente no elemento *Picture Coding Extension* que está contido em cada figura (elemento *Picture*). Para os propósitos desta dissertação, serão consideradas apenas as figuras que representam quadros. Os termos figura e quadro serão utilizados como sinônimos.

As imagens são codificadas em figuras do tipo I (*Intracoded*), P (*Predictive Coded*) ou B (*Bidirectional Predictive Coded*). Os quadros I são codificados utilizando-se informações contidas no próprio quadro original, ou seja, todos os macroblocos contidos nesses quadros indicam que não há dependência em relação a macroblocos de outras figuras. Os quadros P são codificados de forma preditiva em relação ao quadro I ou P anterior, ou seja, há ao menos um macrobloco contido no quadro que indica a dependência em relação a um macrobloco de outra figura. Por fim, os quadros B são codificados de forma preditiva em relação aos quadros I ou P, anteriores ou posteriores. Há, portanto, ao menos um macrobloco que indica dependência temporal a um quadro a ser apresentado posteriormente. Dessa forma, para a decodificação de um quadro B, é necessário que o quadro posterior, ao qual aquele se referencia, já tenha sido decodificado.

Cada figura codificada possui o parâmetro *temporal\_reference* que funciona como um contador, módulo 1.024, o qual é incrementado a cada novo quadro e é utilizado para que o decodificador possa identificar eventuais perdas de quadros.

O parâmetro *vbv\_delay*, contido no elemento *Picture Header*, indica o tempo que a figura deve permanecer no *buffer* de entrada do decodificador, exceto quando possui valor hexadecimal FFFF. Sua utilização é detalhada na Subseção 2.2.2, onde o funcionamento desse *buffer* é apresentado.

Dois outros parâmetros, contidos no elemento *Picture Coding Extension*, são importantes para o processo de decodificação: *repeat\_first\_field* e *top\_field\_first*. Em vídeos progressivos, eles indicam a quantidade de vezes que

uma figura deve ser apresentada, após sua decodificação. A relação entre esses parâmetros é mostrada na Tabela 1.

**Tabela 1: Parâmetros *repeat\_first\_field* e *top\_field\_first*.**

Parâmetros			Significado
<i>progressive_sequence</i>	<i>repeat_first_field</i>	<i>top_field_first</i>	
1	0	0	Quadro deve ser apresentado uma vez.
1	1	0	Quadro deve ser apresentado duas vezes.
1	1	1	Quadro deve ser apresentado três vezes.

A camada *slice* contém um conjunto de macroblocos, pertencentes a uma mesma linha da imagem codificada. No entanto, nem todos os macroblocos precisam ser inseridos no fluxo comprimido. Aqueles que não são codificados no fluxo comprimido são chamados de *skipped macroblocks*. Em seu cabeçalho, um *slice* especifica a linha de macroblocos a que se refere e o fator de escala utilizado na determinação dos coeficientes DCT.

Essa camada é especificada para facilitar o correto posicionamento espacial das amostras, no processo de exibição da imagem, em casos onde haja perda de dados. A perda de algumas amostras pode causar erro no posicionamento espacial das outras amostras pertencentes ao mesmo *slice*. Esse efeito, porém, não é cumulativo e os outros *slices* podem ser apresentados corretamente.

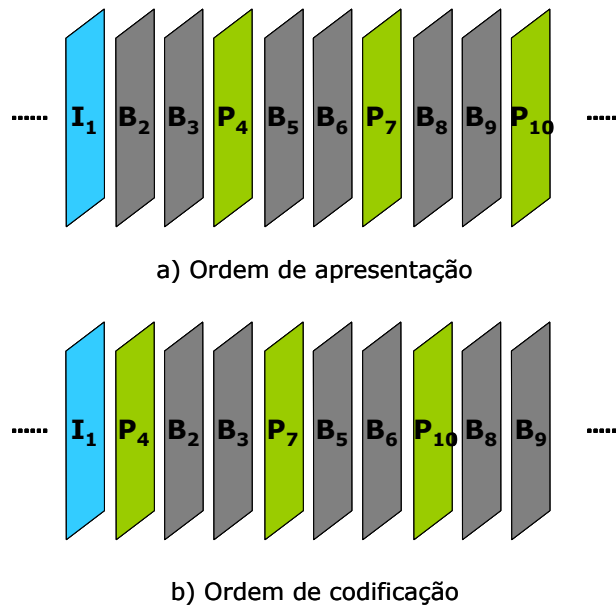
Um resumo das informações mais importantes definidas na sintaxe do padrão MPEG-2 de Vídeo e contidas em cada camada hierárquica pode ser visualizado na Tabela 2.

