

UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

**PRODUÇÃO DE VÍDEO DIGITAL:
TEORIA E PRÁTICA**

Versão 1

RUDINEI GOULARTE
EDSON DOS SANTOS MOREIRA

NOTAS DIDÁTICAS DO ICMC



São Carlos – SP

Instituto de Ciências Matemáticas de São Carlos

ISSN - 0103-2577

**PRODUÇÃO DE VÍDEO DIGITAL:
TEORIA E PRÁTICA
Versão 1**

**RUDINEI GOULARTE
EDSON DOS SANTOS MOREIRA**

Nº 30

NOTAS DIDÁTICAS DO ICMSC

**São Carlos
Fev./1998**

PRODUÇÃO DE VÍDEO DIGITAL:

TEORIA E PRÁTICA

Rudinei Goularte

Edson dos Santos Moreira

Versão 1

Fevereiro de 1998

1 - Áudio e Vídeo Digitais	pág. 3
1.1 - Introdução	pág. 3
1.2 - Áudio	pág. 3
1.2.1 - Áudio Digital	pág. 4
1.2.1.1 - Digitalização de Áudio	pág. 4
1.2.1.2 - Captura e Codificação de Sinais de Áudio	pág. 6
1.3 - Vídeo	pág. 7
1.3.1 - Câmera de Vídeo	pág. 7
1.3.2 - Rastreamento	pág. 8
1.3.3 - Vídeo Digital	pág. 9
1.3.3.1 - Captura e Codificação de Vídeo	pág. 10
1.4 - Técnicas de Compressão e Codificação	pág. 11
1.4.1 - Classificação das Técnicas de Compressão	pág. 12
1.4.2 - Técnicas de Compressão para Áudio Digital	pág. 12
1.4.2.1 - mu-law e A-law PCM	pág. 13
1.4.2.2 - <i>Adaptive Differential Pulse Code Modulation</i> (ADPCM)	pág. 13
1.4.2.3 - G.721 e G.723	pág. 14
1.4.2.4 - <i>Linear Predictive Coding</i> (LPC) e <i>Code Excited Linear Prediction</i> (CELP)	pág. 14
1.4.2.5 - MPEG	pág. 15
1.4.3 - Técnicas de Compressão para Vídeo Digital	pág. 16
1.4.3.1 - JPEG e MJPEG	pág. 16
1.4.3.2 - H.261	pág. 18
1.4.3.3 - Cell	pág. 18
1.4.3.4 - CU-SeeMe	pág. 19
1.4.3.5 - MPEG	pág. 19
2 - Estruturação e Intercâmbio de Documentos Hipermedia	pág. 21
2.1 - Introdução	pág. 21
2.2 - SGML	pág. 21
2.3 - HTML	pág. 23
2.4 - <i>HyTime</i>	pág. 23
2.5 - MHEG	pág. 24
2.6 - <i>QuickTime</i>	pág. 26
2.7 - <i>QuickTime</i> para <i>Windows</i>	pág. 28
2.8 - AVI	pág. 30
3 - Aspectos para o Desenvolvimento de Sistemas Hipermedia Distribuídos	pág. 33
3.1 - Introdução	pág. 33
3.2 - Qualidade de Serviço	pág. 33
3.3 - Serviço de Vídeo sob Demanda	pág. 35
3.4 - Sistema Operacional	pág. 37

3.4.1 - Problemas com os Sistemas Operacionais Convencionais	pág. 38
3.4.2 - Gerenciamento de Recursos	pág. 39
3.4.3 - Suporte Arquitetural	pág. 40
3.5 - Sistemas de Comunicação	pág. 40
3.5.1 - A Abordagem ATM (<i>Asynchronous Transfer Mode</i>)	pág. 43
3.6 - Armazenamento e Acesso	pág. 45
3.7 - Busca e Recuperação	pág. 46
4 - Produção de Vídeo Digital para Computadores	pág. 50
4.1 - Introdução	pág. 50
4.2 - Produção de Vídeo para Computadores	pág. 51
4.3 - Apresentação de Vídeo em Redes de Computadores	pág. 54
Glossário de Termos	pág. 58
Referências Bibliográficas	pág. 62

1.1 - INTRODUÇÃO

Áudio e Vídeo são entidades analógicas conhecidas como som e imagem. Para serem usados por um computador, som e imagem devem ser transformados de sua representação **mecânica/óptica** para uma representação **eletrônica (analógica)** e então de **analógica** para **digital**. Sistemas multimídia envolvem fortemente a captura, a transmissão e a apresentação de áudio e vídeo digitais, logo é importante conhecer os mecanismos de captura, digitalização e codificação de imagens e sons, bem como suas técnicas de compressão, uma vez que estas são largamente utilizadas devido ao grande espaço de armazenamento necessário para esses tipos de dados.

Este capítulo está dividido em três partes: áudio digital, vídeo digital e, técnicas de compressão. A seção sobre áudio começa com uma breve descrição sobre áudio e áudio digital. Em seguida são apresentados os processos de digitalização, captura e codificação de sinais de áudio. A seção sobre vídeo apresenta: uma breve descrição de vídeo; o conceito de rastreamento; o funcionamento das câmeras de vídeo; apresenta vídeo digital e os métodos de captura e codificação de vídeo. Por último, este capítulo discute as técnicas de compressão e suas classificações, apresentando algumas das técnicas mais utilizadas.

1.2 - ÁUDIO

O som é um fenômeno causado por rápidas variações de pressão no ar que se propagam em todas as direções. Quando essas variações chegam aos nossos ouvidos (que funcionam como sensores/transdutores), elas são transformadas em um sinal elétrico e enviadas ao cérebro, então ouvimos o som. Em sistemas de computação, o componente que realiza o papel de

sensor/transdutor é o microfone. Um microfone funciona de maneira semelhante ao ouvido, captando as variações de pressão no ar e transformando-as em um sinal elétrico (analógico) chamado **sinal de áudio** [Lut95] (figura 1.1).

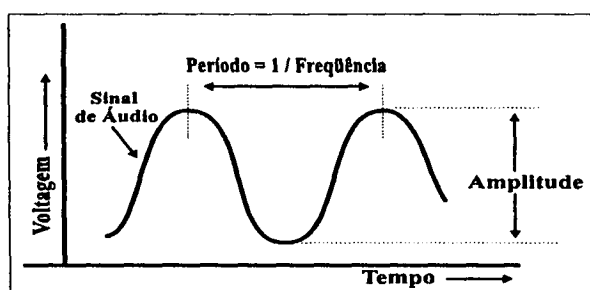


Figura 1.1 - Um Sinal de Áudio [Sis96].

Um sinal de áudio pode ser caracterizado em termos de sua **frequência** e de sua **amplitude**. A frequência é a taxa com que o sinal de áudio varia de escalas positivas para negativas e vice-versa. Essa taxa é medida em *Hertz* (Hz). A amplitude é a máxima oscilação positiva e negativa da escala do sinal de áudio [Lut95]. A escala do sinal de áudio é medida em volts.

1.2.1 - ÁUDIO DIGITAL

Para podermos utilizar áudio em um computador devemos converter o sinal analógico em sinal digital, através de um processo chamado **digitalização**. Uma vez que um sinal de áudio tenha sido digitalizado, um computador poderá realizar seu armazenamento, fazer cópias e reproduções.

1.2.1.1 - DIGITALIZAÇÃO DE ÁUDIO

Um sinal analógico possui duas dimensões (tempo e tensão) que podem ser digitalizadas. A digitalização envolve os processos de **amostragem** e de **quantização**.

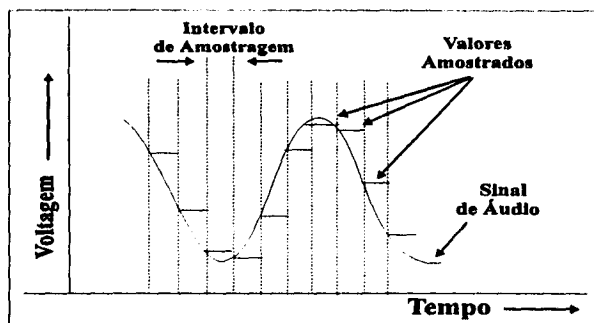


Figura 1.2 - Amostragem [Lut95].

A amostragem converte a escala de tempo para uma escala digital, realizando uma leitura instantânea da voltagem em um espaço de tempo uniforme. A taxa de amostragem é o número de amostras tomadas por segundo e, deve ser (teoricamente) pelo menos duas vezes a frequência de áudio mais alta a ser reproduzida [Lut95] (figura 1.2).

A quantização converte os valores das amostras em valores digitais (antes da quantização os valores das amostras são ainda analógicos, pois refletem exatamente a voltagem analógica no instante da amostragem). A escala de voltagem é dividida, convertida em números de faixas de larguras iguais ou níveis - o número exato de níveis depende do número de *bits* designado para cada amostra e cada nível corresponde a um valor numérico que é representado digitalmente. Esse processo é usualmente chamado de PCM (*Pulse Code Modulation*) linear. Por exemplo, se usássemos 8 *bits* para representar cada amostra, então a escala teria 256 níveis ($256 = 2^8$) porque cada *bit* tem somente dois valores, 0 ou 1 [Lut95]. A figura 1.3 mostra 16 níveis correspondentes a 4 *bits* por amostra.

O processo de quantização das amostras invariavelmente resulta em distorções do sinal, e essas distorções devem ser mantidas dentro de limites aceitáveis. No PCM linear, em geral, o número de níveis é uma potência de 2, e um número elevado de níveis de amplitude mantém as distorções dentro de valores toleráveis. Esse processo pode ser melhorado através de técnicas como quantização PCM não linear logarítmica [Pan93].

Um circuito que representa amostragem e quantização é chamado de conversor analógico digital (ADC). Para utilização em áudio, ADCs operam em uma taxa de amostragem entre 8 kHz (*kiloHertz*) e 44.000 kHz e entre 8 e 16 *bits* por amostra.