**Tabela 2: Resumo da sintaxe das camadas do MPEG-2 de Vídeo.**

Nome da Camada	Elementos da sintaxe
<b>Seqüência</b>	Tamanho dos quadros
	Taxa de quadros por segundo
	Taxa de bits por segundo
	Tamanho do <i>buffer</i> de entrada do decodificador
	Parâmetros de codificação programáveis
<b>GOP</b>	Unidade de acesso aleatório
<b>Figura</b>	Informação de temporização ( <i>temporal_reference</i> )
	Tipo de figura
<b>Slice</b>	Informação de endereçamento em relação ao quadro
<b>Macrobloco</b>	Modo de codificação
	Vetores de movimento
	Quantização
<b>Bloco</b>	Coefficientes DCT

Para facilitar a decodificação, a ordenação das figuras no fluxo transmitido, ou ordem de codificação, é diferente da ordem segundo a qual as figuras devem

ser exibidas, também chamada de ordem de apresentação. A Figura 9 exemplifica a disposição das figuras segundo as ordens de apresentação e de codificação.



**Figura 9: Ordenação de figuras.**

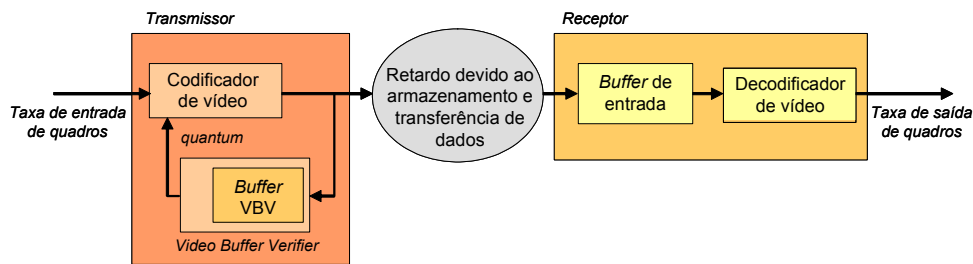
A ordem de codificação garante que as figuras utilizadas como referência na codificação de outras figuras sejam sempre recebidas, pelo decodificador, antes da recepção das figuras que as utilizam como referência. A necessidade da modificação da ordem das figuras no processo de transmissão pode ser visualizada no exemplo acima. Caso o fluxo transmitido fosse composto conforme a ordem de apresentação, a decodificação das figuras B<sub>2</sub> e B<sub>3</sub> só poderiam ocorrer após a decodificação da figura P<sub>4</sub>. Visto que o tipo de cada figura pode ser escolhido livremente pelo codificador, o intervalo de tempo entre a chegada de uma figura do tipo B e a recepção da próxima figura do tipo I ou P (necessária para a decodificação da figura B anterior) poderia ser arbitrariamente longo, dificultando o processo de decodificação.

Com a modificação da ordem das figuras, segundo a definição de ordem de codificação, há a garantia de que sempre que uma figura do tipo B for recebida no decodificador, as figuras I ou P utilizadas como referência no processo de codificação já terão sido recebidas e decodificadas.



**2.2.2. Modelo de transmissão**

O padrão define um modelo de transmissão de dados, representado na Figura 10, segundo o qual os efeitos decorrentes da rede de comunicação e do armazenamento dos dados são desconsiderados e a taxa de exibição de quadros, na saída do decodificador, é mantida constante. A taxa de produção de quadros na saída do codificador é aproximadamente constante, levando-se em consideração também os quadros configurados para serem exibidos mais de uma vez, através dos parâmetros *repeat\_first\_field* e *top\_field\_first*.



**Figura 10: Transmissão no MPEG-2 Vídeo.**

O tamanho do *buffer* de entrada do receptor é determinado pelo codificador e o tamanho dos vários quadros codificados deve ser escolhido de forma a que não haja, exceto em casos especiais, *underflow* nem *overflow* daquele *buffer*.

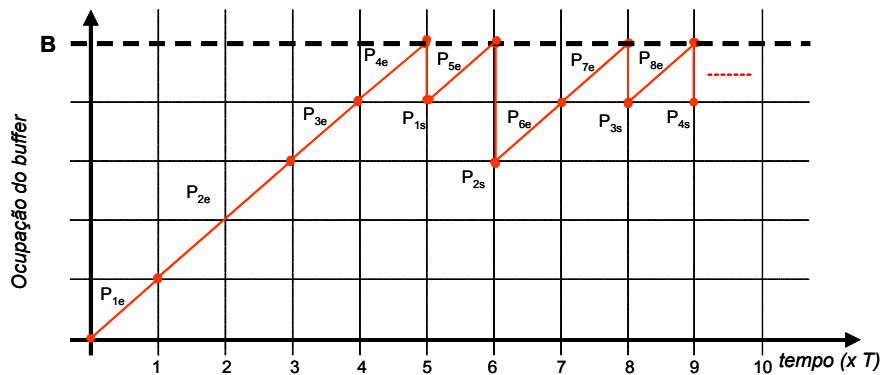
Para que o codificador mantenha o controle da ocupação do *buffer* de entrada do receptor, o modelo define um decodificador hipotético, denominado *Video Buffer Verifier* (VBV), o qual é conceitualmente conectado à saída do codificador. O VBV possui um *buffer* de entrada (*VBV buffer*) cuja ocupação deve corresponder à ocupação do *buffer* de entrada do receptor. Através do monitoramento do estado do *VBV buffer*, ou seja, sua ocupação em bits, o codificador altera o valor do *quantum* utilizado no processo de codificação, produzindo quadros de tamanhos maiores ou menores.

O tamanho B do *VBV buffer* deve ser igual ao do *buffer* de entrada do receptor e é indicado pelo codificador através do parâmetro *vbv\_buffer\_size* presente no cabeçalho de cada seqüência do fluxo de vídeo.

**2.2.2.1. Entrada de dados no VBV buffer**

Duas formas mutuamente exclusivas são definidas para a entrada de dados no *VBV buffer*. Na primeira, representada esquematicamente na Figura 11, o parâmetro *vbv\_delay* de todas as figuras possui o valor hexadecimal FFFF e os dados, enquanto o *buffer* não estiver cheio, sempre entram a uma taxa constante igual a  $R_{m\acute{a}x}$ , em bits por segundo. Os índices *e* e *s* indicam se a respectiva figura está entrando ou saindo do *buffer*, respectivamente. A Figura 11 considera que o quadro  $P_2$  é configurado para ser apresentado duas vezes. Sempre que a ocupação do *buffer* atinge seu nível máximo, o codificador interrompe a geração de quadros, que é restabelecida apenas no momento em que há novamente espaço disponível no *buffer*.

O processo de decodificação se inicia apenas no momento em que o *buffer* apresenta-se cheio pela primeira vez e prossegue retirando quadros de acordo com a taxa de exibição de quadros.