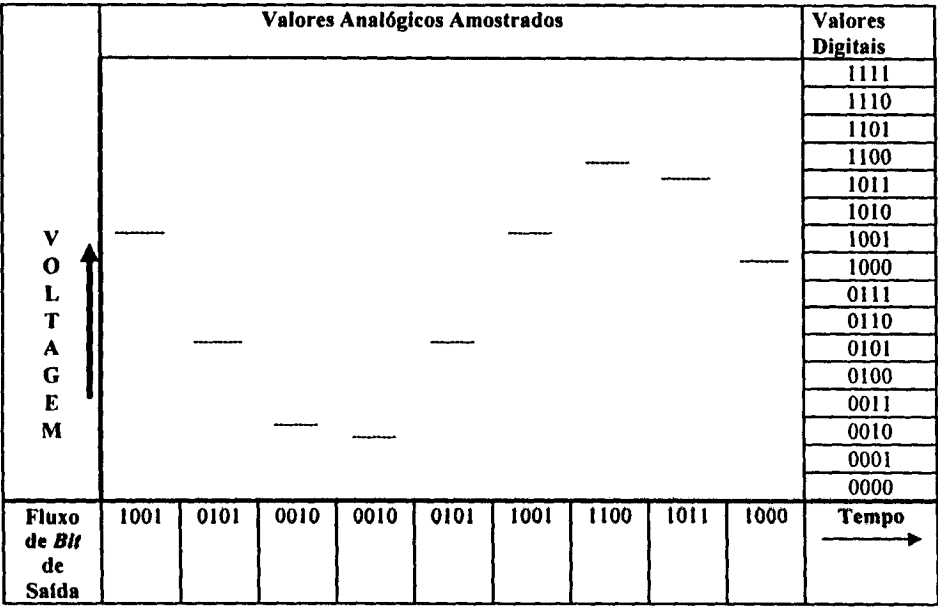


Figura 1.3 - Quantização [Lut95].

Para reproduzir o áudio nas caixas ou fones de ouvido, o áudio digital deve ser convertido novamente em áudio analógico, o que requer um conversor digital analógico (DAC). Um circuito DAC deve ser construído para o mesmo número de *bits* por amostra referente ao áudio digital que se está convertendo. Se o DAC não pode processar todos os *bits* no fluxo de dados do áudio, a qualidade de saída do mesmo é degradada.

1.2.1.2 - CAPTURA E CODIFICAÇÃO DE SINAIS DE ÁUDIO

A captura de sinais de áudio exige processamento em tempo real e normalmente isto é feito por sistemas especializados no processamento de sinais. A taxa de amostragem e o número de *bits* por amostra necessários, respectivamente para captura e codificação, variam em função do tipo de sinal de áudio e da aplicação. Esses parâmetros também influenciam na quantidade de memória necessária para armazenamento e na taxa de transmissão do sinal digitalizado. Soares [Soa92] nos mostra esses parâmetros de acordo com o tipo de aplicação, onde vemos que o espaço requerido para o armazenamento de áudio é grande. Porém, técnicas de compressão de áudio podem ser aplicadas diretamente ao sinal no momento da gravação. Por exemplo, o uso de um codificador ADPCM com 4 *bits* por amostra permite uma gravação de áudio padrão CD utilizando-se apenas um quarto do valor necessário sem a compactação.

1.3 - VÍDEO

Nós percebemos imagens de uma forma paralela, isto é, todos os objetos na imagem refletem luz ao mesmo tempo e o olho humano capta tais reflexões paralelamente. O nervo óptico possui milhões de conexões que levam a informação paralelamente até o cérebro. No mundo eletrônico, porém, uma conexão com milhões de circuitos é impraticável, de modo que utilizamos uma única conexão ou algumas poucas conexões, mais fáceis de manusear, e um sensor de vídeo (câmera) que converte a imagem paralela, através de um único sinal elétrico, em um sinal de vídeo.

Esta seção explica a composição das câmeras, como elas obtêm imagens e como um sinal digital de vídeo é gerado e codificado.

1.3.1 - CÂMERA DE VÍDEO

As câmeras de vídeo são os equipamentos utilizados para a captura de imagens. Elas podem ser monocromáticas, reproduzindo somente imagens em preto e branco, ou podem reproduzir imagens coloridas utilizando-se da combinação das três cores primárias (vermelho, verde e azul) para a reprodução de cor [Lut95].

Uma câmera de vídeo monocromática consiste de uma lente que juntamente com uma superfície sensível à luz (fotossensível) converte uma imagem luminosa através de um padrão de carga elétrica. Por meio da técnica de rastreamento, o padrão de carga elétrica vai sendo lido, utilizando-se para isto um feixe disparado por um canhão de elétrons que se encontra no interior de um tubo a vácuo. O rastreamento passa sobre cada ponto da superfície da imagem, a carga do ponto é convertida para um sinal elétrico na linha de saída. A figura 1.4 mostra o resultado de um rastreamento de uma figura retangular [Lut95].

Câmeras coloridas realizam a reprodução de vídeo utilizando três câmeras que focalizam a mesma imagem. Cada câmera possui um filtro apropriado para uma determinada cor primária. Um prisma é utilizado para dividir a luz de maneira idêntica entre as câmeras, de modo que cada

câmera veja a mesma imagem. Esse esquema é denominado **câmera RGB¹** de três sensores [Lut95] (figura 1.5).

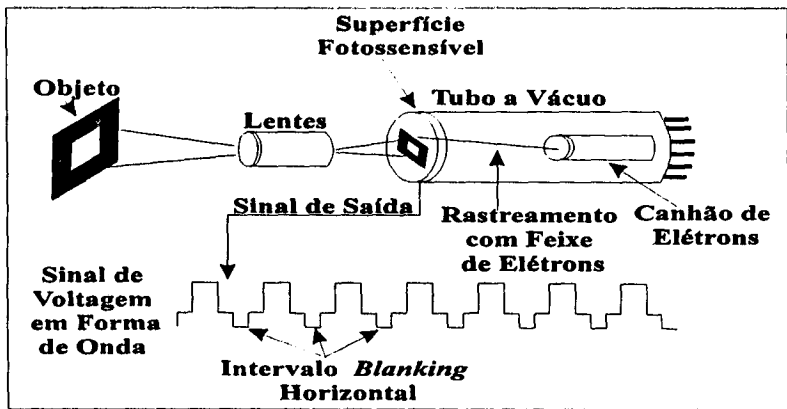


Figura 1.4 - Uma Câmera de Vídeo Monocromática e seu Sinal de Saída [Lut95].

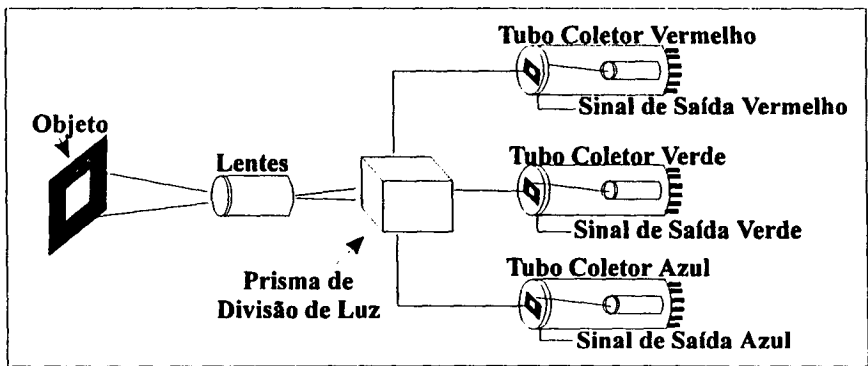


Figura 1.5 - Uma Câmera Colorida com Três Sensores [Lut95].

1.3.2 - RASTREAMENTO

O processo de rastreamento (*scanning*) consiste em capturar eletricamente uma imagem através de uma leitura sequencial dos valores de brilho de uma série de pontos (figura 1.4). O rastreamento produz como saída um sinal analógico (elétrico) representando o brilho de um

¹ RGB - Do inglês : Red, Green e Blue. É um padrão que utiliza a combinação das três cores primárias para obter outras cores.

ponto da imagem. Se o processo for realizado a uma velocidade variando de 30 a 60 vezes por segundo, os olhos perceberão uma imagem contínua.

A figura 1.6 mostra um padrão típico de rastreamento. O movimento de leitura da imagem inicia-se no canto superior esquerdo, realizando sucessivos retraços horizontais. Quando a leitura atinge o canto inferior direito da imagem, ocorre um retraço vertical e a leitura recomeça no canto superior esquerdo, completando um *frame* de rastreamento. Luther [Lut95] explica os processos de rastreamento e de retraço em maiores detalhes.

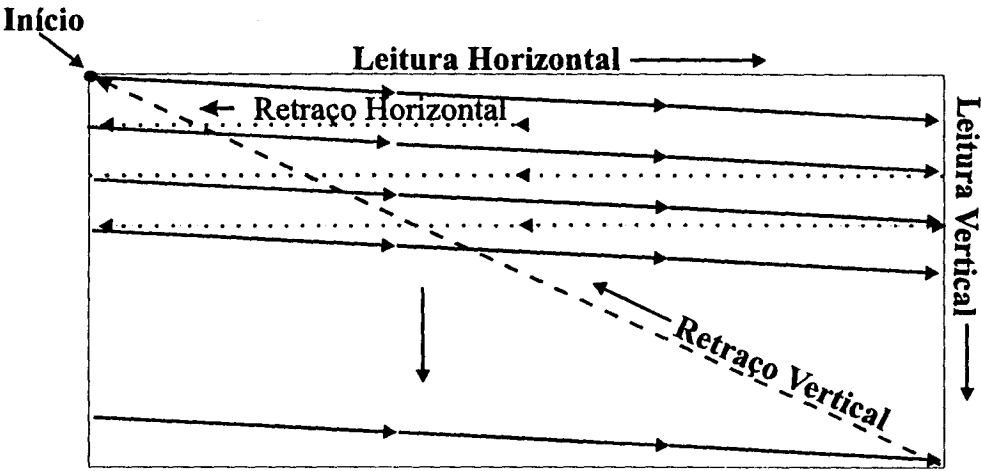


Figura 1.6 - Rastreamento Progressivo [Lut95].

1.3.3 - VÍDEO DIGITAL

Um sinal de áudio digital e um sinal de vídeo digital monocromático são similares a não ser pela estrutura das linhas e *frames* obtidos do rastreamento. A digitalização segue exatamente o mesmo processo que o descrito para áudio - amostragem e quantização. A diferença reside no fato de que a taxa de amostragem produz um número específico de amostras por linha e essas amostras representam pontos individuais chamados *pixels*, como ilustrado na figura 1.7 [Lut95].