**Figura 11: Ocupação do VBV buffer quando *vbv\_delay* é igual a FFFF.**

A segunda forma, representada simplificada na Figura 12, permite o uso de taxa variável de bits para cada quadro. Nessa situação, o parâmetro *vbv\_delay* possui valor hexadecimal diferente de FFFF e informa o tempo que a respectiva figura deve permanecer no *buffer*. Para manter a taxa de geração de quadros aproximadamente constante, a taxa de transmissão, em bits por segundo, deve satisfazer a seguinte equação (ITU, 2000b):

$$(Eq. 2) R(n) = \frac{d_n^*}{(\tau(n) - \tau(n+1) + t(n+1) - t(n))}$$

Onde  $R(n)$  é a taxa de transmissão, em bits por segundo, da figura  $n$ ;  $d_n^*$  é a quantidade de bits da figura  $n$ , incluindo os bits de todas as camadas seqüência e GOP até o início da próxima figura;  $\tau(n)$  é o tempo que a figura  $n$  deve permanecer no *buffer*, informado pelo parâmetro *vbv\_delay* da figura  $n$  e medido em segundos; e  $t(n)$  é o instante, medido em segundos, em que a figura  $n$  deve ser retirada do *buffer*. Os parâmetros  $d_n^*$ ,  $\tau(n)$  e  $t(n)$  são determinados pelo codificador para que a taxa de transmissão  $R(n)$  garanta que situações de *overflow* e *underflow* não ocorrem.

Para figuras cuja apresentação não deve ser repetida, a equação estabelece que a taxa, em bits por segundo, a ser utilizada para a codificação de uma figura, deve garantir que todos os seus dados entrem no *buffer* em um intervalo de tempo igual ao inverso da taxa de quadros do vídeo. Se a figura for configurada para ser exibida mais de uma vez, a taxa, em bits por segundo, deve garantir que o intervalo de tempo de entrada da figura no *buffer* seja igual ao número de vezes que a figura deve ser exibida, multiplicado pelo inverso da taxa de quadros.

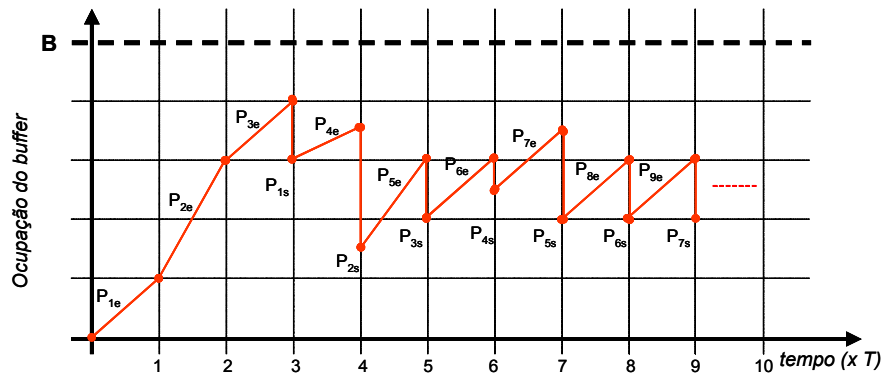


Figura 12: Ocupação do *VBV buffer* quando *vbv\_delay* é diferente de *FFFF*.

O processo de decodificação de uma figura inicia-se quando o intervalo de tempo indicado pelo *vbv\_delay* é decorrido. Nesse instante, a respectiva figura é retirada do *buffer*. No esquema mostrado na Figura 12, o *vbv\_delay* de cada figura foi considerado igual a  $3 * T$  e nenhuma figura foi configurada para ser repetida.

### 2.2.2.2.

#### Saída de dados do VBV buffer

Para a retirada de dados do *VBV buffer*, no caso de vídeos não entrelaçados, o decodificador examina o conteúdo do *buffer* a intervalos sucessivos, múltiplos do intervalo de tempo  $T$ , definido como o inverso da taxa de quadros. As regras que disciplinam o momento de exame do *buffer* dependem da existência de figuras do tipo B no fluxo, do tipo da última figura retirada do *buffer* e da indicação de repetição de exibição das figuras. Elas baseiam-se em que uma figura só deve ser retirada do *VBV buffer* quando for necessária sua decodificação, o que ocorre devido à necessidade de sua apresentação ou quando a figura foi utilizada como referência no processo de codificação preditiva da próxima figura a ser retirada do *buffer*. Essas regras podem ser resumidas nos seguintes critérios:

*I – Para estruturas de seqüência que não contenham quadros do tipo B*

- a) Se a última figura retirada do *buffer*, do tipo I ou P, tiver sido configurada para ser exibida uma, duas ou três vezes, a próxima figura deve ser retirada do *buffer* após o intervalo de tempo igual a  $T$ ,  $2*T$  ou  $3*T$ , de acordo com o número de repetições solicitadas, respectivamente.

*II – Para estruturas de seqüência que contenham quadros do tipo B*

- a) Se a última figura retirada do *buffer* tiver sido do tipo B e configurada para ser exibida uma, duas ou três vezes, a próxima figura deve ser retirada do *buffer* após o intervalo de tempo igual a  $T$ ,  $2*T$  ou  $3*T$ , de acordo com o número de repetições solicitadas, respectivamente.
- b) Se a última figura retirada do *buffer* tiver sido do tipo I ou P e a figura I ou P anteriormente recebida tiver sido configurada para ser exibida uma, duas ou três vezes, a próxima figura deve ser retirada do *buffer* após o intervalo de tempo igual a  $T$ ,  $2*T$  ou  $3*T$ , de acordo com o número de repetições solicitadas, respectivamente..

### 2.2.3.

#### Controle de ocupação do *buffer*

O gerenciamento da ocupação do *buffer* de entrada do receptor é realizado através da manipulação da quantidade de bits de cada figura, da taxa de bits do fluxo codificado e do tamanho do *buffer*, definido pelo codificador. O padrão MPEG-2 sugere a adoção do controle de taxa do modelo TM5 (*Test Model 5*), cujo objetivo é determinar, de forma adaptativa, o tipo de predição temporal para cada macrobloco, a matriz de quantização a ser aplicada e a taxa de transmissão, em bits por segundo, de cada figura, a fim de evitar situações de *overflow* e *underflow* do *buffer* do receptor. As operações são realizadas nas camadas GOP, de figura e de macrobloco, e são compostas por três atividades: a alocação de bits, o controle de taxa e a quantização adaptativa.

Na primeira etapa, um número fixo de bits é alocado para cada GOP, obtido através da taxa máxima de transmissão desejada, em bits por segundo, e o número de quadros contidos no GOP. Essa etapa é responsável pela estimativa do número de bits disponível para a codificação da próxima figura, sendo realizada, portanto, antes da codificação de cada quadro. Depois, à medida que cada figura é codificada, a ocupação do *buffer* é monitorada, realimentando o sistema e definindo um valor de referência do *quantum* a ser utilizado por cada macrobloco. Por último, a quantização adaptativa manipula o valor de referência de acordo com a atividade espacial em cada macrobloco para determinar o valor exato do *quantum* relativo à codificação do respectivo macrobloco.

Detalhando o algoritmo especificado pelo TM5, inicialmente, é calculado o total de bits que o GOP deve conter, de acordo com a taxa de bits desejada, a taxa de quadros e o número de quadros dos tipos I, P e B que ainda devem ser inseridos no GOP corrente, conforme a equação abaixo:

$$\text{(Eq. 3)} \quad R_{GOP} = (N_i + N_p + N_b) \times \left( \frac{\text{bit\_rate}}{\text{frame\_rate}} \right)$$

Em seguida, são determinadas as “medidas de complexidade globais” (*global complexity measures*), denominadas de  $X_i$ ,  $X_p$  ou  $X_b$ , de acordo com o tipo de quadro, segundo as seguintes fórmulas:

$$X_i = S_i Q_i$$

**(Eq. 4)**  $X_p = S_p Q_p$

$$X_b = S_b Q_b$$

Onde  $S_i$ ,  $S_p$  e  $S_b$  representam o número de bits, conforme seu tipo, e as variáveis  $Q_i$ ,  $Q_p$  e  $Q_b$  são os parâmetros de quantização médios, calculados de acordo com a média dos valores de quantização utilizados durante a codificação de todos os macroblocos. Todos os parâmetros anteriores referem-se às últimas figuras codificadas, de acordo com o respectivo tipo. Os valores iniciais para os parâmetros de quantização médios são:

$$X_i = \frac{(160 \times bit\_rate)}{115}$$

**(Eq. 5)**  $X_p = \frac{(60 \times bit\_rate)}{115}$

$$X_b = \frac{(42 \times bit\_rate)}{115}$$

A quantidade de bits ( $T_i$ ,  $T_p$  e  $T_b$ ) a ser utilizada na codificação da próxima figura é determinada conforme as equações abaixo. Os parâmetros  $K_p$  e  $K_b$  são constantes cujos valores atribuídos são 1.0 e 1.4, respectivamente:

$$T_i = \max \left\{ \frac{R_{GOP}}{\left( 1 + \frac{N_p X_p}{X_i X_p} + \frac{N_b X_b}{X_i K_b} \right)}, \frac{bit\_rate}{8 \times picture\_rate} \right\}$$

**(Eq. 6)**  $T_p = \max \left\{ \frac{R_{GOP}}{\left( N_p + \frac{N_b K_p X_b}{K_b X_p} \right)}, \frac{bit\_rate}{8 \times picture\_rate} \right\}$

$$T_b = \max \left\{ \frac{R_{GOP}}{\left( N_b + \frac{N_p K_b X_p}{K_p X_b} \right)}, \frac{bit\_rate}{8 \times picture\_rate} \right\}$$

A segunda etapa refere-se ao controle da taxa e destina-se à obtenção do parâmetro de quantização  $Q_j$  referente ao macrobloco  $j$ , conforme a expressão:

$$(Eq. 7) \quad Q_j = \left( \frac{d_j \times 31}{2} \right) \left( \frac{picture\_rate}{bit\_rate} \right)$$

onde  $d_j$  representa o nível de ocupação do *buffer* virtual e é calculado, de acordo com o tipo de figura.

A terceira etapa determina o valor de cada elemento da matriz de quantização a ser utilizado, a partir da atividade espacial ( $act_j$ ) calculada para cada macrobloco e o respectivo parâmetro de quantização  $Q_j$ . O modelo TM5 especifica que:

$$(Eq. 8) \quad act_j = 1 + \min(vblk_1, vblk_2, \dots, vblk_8)$$

$$(Eq. 9) \quad vblk_n = \frac{1}{64} \times \sum_{k=1}^{64} (P_k^n - P\_mean_n)^2$$

$$(Eq. 10) \quad P\_mean_n = \frac{1}{64} \times \sum_{k=1}^{64} P_k^n$$

O valor da atividade ( $act_j$ ) é, então, normalizado, obtendo-se  $N\_act_j$  e, finalmente, determinando-se o valor do *quantum* a ser aplicado ao respectivo macrobloco:

$$(Eq. 11) \quad mquant_j = Q_j \times N\_act_j$$

Além do modelo TM5 do MPEG, outros algoritmos para o controle da ocupação do *buffer* do receptor foram propostos, tais como: o algoritmo de alocação de bits proposto por Song e Chun (Song & Chun, 2003); o esquema de controle de taxa através de histogramas baseado em estimativas de taxa e distorção (*Rate-Distortion Estimation*) proposto por Hong *et al.* (Hong *et al.*, 2003); e o algoritmo proposto por He e Mitra (He & Mitra, 2002), que adota um relacionamento linear entre a taxa de codificação de bits e o percentual de zeros nos coeficientes DCT.