A resolução em um sistema de vídeo digital é determinada pelo número de *pixels* horizontais e verticais; por exemplo, um monitor VGA tem 640 x 480 *pixels*, ou seja, 640 *pixels* horizontais e 480 *pixels* verticais. Dessa forma a representação de uma imagem digital é feita por meio de um vetor de duas dimensões de amostras, e o padrão de medida do valor do ponto é em

bits por pixel (bpp) [Lut95]. A precisão de leitura do *pixel* determina os muitos níveis de intensidade com que ele pode ser representado, e é expresso por um número de *bits*/amostra. De acordo com essa precisão, as imagens podem ser classificadas em [Fur95]:

- **Imagens Binárias** representadas por 1 *bit*/amostra. Por exemplo, fotografias em branco e preto e imagens de *fac-símile*.

- **Gráficos de Computadores**, representados por 4 *bits*/amostra.

- **Imagens Grayscale** representadas por 8 *bits*/amostra.

- **Imagens Coloridas** representadas com 16, 24 ou mais *bits*/amostra.

Além do padrão RGB, existe um outro para a representação de imagens coloridas chamado YUV, que descreve componentes de **luminância** e de **crominância** de uma imagem [Fur95].

Tem-se ainda o YCrCb, que é uma representação de cores intensamente utilizada para compressão de imagens, como por exemplo, no padrão MPEG [Pat93].

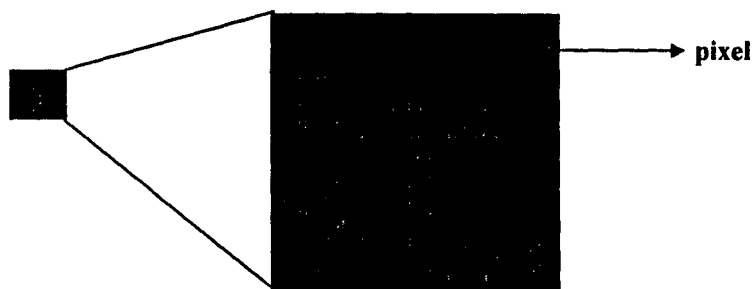


Figura 1.7 - Conjunto de Pontos de Amostragem (*pixels*).

1.3.3.1 - CAPTURA E CODIFICAÇÃO DE VÍDEO

Para a realização da captura de sinais de vídeo os sinais gerados por câmeras, videocassetes ou estações de TV são digitalizados. Os sinais analógicos disponibilizados por uma câmera (colorida) seguem algum formato padrão. Existem vários formatos padronizados para sinal de vídeo analógico: o sistema NTSC (*National Television Standards Committee*), o sistema PAL (*Phase Alternation by Line*) e o SECAM (*Sequential Couleur à Mémoire*). Cada qual possui suas especificações e seu território de abrangência [Lut91].

No Brasil adota-se o sistema de transmissão a cores PAL padrão M. O padrão M especifica a transmissão de 30 quadros por segundo, cada quadro possuindo 525 linhas e em uma frequência de 4.2 MHz (*MegaHertz*), o que implica em uma taxa mínima de amostragem para digitalização de 8.4 MHz. Na literatura é comum encontrar referências ao “padrão PAL-M”. Deve-se porém notar que o PAL é o sistema de transmissão de cor, o padrão é o M.

A utilização direta de PCM linear para a digitalização de sinais de vídeo gera uma quantidade muito grande de dados devido às altas taxas de amostragem necessárias. Por exemplo, a uma taxa de amostragem de 8.4 MHz com digitalização em 8 *bits* por amostra, teríamos 8.4 MBps (*MegaBytes* por segundo) de vídeo, exigindo uma taxa de transmissão de 67.2 Mbps. Por essa razão utilizam-se técnicas de compressão (que serão vistas mais adiante neste capítulo) simultaneamente ao processo de digitalização [Taw93].

1.4 - TÉCNICAS DE COMPRESSÃO E CODIFICAÇÃO

Existem três razões principais para que os dados em sistemas multimídia sejam comprimidos: necessidade de grande espaço de armazenamento, dispositivos de armazenamento lentos que impossibilitam disponibilizar os dados (por exemplo, vídeo) em tempo real e meios de transmissão lentos. Por exemplo, seja um vídeo (com o áudio associado) de 15.4 segundos, com os seguintes atributos: áudio capturado a 22.05 kHz, canal mono e 8 bits por amostra; vídeo com tamanho de quadro (*frame*) 320x240 *pixels*, capturado a 25 fps e utilizando 24 bits para cores. O arquivo gerado sem compactação com essa configuração chega a 84.9 MB, fazendo as contas: $320 \times 240 \times 25 \times 3 \text{ (24bits = 3Bytes)} \times 15.4 = 88.704.000 \text{ Bytes}$, $22.050 \times 1 \text{ (canal mono)} \times 8 \times 15.4 = 339.570 \text{ Bytes}$, $88.704.000 + 339.570 = 89.043.570 \text{ Bytes} = 84.9 \text{ MB}$. Note-se que são apenas 15.4 segundos de vídeo. Utilizando compactação o tamanho do arquivo cai para 3MB, o que é mais condizente com as atuais tecnologias de armazenamento.

A seguir nesta seção, serão apresentadas as classificações das técnicas de compressão e então, algumas das técnicas mais conhecidas para a compressão de áudio e para a compressão de vídeo serão discutidas.

1.4.1 - CLASSIFICAÇÃO DAS TÉCNICAS DE COMPRESSÃO

Segundo Furht [Fur95], as técnicas de compressão podem ser classificadas em técnicas *lossless* e técnicas *lossy* (figura 1.8). As técnicas *lossless* recuperam perfeitamente a representação original. As técnicas *lossy* recuperam uma representação próxima à original e são mais utilizadas em compressão de imagens gráficas e vídeos por apresentarem uma alta taxa de compressão para esse tipo de mídia.

Padrões para multimídia digital têm sido desenvolvidos baseados em técnicas híbridas, como ilustra a figura 1.8.

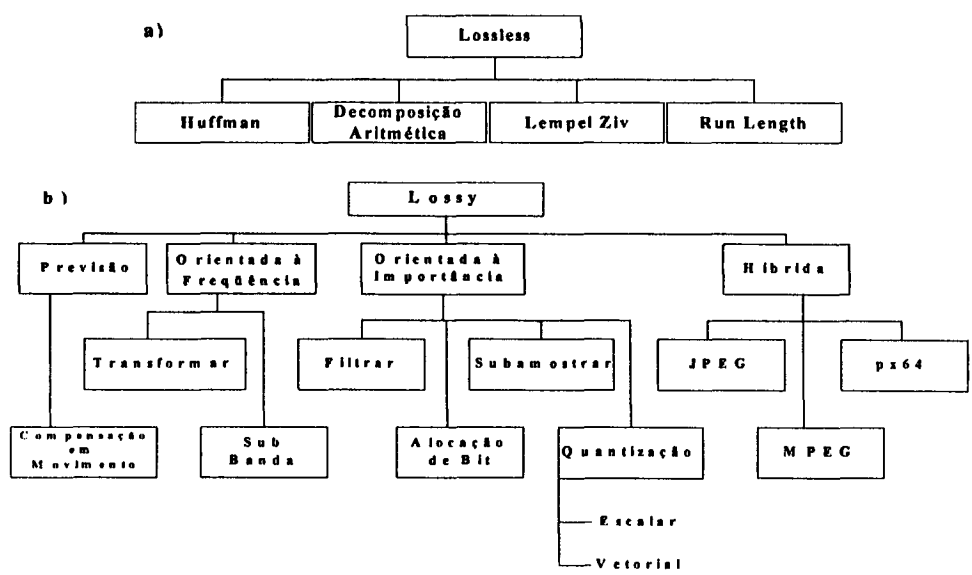


Figura 1.8 - Classificação para Técnicas de Compressão (a) *Lossless* e (b) *Lossy* [Sis96].

1.4.2 - TÉCNICAS DE COMPRESSÃO PARA ÁUDIO DIGITAL

Da mesma forma que vídeo digital, áudio digital não comprimido requer uma grande quantidade de largura de banda para ser transmitido. Dentre as várias técnicas existentes para comprimir áudio digital, algumas das mais importantes (ou mais utilizadas) são descritas a seguir. Algumas dessas técnicas utilizam qualquer tipo de sinal de áudio, outras apenas sinais de voz, mas tipicamente todas essas técnicas podem realizar compressão e descompressão em tempo real via *software* ou *hardware* barato.

1.4.2.1 - MU-LAW E A-LAW PCM

Com métodos de codificação PCM (*Pulse Code Modulation*), cada amostra é representada por uma palavra código (*word-code*). PCM uniforme usa um quantizador uniforme de espaçamento. Executando uma transformação, o quantizador uniforme de espaçamento pode ser alterado para logarítmico (PCM logarítmico), permitindo que um intervalo maior de valores possa ser abrangido com o mesmo número de *bits* (8 *bits* por amostra). Com 8 *bits* por amostra, utilizando PCM logarítmico, pode-se representar intervalos maiores de valores, equivalentes ao requerido por 14 *bits* para PCM linear [Pan93]. Isso ocorre através de uma razão de compressão de 1.75:1 (quantia original de informação: quantia comprimida de informação). As técnicas de compressão de áudio mu-law e A-law utilizam esse tipo de transformação.

Os métodos de codificação PCM mu-law e A-law são formalmente especificados na *International Telecommunications Union* (ITU) - Setor de Padronização de Telecomunicações (ITU-T)-, recomendação G.711, “modulação por código de pulso ou *Pulse Code Modulation* (PCM) de frequência de voz”. O formato de codificação PCM mu-law é comumente utilizado na América do Norte e no Japão, em telefonia digital com rede digital de serviço integrado (*Integrated Services Digital Network* - ISDN). O formato de codificação PCM A-law é comumente utilizado com ISDN em outros países. G.711 é um padrão de áudio especificado em H.320 [Ret95].

1.4.2.2 - ADAPTIVE DIFFERENTIAL PULSE CODE MODULATION (ADPCM)

O método PCM codifica as amostras de áudio independentemente de outras amostras. Porém, as amostras adjacentes são similares, de forma que os valores de uma amostra podem ser prognosticados com a mesma precisão usando o valor da amostra adjacente. O ADPCM calcula as diferenças (utilizando 4 bits) entre as amostras e seu valor prognosticado e, codifica a diferença (por isso o termo *Differential*). O termo *Adaptive* vem do fato de que codificadores podem adaptar sinais característicos pela troca de parâmetros de previsão e quantização. Tipicamente, ADPCM obtém razões de compressão de 2:1 quando comparado com PCM mu-law ou A-law.