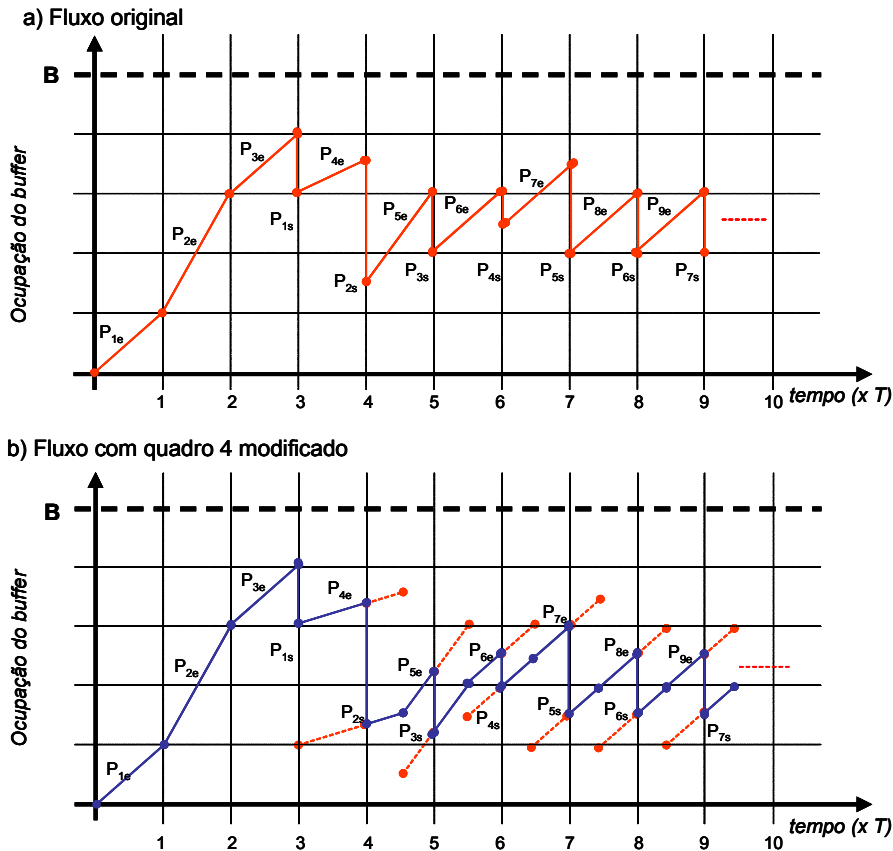
### 2.2.3.1. Situações de *overflow* e *underflow*

A ocorrência de *overflow* ou *underflow* do *buffer* de entrada do receptor é gerada por problemas no processo de codificação, alterando a taxa de produção de quadros em relação à taxa nominal. A taxa de codificação de quadros pode sofrer variações devido aos métodos utilizados para o cálculo da quantidade de bits e da taxa de transmissão dos dados, como, por exemplo, o estabelecido pelo TM5. O processo de codificação é responsável por, mesmo havendo pequenos desvios da taxa instantânea de codificação de quadros em relação à taxa nominal, manter a ocupação do *buffer* dentro dos limites adequados, evitando o *overflow* ou *underflow*. Fatores externos ao padrão MPEG também podem causar perturbações no decodificador, tais como os decorrentes da rede de comunicação.

A situação de *underflow* ocorre se a taxa de quadros gerada na saída do codificador for inferior à taxa de quadros prevista para o fluxo de vídeo. Nessa situação, o decodificador consome quadros do *buffer* mais rapidamente do que novos quadros entram no mesmo. Se essa situação persistir por um tempo suficientemente grande, todos os quadros do *buffer* serão retirados e o decodificador não encontrará um novo quadro completo para ser apresentado.

A diminuição da taxa de codificação de quadros, mesmo que temporariamente, faz com que a ocupação do *buffer* diminua, em relação à curva de ocupação em que os quadros são gerados à taxa constante. Essa situação pode ser observada no esquema apresentado na Figura 13, em que o segundo gráfico é obtido quando o quadro  $P_4$  é codificado utilizando uma taxa, em bits por segundo, inferior à necessária, dado a quantidade de bits do quadro. A linha cheia representa a curva de ocupação do *buffer*, enquanto que as linhas tracejadas são mostradas apenas para facilitar a visualização do tamanho de cada quadro. Cabe ao codificador perceber a diminuição momentânea da taxa de codificação de quadros e fazer a compensação nos quadros seguintes.





**Figura 13: Efeito da diminuição da taxa de codificação de quadros.**

O mesmo efeito é obtido se a rede de comunicação limitar a taxa de transferência dos dados em um valor menor que a máxima taxa, em bits por segundo, configurada nas figuras do fluxo de vídeo. Nessa situação, o codificador opera corretamente, mas a transmissão dos dados introduz a perturbação que pode causar o *underflow*.

O *overflow* ocorre quando a taxa de entrada de quadros no *buffer* é superior à taxa nominal de apresentação de quadros, por um tempo suficientemente grande. Os quadros se acumulam do *buffer* até que sua ocupação alcance o limite máximo. A partir desse momento, novos dados que cheguem ao *buffer* são descartados, gerando perda de quadros. Situações de *overflow* também podem acontecer se o algoritmo de alocação de bits, no processo de codificação, for inadequado e não limitar corretamente a quantidade de bits de cada figura. Essa condição é representada na Figura 14.

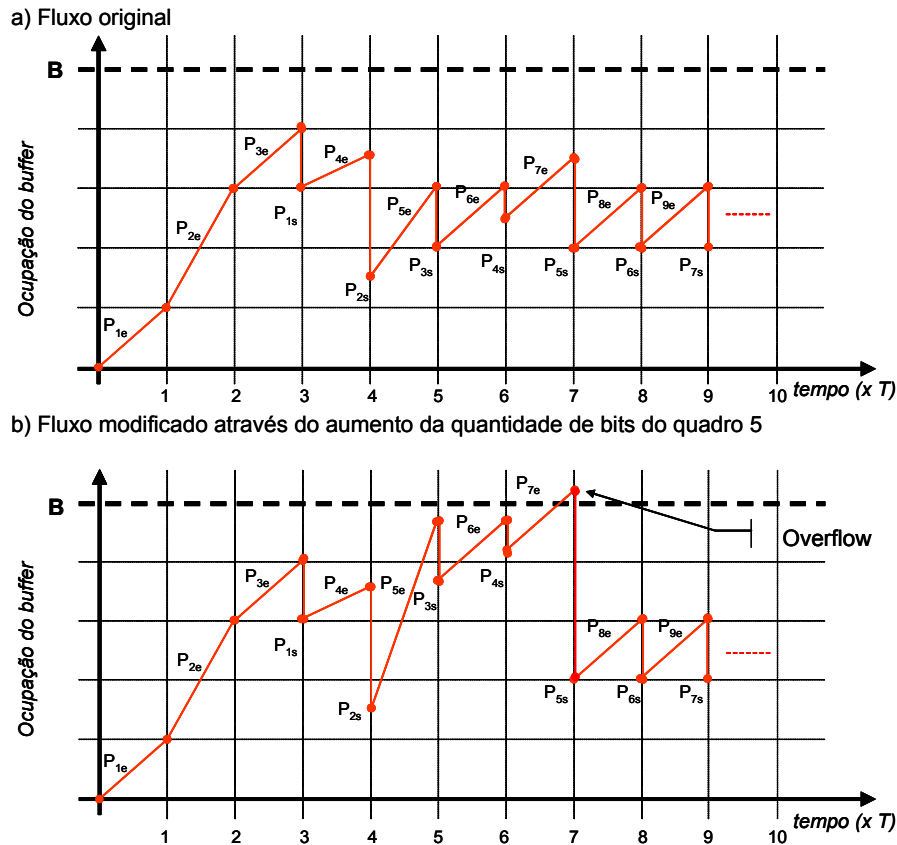


Figura 14: *Overflow* devido a erro na alocação de bits.

Situações de *overflow* e *underflow* também podem ser provocadas por falhas na rede de comunicação devido a perdas de pacotes, fazendo com que a seqüência de quadros que entrem no *buffer* seja diferente da gerada pelo codificador. A perda de quadros, no entanto, prejudica a correta operação do decodificador, pois, caso tenham sido perdidos quadros configurados para terem sua apresentação repetida, o decodificador não recebe essa informação, adiantando a exibição do quadro seguinte. O funcionamento é normalizado após o recebimento de um novo GOP, que provê informações temporais para o acesso aleatório do fluxo.