Muitos sistemas de videoconferência utilizam o método de codificação ADPCM. A ITU-T possui várias recomendações definindo diferentes métodos ADPCM (G.721, G.722, G.723, G.726, G.727). Um dos métodos de codificação de áudio especificado pelo H.320 ([Ret95, Flu95]) é o G.722, “7 kHz de codificação de áudio dentro de 64 Kbps”, que usa codificação SB-ADPCM (*Sub-Band ADPCM*). O G.722 codifica amostras a uma taxa de 16 kHz com 14 *bits* de precisão. Com o método SB-ADPCM, a frequência de banda é dividida em duas sub-bandas (alta e baixa) e os sinais em cada sub-banda são codificados usando ADPCM. O G.722 tem três modos de operação: 64, 56 e 48 Kbps. Com o modo 56 ou 48 Kbps, o adicional 8 ou 16 Kbps de largura de banda (assumindo um canal de comunicação de 64 Kbps) pode ser usado para outros dados.

1.4.2.3 - G.721 E G.723

A recomendação G.721 da ITU converte um fluxo de *bit* de 64 Kbps para um fluxo de 32 Kbps. O G.721 é baseado na técnica ADPCM. Cada diferença de valor é codificada em 4 *bits*. Como em G.711, a taxa de amostragem é de 8 kHz [Flu95]. O G.723 é outro padrão de compressão baseado em ADPCM *lossy* que opera a 24 Kbps. A qualidade do som resultante é inferior à descompressão do padrão PCM G.711 ou do padrão G.722 baseado em SB-ADPCM [Flu95].

1.4.2.4 - LINEAR PREDICTIVE CODING (LPC) E CODE EXCITED LINEAR PREDICTION (CELP)

Esses dois métodos são especialmente destinados à codificação de sinais de voz. Eles utilizam os modelos característicos dos sinais de voz para obter bons resultados, contudo não trabalham bem com outros sinais de áudio.

O codificador LPC usa um modelo analítico simples do sistema vocal e os melhores parâmetros são transmitidos e usados pelo decodificador para gerar voz artificial, similar à original. LPC é usado para comprimir áudio a 16 Kbps e abaixo disso. A codificação GSM (*Groupe Speciale Mobite*) usa uma variação do LPC chamado RPE-LPC (*Regular Pulse Excited LPC*). O GSM teve início como padrão de codificação de voz para telefone celular na Europa. O

GSM comprime 160 amostras de 13 *bits* (2080 *bits*) para 260 *bits* com uma razão de compressão de 8:1.

O codificador CELP realiza uma modelagem do sistema vocal idêntico ao codificador LPC. Ele calcula o erro entre a voz de entrada e o modelo, transmitindo o parâmetro do modelo e o valor do erro. Os erros são índices de um livro código compartilhado entre codificadores e decodificadores.

A recomendação ITU-T G.728, que é um dos formatos de codificação de áudio especificado pela H.320 (discutido por Rettinger [Ret95]), usa uma variação do CELP, o LD-CELP (*Low Delay CELP*). O G.728 requer uma largura de banda de 16 Kbps e é computacionalmente complexo, requerendo *hardware* especial.

1.4.2.5 - MPEG

MPEG (*Motion Pictures Experts Group*) é um padrão ISO (*International Standardization Organization*) para compressão de áudio com alta qualidade (o MPEG também é utilizado para compressão de vídeo, como veremos na seção 2.4.3.5). Seu algoritmo é complexo, formado por um conjunto de três sistemas de codificação e decodificação, chamados de camada 1 (menos complexa), camada 2 e camada 3 (mais complexa). Essas três camadas são hierarquicamente compatíveis, permitindo que o decodificador da camada 3 trabalhe com fluxos comprimidos pelas camadas 1 e 2, por exemplo.

- **Camada 1:** versão simplificada do MUSICAM. Esta permite a construção de simples codificadores e decodificadores com um desempenho médio e que operam razoavelmente bem a 192 ou 256 Kbps por canal.

- **Camada 2:** idêntica ao padrão MUSICAM. É otimizada para uma taxa de *bits* de 96 ou 128 Kbps por canal monofônico. No modo estereofônico, esse sistema MPEG equivale à qualidade de CD.

- **Camada 3:** sistema que possui maior desempenho na série. É uma combinação dos esquemas MUSICAM e ASPEC, um esquema de compressão de som projetado em Erlangen, Alemanha. Objetiva uma taxa de *bits* de 64 Kbps por canal de áudio. Nessa velocidade, a

qualidade está muito próxima do CD e é definitivamente melhor que a camada 2 operando a 64 Kbps [Flu95].

As três camadas utilizam o mesmo princípio para realizar compressão: uma combinação de *transform encoding* e *sub-band division* [Flu95, Pan93]. Dessa forma temos o seguinte:

- o *spectrum* é dividido em 32 sub-bandas;
- uma transformação de Fourier rápida - que é uma técnica de *transform encoding* - é aplicada para requisitar o sinal no domínio frequência;
- um modelo psico-acústico é aplicado ao sinal transformado para avaliar justamente o nível de ruído perceptível, ou seja, o decodificador psico-acústico realiza uma busca no sinal, marcando os pontos de ruídos. Depois de marcados os pontos, o decodificador analisa o melhor sinal de áudio que poderá ser representado, considerando apenas os pontos não marcados.

A família de padrões de áudio MPEG é destinada a operar a uma taxa de *bits* que varia de 32 a 448 Kbps por canal monofônico. A taxa de amostragem pode ser 32, 44.1 ou 48 kHz. O MPEG-II adiciona três outras frequências, 16 kHz, 22.05 e 24 kHz. O MPEG-I permite dois canais de áudio, enquanto que o MPEG-II permite mais cinco canais de áudio adicionais de baixa frequência [Cro95].

1.4.3 - TÉCNICAS DE COMPRESSÃO PARA VÍDEO DIGITAL

Agora serão discutidas algumas técnicas importantes para codificação e compressão de vídeo. Algumas medidas importantes de compressão são a proporção de compressão e o número de *bits* necessários para representar um *pixel* de uma imagem (*bits* por *pixel*).

1.4.3.1 - JPEG E MJPEG

JPEG (*Joint Photographic Experts Group*) é um padrão de codificação para imagens não noventes. Contudo, empregando-se uma abordagem conhecida com *Motion JPEG* (MJPEG), que utiliza um *hardware* especializado, é possível codificar e decodificar uma série de imagens JPEG em tempo real realizando assim vídeo em movimento [Ret95]. Apesar de utilizar codificação JPEG, não existe oficialmente o padrão MJPEG.

Os modos de operação para o JPEG [Cro95] são: Seqüencial, Progressivo, *Lossless* e Hierárquico.

- **Seqüencial**: esse modo é muito simples, contudo é muito implementado em *hardware* e *software*. Aqui, as imagens são codificadas em uma simples leitura da esquerda para a direita e de cima para baixo.

- **Progressivo**: diferente do modo seqüencial, o modo progressivo codifica as imagens em múltiplas leituras. Isto é útil em aplicações cujo tempo de transmissão é longo e o usuário prefere ver a imagem sendo construída passo a passo e com uma qualidade inferior.

- **Lossless**: esse modo de codificação garante que a imagem fonte será exatamente recuperada. Isto é importante para aplicações que não permitem qualquer perda na imagem, como algumas aplicações médicas.

- **Hierárquico**: aqui a imagem é codificada em múltiplas resoluções. Quando há transmissão sobre redes de comutação de pacotes, somente os dados importantes para uma dada resolução (determinada pela aplicação) podem ser transmitidos, permitindo que várias aplicações compartilhem os mesmos recursos de rede. Nas transmissões em tempo real, uma rede congestionada pode perder pacotes contendo dados de alta resolução, resultando em uma qualidade degradada da imagem em lugar de atraso.

JPEG utiliza uma técnica de compressão espacial *intraframe* e codificação com transformação Co-seno Discreta (DCT). Esse tipo de técnica é usado por outros métodos de codificação, como o H.261 (discutido na seção 2.4.3.2). Os passos básicos da codificação baseada em DCT são mostrados na figura 1.9. Os processos de codificação e decodificação estão detalhados por Rettinger [Ret95]. Codificação JPEG obtém compressão na ordem de 10:1 até 20:1 [Ret95].

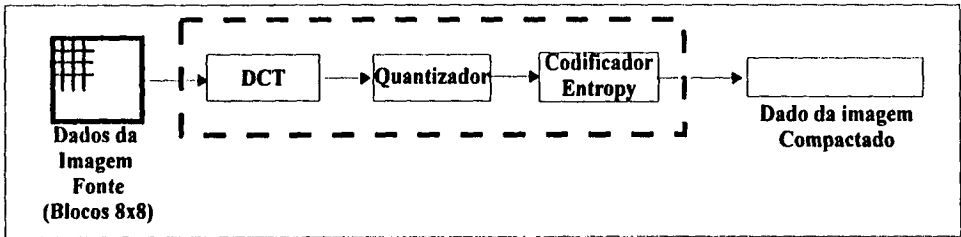


Figura 1.9 - Codificação em DCT [Sis96].

1.4.3.2 - H.261

O H.261 é primeiramente destinado à videoconferência sobre redes ISDN e é especificado por H.320 ([Ret95]). Esse padrão de compressão de vídeo visa (recomendação ITU-T H.261) larguras de banda de comunicação, entre 64 Kbps (*Kilobits* por segundo) e 2 Mbps (*Megabits* por segundo), medidas em intervalos de 64 Kbps. Essa técnica é também conhecida como “*px64*” onde “*p*” varia de 1 até 30.

O padrão H.261 utiliza codificação temporal *interframe* e codificação espacial *intraframe*. No modo de codificação *intraframe* é utilizada compressão espacial baseada em DCT. No modo de codificação *interframe*, um processo de **compensação em movimento**² é executado para calcular as diferenças entre os *frames*, que são então codificados em DCT.

1.4.3.3 - CELL

Existem dois tipos de compressão por células: CellA e CellB. CellA é uma técnica assimétrica onde se utiliza mais computação para compressão do que para descompressão. CellB é mais simétrica computacionalmente sendo adequada para ser usada em tempo real, como por exemplo, em videoconferência. CellA utiliza RGB como entrada, enquanto que a CellB utiliza YUV.

A técnica de codificação Cell baseia-se no método chamado codificação por truncamento de bloco (*Block Truncation Coding* - BTC). A imagem de entrada é dividida em blocos de 4x4 de *pixels* chamados de células. Cada célula é codificada por uma máscara de 16 *bits* e duas intensidades de 8 *bits*. A diferença entre CellA e CellB está no sentido em que as intensidades são escolhidas. CellA cataloga os valores através de um mapa de cores enquanto CellB utiliza uma tabela de quantização vetorial [Ret95].

Compensação em Movimento - Em compressão de vídeo, é o processo de determinar que parte da figura será movida de um *frame* para outro. Somente as partes que foram movidas serão incluídas no fluxo de dados.

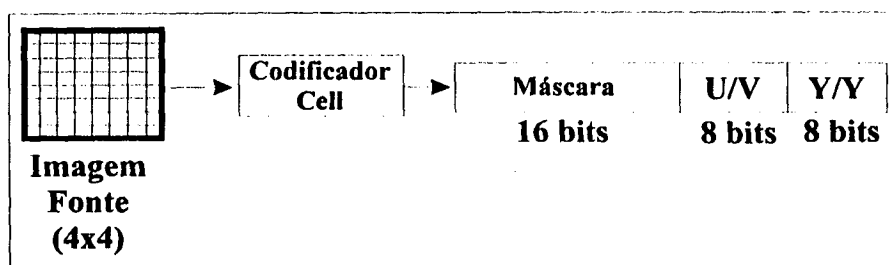


Figura 1.10 - Codificação da CellB [Ret95].

1.4.3.4 - CU-SEEME

A universidade de Cornell desenvolveu o CU-SeeMe como uma ferramenta de videoconferência para Internet. Ela também utiliza compressão temporal (*interframe*) e espacial (*intraframe*), representando a entrada de vídeo em 16 tonalidades de cinza, utilizando 4 *bits* por *pixel*. Cada imagem é dividida em blocos de 8x8 *pixels* para análise e os novos *frames* são comparados com os anteriores. No caso de um bloco ter sido muito alterado, ele é retransmitido.

O algoritmo de compressão utilizado é do tipo *lossless* e foi desenvolvido em Cornell. O CU-SeeMe foi projetado para operar em computadores *Macintosh* e o tamanho da compressão está acima de 60% da original [Ret95].

1.4.3.5 - MPEG

Moving Pictures Experts Group (MPEG) é o nome do comitê ISO que trabalha com compressão de áudio digital e vídeo (colorido) digital. A especificação MPEG é desenvolvida especialmente para permitir a transmissão de imagens padrão *broadcast* a uma taxa de dados que corresponda à velocidade de recuperação de dados de um CD-ROM e de fitas DAT (entre 1 e 1.5 Mbps) [Mat94, Gad92].

Esse padrão divide-se em: Codificação de Vídeo, Codificação de Áudio e Sistemas. A subdivisão de Sistemas inclui informações sobre a sincronização dos fluxos de áudio e vídeo. O fluxo de vídeo utiliza aproximadamente 1.15 Mbps, e o restante da largura de banda é usado pelos fluxos de áudio e dados do sistema [Gad92].

As imagens em MPEG são formadas por três matrizes retangulares de inteiros: uma de luminância (Y) e duas de cromaticidade (Cb e Cr), dando origem a três formatos. De acordo com

esses formatos as imagens classificam-se em quatro tipos: *I-frame*, *P-frame*, *B-frame* e *D-frame*, que são detalhados por Crowcroft [Cro95].

As especificações do MPEG têm sido aproximadas pelo grupo ISO em algumas fases. Essas fases não ocorrem rigorosamente em uma ordem consecutiva, havendo sobreposição entre algumas delas. As fases são [Flu95]:

- MPEG-1 destina-se à armazenagem de seqüências de áudio sincronizado e vídeo colorido, com qualidade VCR sobre CD-ROMs. Essa opção funciona a 1.2 Mbps com imagens *SIF*. Na prática, se o canal de áudio requerer 200 a 250 Kbps para qualidade de CD, a taxa de *bit* do canal de vídeo permanecerá entre 1.15 a 1.2 Mbps. A razão de compressão de vídeo é na ordem de 26:1;

- MPEG-2 destina-se a TV com qualidade de estúdio e canais de áudio com qualidade de CD com taxa de 4 a 6 Mbps. Pode manipular resoluções que chegam a 720x480 *pixels* para sinal de luminância e 360x480 *pixels* para o espaço de cor.

- MPEG-3 destina-se a codificação e compressão de HDTV (*High-Definition Television*). Contudo, com um resultado rápido e positivo, o MPEG-2 incluiu o padrão HDTV, de forma que o MPEG-3 foi cancelado;

- MPEG-4 destina-se a ser um padrão extremamente poderoso para videoconferência a uma taxa extremamente baixa, que varia de 4.8 a 64 Kbps. Seu campo de aplicação será o videofone e a videoconferência. A tabela 1.1 mostra um resumo das técnicas JPEG, H.261 e MPEG.

Tabela 1.1 - Padrões de Compressão Multimídia [Sis96].

Codiname	Designação	Grupo de Padrões	Razão de Compressão
JPEG	Compressão Digital e codificação de imagens paradas com tonalidade contínua	Grupo especialista em fotografias	15:1 (aplicações com quadros parados cheios)
H.261, px64	Codificador/Decodificador para serviços audiovisuais no px64 Kbps (<i>Kilobits</i> por segundo)	Grupo especialista em codificação para transmissão visual sobre linhas telefônicas.	100:1 até 2000:1 (telecomunicações baseadas em vídeo)
MPEG	Codificação de figuras em movimento e áudio associado	Grupo de especialista em figuras em movimento	200:1 aplicações intensivas em movimento

2.1 - INTRODUÇÃO

Os sistemas hipertexto clássicos desenvolvidos nos anos 80 evoluíram de modo independente, refletindo essa independência nos hiperdocumentos por eles manipulados. Como consequência, os hiperdocumentos gerados por um sistema não podem ser utilizados em outro sistema ou plataforma. Desse modo, padrões como o *HyTime* (*Hypermedia/Time-Based Structuring Language*) [New91, ISO92] têm sido desenvolvidos com o objetivo de permitir que hiperdocumentos possam ser descritos em termos de sua estrutura e conteúdo e assim, possam ser utilizados por quaisquer sistemas que utilizem o padrão.

A expansão da Internet e, conseqüentemente, da comunidade de usuários da *World Wide Web* (WWW), tem provocado um crescente interesse por aplicações hipermídia. Com isso, aumentou a importância dos padrões para intercâmbio e referência entre documentos hipermídia.

Este capítulo discute alguns padrões para estruturação e outros para intercâmbio e apresentação de documentos hipermídia. São eles, pela ordem: SGML, HTML e *HyTime* para estruturação; MHEG, *QuickTime*, AVI e *QuickTime* para *Windows* para intercâmbio e apresentação.

2.2 - SGML

O padrão SGML (*Standard Generalized Markup Language*) [ISO86] foi proposto para permitir a definição de documentos de acordo com sua estrutura e conteúdo, independentemente de sua apresentação [Bro89]. SGML é uma linguagem genérica para a descrição da estrutura lógica de documentos, permitindo a definição de linguagens específicas, sendo que essas últimas afinam a estrutura hierárquica de um tipo de documento. SGML tem flexibilidade para definir um conjunto ilimitado de linguagens específicas, por exemplo, uma para memorandos, uma para

artigos, uma para livros, etc. [Her94]. Portanto, cada documento SGML está associado a uma linguagem específica que o define. Essa definição é feita através da criação de um DTD (*Document Type Definition*). Desse modo, a linguagem definida em um DTD especifica uma classe de documentos, por exemplo, livro, jornal, revista, a qual é compartilhada por muitas instâncias de documentos.

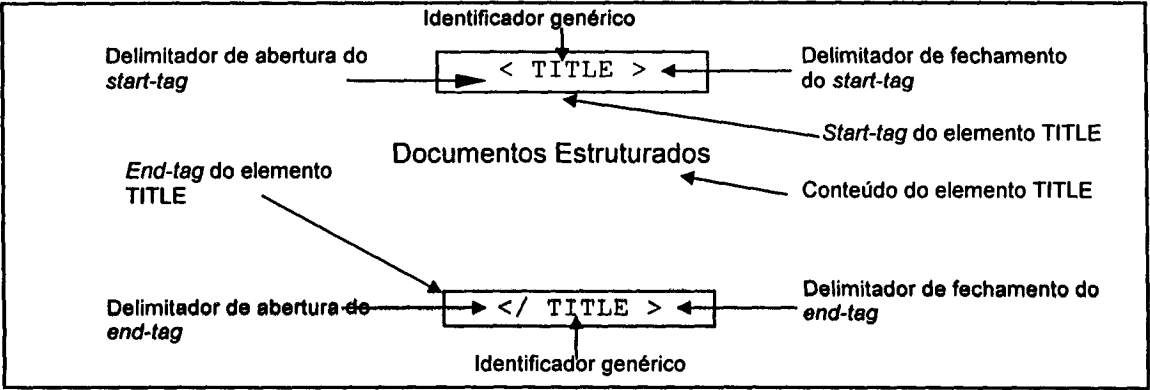


Figura 2.1 - Anatomia de um elemento [Her94].

O DTD define a estrutura lógica de um documento, que é descrita através de marcas padronizadas (*markup*). Essas marcas identificam o início e o fim de cada item lógico, chamado elemento SGML (figura 2.1), e permitem a especificação de seus atributos. É possível atribuir um identificador único a um elemento SGML, de forma que ele possa ser referenciado em qualquer ponto de um documento. Essa característica tem sido utilizada para a especificação de ligações hipertexto (*links*).

Depois que um documento SGML foi editado (figura 2.2), deve-se assegurar que sua marcação seja consistente, esteja livre de erros, e seja interpretada corretamente. Todo sistema que utiliza o padrão SGML deve conter um programa que reconhece a marcação em documentos SGML. Esse programa é chamado de SGML *parser*. O SGML *parser* checa:

- se o DTD de um documento segue o padrão SGML;
- se a instância do documento está de acordo com o DTD.

Parsers estão freqüentemente incluídos em um editor SGML, mas eles também podem ser independentes. *Parsers* ajudam a prevenir erros e o mau uso de marcação.

```

<CHAP>
  <TITLE>Estruturação de Documento Hiperídia</TITLE>
  <SECT>
    <SUBT>ODA</SUBT>
    <PAR>O padrão ODA (Office Document Architecture)...
    ... </PAR>
    <PAR> ... </PAR>
    ...
  </SECT>
</CHAP>

```

Figura 2.2 - Exemplo de um trecho de um documento SGML [Tri96].

2.3 - HTML

O padrão HTML (*Hypertext Markup Language*) [Ber94], amplamente utilizado como formato para hiperdocumentos na WWW, é formalmente definido por um DTD SGML e, desse modo, define como um conjunto de elementos deve ser estruturado para compor um documento. HTML define, entre seus elementos, *links* hipertexto e inclusão de outras mídias, e seu formato é reconhecido pela maioria dos *browsers* existentes.

A maioria dos documentos existentes hoje na WWW não segue totalmente a definição DTD para HTML. Isso ocorre porque os *browsers* atualmente em uso na WWW processam documentos HTML com flexibilidade e sem utilizar todas as características de validação oferecidas por um SGML *parser*. Isso significa que os *browsers* em uso na WWW possuem, embutidos, um *parser* encarregado de reconhecer elementos HTML, *parser* este que processa uma estrutura menos rígida que a definida pelo DTD HTML.

HTML não usa os identificadores únicos SGML para referenciar porções do documento. Ele utiliza a notação URL (*Universal Resource Locator*) para especificar a posição de documentos na WWW. Isto provê um modo comum e unificado de acesso a arquivos e informações através da *web* [Rut95]. Porém, HTML não satisfaz todos os requisitos necessários para atender à nova geração de documentos hiperídia integrados e abertos. O SGML e o *HyTime* (seção 2.4) satisfazem muitos desses requisitos. Por isso, é interessante que os documentos HTML sejam adaptados para *HyTime* [Tri96].

2.4 - HYTIME

O padrão *HyTime* (*Hypermedia/Time-Based Structuring Language*) [ISO92], que pode ser visto como uma extensão ou uma aplicação de SGML, é definido como um DTD e pode ser

usado para descrever a estrutura de documentos multimídia e hipermídia. Segundo [Pim95], *HyTime* é uma metalinguagem para representação de ligações hipertexto (*hyperlinks*), bem como para o escalonamento e a sincronização de eventos no espaço e no tempo, fornecendo mecanismos e modelos para identificação, localização e ligação desses objetos de informação.

O *HyTime* é definido formalmente por um conjunto de regras, chamadas **formas arquiteturais** (AFs), que os projetistas de aplicações podem usar em seus DTDs. Cada forma arquitetural especifica como um elemento SGML pode ser definido em um DTD cujas instâncias contêm informações com relação a certos conceitos ou grupos de conceitos *HyTime* [Rut93]. Ele é estruturado como um conjunto de **módulos** que são adicionados a uma aplicação conforme o necessário. Os módulos *HyTime* são: *Base Module*, *Measurement Module*, *Location Address Module*, *Hyperlinks Module*, *Scheduling Module* e *Rendition Module* [Fuj95, Rut93].

Duas classes definidas no padrão - *Minimal Hyperlinking* e *Minimal Scheduling* - fornecem suporte, respectivamente, para a especificação da estrutura necessária para hiperdocumentos e a especificação de elementos multimídia [Pim96]. A classe *Minimal Hyperlinking*, baseada no *Hyperlinks Module* e no *Location Address Module*, permite a definição de hiperdocumentos de estrutura elaborada. A classe *Minimal Scheduling*, baseada no *Scheduling Module* e no *Measurement Module*, através de suas AFs, permitem a formalização da estrutura de documentos multimídia.

Assim como o SGML possui o *SGML parser*, o *HyTime* possui o *HyTime engine*. Um *HyTime engine* interage com o *SGML parser*, reconhece as formas arquiteturais *HyTime* e executa o processamento específico do *HyTime*. Esse processamento inclui a verificação de erros, a extração de atributos específicos de *HyTime* e a resolução de certos tipos de endereços. A informação gerada pelo *HyTime engine* é colocada à disposição da aplicação. O papel principal do *HyTime engine* é dar suporte à composição de informações hipermídia, inclusive resolução de *links*, endereçamento de objetos e a representação de sistemas de coordenadas [Buf94].

2.5 - MHEG

MHEG (*Multimedia and Hypermedia information coding Expert Groups*) é um modelo orientado a objetos para intercâmbio multimídia e hipermídia definido pela ISO. O padrão

MHEG está relacionado com a composição de objetos de mídia baseados no tempo. A codificação desses objetos de mídia envolvidos na composição é definida por padrões como o ASN.1, por exemplo. O MHEG é uma forma de representação de mídia composta que suporta interatividade e despacho em tempo real através de uma rede de computadores. Existem vários padrões MHEG já definidos sendo o mais recente o MHEG-6. Este define a interface e o ambiente de execução de uma “*virtual machine script engine*” (implementada em linguagem JAVA) que executa no topo de um MHEG-5. Nosso interesse aqui é o suporte à multimídia fornecido pelo padrão MHEG-5 [Jos95].

O MHEG-5 foi desenvolvido para suportar a distribuição de aplicações multimídia em uma arquitetura cliente-servidor composta por plataformas de diferentes tipos, definindo uma representação final para intercâmbio entre as aplicações. As aplicações consistem basicamente de código declarativo, mas também é possível executar códigos procedurais [Jos95]. Aplicações MHEG-5 precisam ser criadas (*authored*) apenas uma vez e podem executar em qualquer plataforma que suporte o MHEG-5.

A principal função do MHEG-5 é definir a sintática e a semântica de um conjunto de classes de objetos que podem ser usados para a interoperabilidade de aplicações multimídia entre plataformas de recursos mínimos. As aplicações desenvolvidas devem ser armazenadas em um servidor e, a medida que porções da aplicação são requeridas, elas devem ser fornecidas ao cliente. É responsabilidade do cliente ter uma rotina que interprete as partes da aplicação, apresente a aplicação para o usuário e gerencie a interação local com o usuário [Jos95].

Uma aplicação MHEG-5 é feita de cenas e objetos que são comuns a todas as cenas. Uma cena contém um grupo de objetos, chamados ingredientes, que representam informações (imagens, sons, vídeos, etc.) juntamente com procedimentos localizados que são disparados por eventos (apertar um botão do *mouse* ativa um som, por exemplo). Pelo menos uma cena está ativa em um dado momento. A navegação em uma aplicação se dá através da transição entre cenas.

Um MHEG *engine* tem a habilidade de mostrar um objeto de vídeo em um sistemas de coordenadas retangulares com um tamanho fixo além de “tocar” objetos de áudio. Um dispositivo de entrada qualquer (*mouse*, por exemplo) pode ser usado para interagir com a aplicação através da MHEG *engine* [Jos95].

2.6 - QUICKTIME

QuickTime é uma extensão multimídia para o sistema operacional System 7 da Apple para computadores Macintosh. Ele provê capacidade para capturar, armazenar, controlar, sincronizar e exibir dados de mídia contínua. Vídeo digitalizado é introduzido como tipo de dado padrão do sistema e admite um manuseio facilitado de outras mídias contínuas como áudio e animação.

O tipo de dado básico do *QuickTime* é uma **Apresentação (movie)**. Todos os tipos de dados de mídia contínua são armazenados em documentos do tipo *movie*. Informações de tempo (data de criação e de última modificação, duração, etc.) relacionadas à apresentação, são também armazenadas no documento *movie* [App96].

O *QuickTime* utiliza um modelo de **Trilha** para armazenar temporariamente dados relacionados a uma apresentação (*movie*) (figura 2.4). Um *movie* pode conter uma ou mais trilhas. Uma trilha é uma seqüência de tempo ordenada, de um certo tipo de mídia. As mídias são endereçadas utilizando uma lista editada, contendo os pontos finais de *clips* de mídia digital ou segmentos (figura 2.4) [App96].

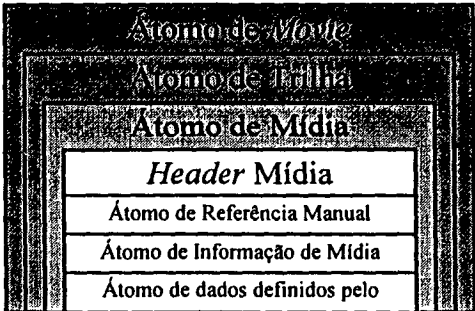


Figura 2.3 - Modelo abstrato de Átomo.

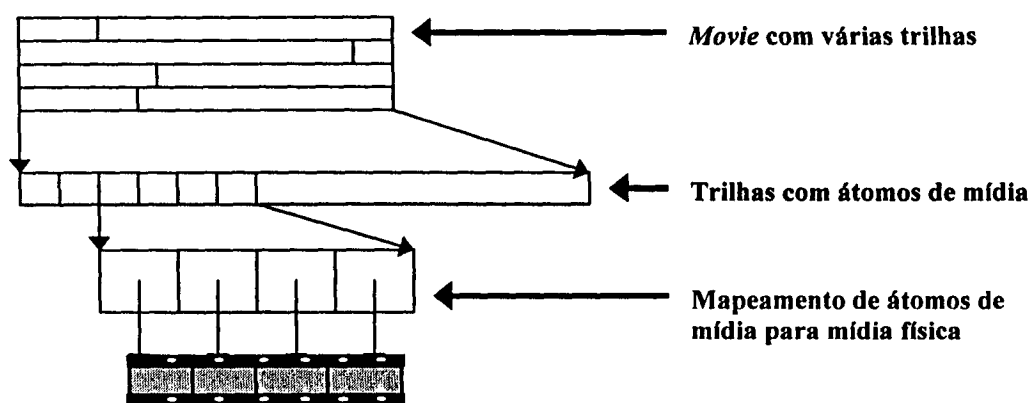


Figura 2.4 - Componentes em *QuickTime*.

A trilha representa um *stream* de apresentação (dados de vídeo ou de áudio) que flui em paralelo com todas as outras trilhas. Com cada trilha são armazenadas informações como criação e modificação de dados, duração, número da trilha, características especiais, lista de trilhas relacionadas, volume e tempo de *start*, padrão de *playback* e uma referência de dados para cada segmento de mídia. Um segmento de mídia é um conjunto de referências para dados de áudio e vídeo, incluindo informações de tempo, linguagem, *display* ou qualidade de som, tipo de dados de mídia e ponteiro de dados.

Todas as trilhas podem ser vistas e/ou ouvidas concorrentemente. As trilhas de uma apresentação são sempre sincronizadas. Visto que *movies* são documentos, eles podem também ser editados, em vez de apenas apresentados. Operações como *cut*, *copy* e *paste* são possíveis.

A arquitetura do *QuickTime* contém três componentes principais:

Movie Toolbox: oferece um conjunto de serviços que permite ao usuário incorporar *movies* nas aplicações. Essas aplicações podem manipular características de dados de áudio e vídeo. Dados *movie* podem ser incorporados e exportados com o sistema *Clipboard* e também podem ser editados dentro desse componente.

Image Compression Manager: provê uma interface comum para compressão e descompressão de dados, independente da implementação de *Hard Disk*, CD-ROM e *Floppy*. Ele oferece um serviço de diretório para selecionar o componente de compressão correto. Diferentes níveis de interface para as necessidades de diferentes aplicações estão disponíveis. As técnicas de compressão são: um esquema proprietário de compressão de imagem, uma

implementação JPEG e um compressor proprietário de dados de vídeo digitalizado. Um compressor de animação pode comprimir dados digitais nos modos *lossy* e *lossless*. Um compressor gráfico também está disponível.

- **Component Manager:** provê um serviço de diretório relacionado aos componentes. É a interface entre a aplicação e os vários componentes do sistema. Protege os reveladores (fotográficos) da tarefa de negociar detalhes de interface com um *hardware* específico. Em um *Component Manager* são aplicados conceitos de orientação a objetos como: herança, classes, hierarquia, e assim por diante. Os componentes gerenciados por esse módulo são: *clock*, compressor e descompressor de imagem, controlador de *movie*, controlador de seqüência, canais de captura de seqüência e digitalizador de vídeo.

2.7 - QUICKTIME PARA WINDOWS

O *QuickTime* para *Windows* é uma versão do *software* (*QuickTime*) desenvolvido pela Apple Computers para executar em sua linha de máquinas. Toda a estrutura do *QuickTime* para *Windows* é similar em muitos aspectos ao *software* *Video for Windows* (seção 2.8). Existe um *driver* de interface de controle de mídia chamado MCIQTW.DRV, que é especificado na seção *mci*] do arquivo SYSTEM.INI do *Windows*. A terminologia na documentação do *QuickTime* para *Windows* SDK (*Software Development Kit*) é algo diferente da terminologia de documentação do *Video for Windows*, provavelmente refletindo a diferença de cultura entre a Microsoft e a Apple, bem como as diferenças na infra-estrutura do *software* [Mat94].

O *QuickTime* para *Windows* é construído sobre um objeto conceitual conhecido como *movie controller*. Vários aspectos importantes do *movie controller* são claramente explicados na documentação do *QuickTime* [Mat94]:

• *Movies* podem somente ser executados usando um *movie controller*.

Não é possível criar um componente *movie controller* personalizado.

O *movie controller* é funcionalmente idêntico ao usado no ambiente *QuickTime* em um computador Macintosh.

QuickTime segue fortemente o conceito de componente (*component*). Um componente é um objeto de *software* que age como uma extensão para o *QuickTime* para *Windows*. O

QuickTime provê um *framework* de *software* conhecido como *Component Manager*, o qual provê serviços para classificar componentes por função. Componentes são tipicamente implementados como DLLs (*Dynamic Link Library*). Um componente típico pode ser um *driver* de descompressão de vídeo. *QuickTime* para *Windows* não provê um *hardware* especializado para o gerenciamento de captura de vídeo em um PC [Mat94].

O *Component Manager* classifica os componentes de acordo com os tipos de serviços fornecidos e de acordo com os níveis de serviço. O tipo do componente é identificado por um código de quatro caracteres. Todos os componentes de um dado tipo requerem suporte ao mesmo conjunto de funções API (*Application Program Interface*) [Mat94].

Seqüências de vídeo sob *QuickTime* são chamadas de *movies*. Sob o *Windows*, um *movie QuickTime* é armazenado em um formato de arquivo padrão DOS com a extensão MOV. Para ser executado, um *movie* deve estar associado a um *movie controller*. *Movies QuickTime* podem ser executados sob o *Windows*, mas não podem ser editados ou gravados. O modelo de programação do *QuickTime* tem por objetivo isolar o programador de qualquer detalhe do formato atual dos dados dos *movies* armazenados. *QuickTime* define o tempo de duração da execução de um *movie* em termos de dois valores: **escala** e **taxa**. A escala define o número de intervalos por segundo usados para tomada de tempo. A taxa é um multiplicador definindo o quanto destas unidades de tempo devem passar por segundo. O *QuickTime* mantém um escalonador interno que reserva porções de tempo para um *movie* que está ativo. Um *movie* ativo simplesmente significa que está pronto para ser executado. Para ser executado, um *movie* deve estar associado a um *movie controller*, que é a interface suprida pelo *QuickTime* para controlar *movies* [Mat94]. Um *movie controller* tem elementos de interface como: botões de início e de parada (*start and stop*), avanço rápido, recuo rápido, barras de rolagem, e assim por diante.

O *QuickTime* para *Windows* provê um desempenho relativamente bom para a execução de vídeo e uma interface relativamente simples de operar. O *QuickTime* para *Windows* SDK é uma ferramenta suficiente para o desenvolvimento de aplicações incorporando um *movie player* ou um *driver* de descompressão de vídeo que utilize a interface *QuickTime* do usuário, mas não provê suporte para criação e edição completas de vídeo. É razoável assumir que o subconjunto *Windows* do *QuickTime* foi implementado, primariamente, para desenvolvedores Macintosh que desejavam obter portabilidade (Macintosh-PC) em suas aplicações [Mat94].

2.8 - AVI

Video for Windows (VfW) é a extensão oficial de vídeo movente para Microsoft *Windows*. Ele é um pacote de *software* que consiste de *drivers* com algoritmos instaláveis e um conjunto de ferramentas para captura e edição de áudio e vídeo.

VfW tem seu próprio formato de arquivo especial, chamado *audio-video interleaved* (AVI), que suporta áudio, vídeo, ou ambos juntos. Apesar de o computador teoricamente poder executar áudio e vídeo simultaneamente a partir de dois arquivos separados, o tempo de *seek* muito longo das unidades de armazenamento de massa torna essa abordagem impraticável. Muito provavelmente a execução do áudio ou do vídeo irá ser interrompida ou a sincronização irá ser perdida durante o *seek*. Esse problema é resolvido intercalando (*interleaving*) o áudio e o vídeo em um único arquivo (ou *stream* de dados), como mostra a figura 2.5. O arquivo está dividido em blocos seqüenciais, com cada bloco contendo dados de áudio e vídeo correspondendo ao tempo de duração do bloco durante sua execução [Lut95].

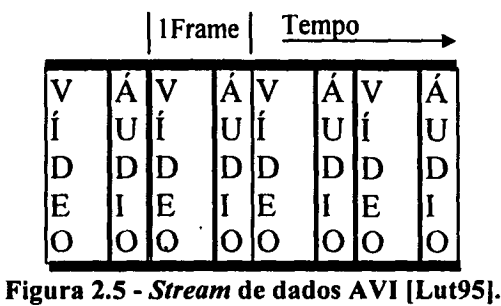


Figura 2.5 - *Stream* de dados AVI [Lut95].

Usualmente os blocos correspondem a um único *frame* de vídeo, sendo que cada bloco mantém um *frame* de vídeo com uma quantidade de áudio associada [Lut95].

Os serviços básicos de entrada e saída (E/S) de arquivos são similares aos usados em sistemas de arquivos em linguagem C. As aplicações devem abrir o arquivo para acessá-lo, realizar operações de E/S como leitura, escrita, etc., e fechar o arquivo quando terminar.

Operações de E/S bufferizadas reduzem o número de vezes em que os dispositivos de armazenamento físico, como disco rígido e CD-ROM, são acessados. Isto é importante porque durante as operações de E/S muito tempo é consumido esperando o dispositivo localizar e transferir as informações entre o dispositivo de armazenamento físico e a memória. Se uma aplicação está lendo ou escrevendo numerosos pequenos blocos de dados, o dispositivo de armazenamento pode gastar muito tempo realizando *seek* para uma localização física na mídia para cada operação de leitura ou escrita. Uma aplicação pode ganhar desempenho substancial utilizando serviços de E/S bufferizados [Sim96].

Serviços de E/S bufferizados utilizam um grande bloco de memória onde as aplicações lêem e escrevem dados. O dispositivo de armazenamento é somente acessado quando o *buffer* deve ser preenchido com novas informações durante operações de leitura e escrita. Isto ocorre quando o buffer é totalmente preenchido por operações de escrita ou quando os dados requisitados não estão dentro do *buffer* durante uma operação de leitura [Sim96].

Arquivos RIFF (*Resource Interchange File Format*) provêem um método geral para armazenamento e intercâmbio de dados multimídia. Aplicações podem acessar arquivos RIFF usando serviços de E/S bufferizados e não bufferizados. Arquivos RIFF utilizam um código de quatro caracteres para identificar elementos de um arquivo. Os arquivos RIFF consistem em uma série de elementos organizados hierarquicamente, ou *chunks*. Um *chunk* é uma unidade lógica de dados, tal como um *frame* AVI ou uma sequência de áudio. Os dois tipos de *chunks* básicos são RIFF e LIST [Sim96].

Em implementações recentes, arquivos AVI são armazenados como um arquivo RIFF (figura 2.6). Funções e macros AVI têm sido providas especialmente para o processamento de E/S de arquivos AVI. Isto elimina a necessidade de usar funções de E/S multimídia de baixo nível [Sim96].

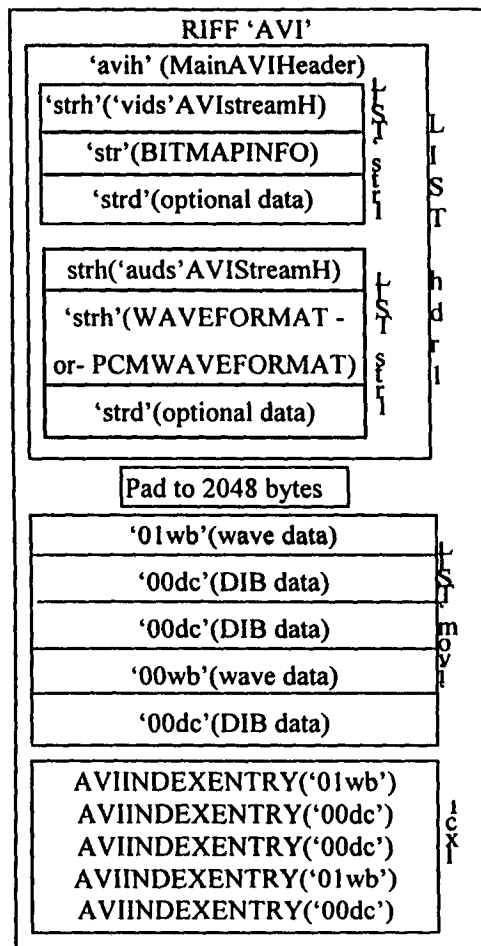


Figura 2.6 - Arquivo RIFF AVI [Sim96].

3

ASPECTOS PARA DESENVOLVIMENTO DE SISTEMAS HIPERMÍDIA DISTRIBUÍDOS

3.1 - INTRODUÇÃO

Sistemas hipermídia utilizam tanto mídias discretas (como texto e gráficos) quanto mídias contínuas (como áudio e vídeo). Os dados de mídia contínua (ou dados baseados no tempo) são gerados em uma determinada taxa e devem ser reproduzidos nessa mesma taxa, ou seja, existe um prazo (*deadline*) para que o dado seja exibido sem perda na qualidade [Buf94a]. Por exemplo, se um aplicação deseja exibir *frames* a uma taxa de 30fps e em algum momento da transmissão (imaginando que a exibição é remota) essa taxa cai, ocorrerão tremores na imagem (*jitter*) sendo exibida. As tecnologias atuais encontram algumas dificuldades em suportar essas características de tempo real dos dados de mídia contínua, de forma que muitas pesquisas tem sido feitas no sentido de superar essas dificuldades provendo um gerenciamento eficiente desses dados.

Este capítulo discute vários aspectos importantes para o desenvolvimento de sistemas hipermídia distribuídos, que são: qualidade de serviço, serviço de vídeo sob demanda, problemática dos sistemas operacionais convencionais em suportar multimídia, problemática das redes convencionais em suportar multimídia, armazenamento e acesso a dados multimídia e busca recuperação.

3.2 - QUALIDADE DE SERVIÇO

Servidores de dados multimídia devem suportar transmissão a clientes heterogêneos e garantir que o vídeo seja apresentado com a mesma qualidade com a qual foi gerado. Isto requer parâmetros de qualidade de serviço (*Quality of Service - QoS*). Qualidade de serviço é a parametrização, para o sistema, da vontade ou necessidade do usuário [Nah95]. Por exemplo, se

um cliente deseja transmitir um vídeo no formato NTSC, a taxa de *frames* deve ser de 30fps, e os sistemas devem garantir que essa taxa seja mantida. Os parâmetros de QoS podem estar relacionados com a aplicação, com o sistema operacional, com a rede ou com o dispositivo (figura 3.1) [Hac96].

Serviços de negociação e tradução são necessários para realizar comunicação de parâmetros de QoS. A negociação visa estabelecer valores de parâmetros comuns entre os serviços do usuário e do fornecedor e garantir acesso (através de reserva de recursos e testes de admissão [Buf94a]) a recursos escassos, reservando somente a demanda real [Nah95]. A tradução é necessária pois diferentes clientes possuem diferentes parâmetros de QoS [Nah95].

O gerenciamento de QoS para aplicações de mídia contínua pode ser subdividido em 2 esquemas de controle: *Estático e Dinâmico* [Buf94a]. No esquema de controle estático, o usuário simplesmente especifica o nível de QoS no momento da criação de uma determinada sessão, sendo que esse nível será mantido até o final da mesma. Por outro lado, o esquema de controle dinâmico permite que o sistema ou o programa de usuário modifique o nível de QoS durante o andamento da sessão. Essa mudança pode ser solicitada pelo gerente de QoS - quando a disponibilidade dos recursos do sistema torna-se baixa - ou pela aplicação do usuário - quando este deseja melhorar (aumentar) seu nível de QoS ou até mesmo degradá-lo.

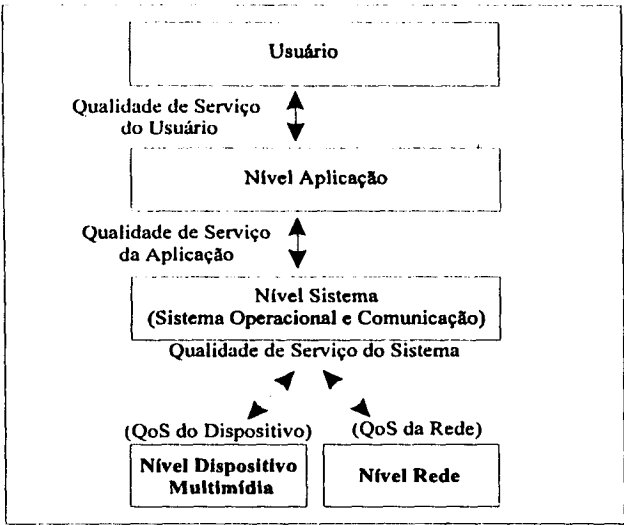


Figura 3.1 - Níveis de QoS [Hac96].

3.3 - SERVIÇO DE VÍDEO SOB DEMANDA

Aplicações VOD são divididas em duas categorias, incluindo televisão interativa (ITV - *Interactive TeleVision*) e aplicações corporativas/educacionais. O foco da ITV é a apresentação de vídeo a partir de uma pequena coleção. O foco das aplicações corporativas/educacionais é a apresentação aleatória de dados de vídeo, selecionados de uma grande coleção de vídeos de tamanhos variados. Um sistema VOD para ITV pode armazenar dados de vídeo economicamente em discos magnéticos. Contudo, devido ao grande volume de dados de vídeo armazenados para uma aplicação corporativa/educacional, um sistema VOD para esse tipo de aplicação tipicamente usa uma combinação de discos magnéticos com armazenamento terciário [Ber95].

No desenvolvimento de um sistema VOD, pesquisadores tem que resolver muitos problemas. Primeiro, o sistema deve estar apto a localizar um vídeo desejado. Segundo, o sistema deve armazenar e gerenciar uma grande coleção de dados de vídeos estruturados. Terceiro, o sistema deve transportar um vídeo selecionado para o usuário final e executá-lo [Ber95].

Para localizar um vídeo desejado, uma interface amigável deve ser criada. A interface irá suportar pesquisas em uma base de metadados (seção 3.7), e então ajudar o usuário a encontrar o vídeo no repositório. O projeto da interface de pesquisa deve considerar quais tipos de pesquisas devem ser suportadas e como as pesquisas devem ser especificadas.

Armazenamento e gerenciamento de material de vídeo digital é difícil devido ao enorme tamanho desse material e do custo resultante desse armazenamento. Similarmente, o transporte e a execução de dados de vídeo requer um certo custo por *stream* porque os dados de vídeo requerem entrega em tempo real e consomem considerável largura de banda de rede quando transmitidos [Ber95].

Os aspectos de localização, armazenamento, gerenciamento e transporte são discutidos em mais detalhes nas próximas seções desse capítulo.

Serviços de vídeo interativos são similares ao serviço de difusão de TV. A diferença é que a interface da TV foi projetada para suportar muitos canais simultâneos sobre um mesmo banco de dados. No sistema de TV tradicional, cada estação realiza a sua programação simultaneamente com outras estações, e o usuário seleciona um canal específico. Em contraste, um sistema

interativo faz com que toda a programação seja apresentada ao usuário sem nenhuma restrição. O usuário pode selecionar qualquer parte da programação a qualquer hora [Lit94].

De acordo com o nível de interatividade, os serviços de vídeo podem ser classificados em [Hac96]:

- Serviços *Broadcast* (No-VOD), similar aos canais de televisão, em que o usuário é um participante passivo e não possui nenhum controle sobre a sessão.
- Serviço *Pay-Per-View* (PPV), em que o usuário paga por uma programação específica, similar ao serviço existente em CATV (*Cable TV*).
- Serviço *Quasi video-on-demand* (Q-VOD), em que os usuários são agrupados pelo interesse comum que possuem. Os usuários possuem um controle rudimentar, ativado por chaveamento para os diferentes grupos.
- Serviços *Near Video-On-Demand* (N-VOD), em que funções de avançar e voltar são simuladas por transições em um intervalo de tempo discreto (da ordem de 5 minutos). Isto é feito utilizando vários canais que possuem a mesma programação, porém em tempos diferentes.
- Serviço *True Video-On-Demand* (T-VOD), em que o usuário possui total controle sobre a apresentação. O usuário tem todas as funções de VCR, incluindo avançar, voltar, congelar e posicionar diretamente. T-VOD aloca para cada usuário um canal exclusivo; com isso, o número de usuários do serviço fica limitado pelo número de canais disponíveis. T-VOD necessita de um sinal bidirecional entre o usuário e o controle central. Serviços T-VOD são essenciais para aplicações de jogos interativos, onde todas as ações dos jogadores são vistas pelos outros jogadores.

Os serviços interativos podem ser usados nas mais diversas aplicações, como visto na tabela 3.1, abaixo.

Tabela 3.1 - Aplicações de serviços interativos [Hac96].

Aplicação	Descrição
Movies-on-Demand (Filmes sob Demanda)	O consumidor pode selecionar e ver o vídeo com total capacidade de interação.
Vídeos Game Interativos	Pode-se jogar sem ter que adquirir uma cópia do jogo.
Noticiário de TV Interativo	O consumidor é capaz de selecionar uma notícia bem como seu nível de detalhe.
Catálogos interativos	O consumidor pode examinar e comprar produtos .
Aprendizado a distância	Um estudante se inscreve em um curso, remotamente. O estudante pode ter preferências e horários individuais.

Um sistema VOD pode ser acessado simultaneamente por centenas de usuários com diferentes preferências. O sistema deve ser capaz de garantir uma qualidade de serviço satisfatória a todos os clientes. Um cenário típico de um VOD consiste em um banco de dados local, conectado a uma casa por uma rede de comunicação. Na casa do usuário, há uma interface de rede e um monitor. A interação do usuário com o sistema pode ser feita através de um controle remoto ou um teclado de computador. Esse sistema consiste na informação/arquivo de programação armazenada em um servidor central. Essa informação é transmitida à servidores regionais através de uma rede de alta velocidade (a rede de alta velocidade interliga os servidores regionais), ficando a disposição dos usuários. Os usuários ligam-se aos servidores regionais.

Sessões VOD são geralmente longas e requerem alta banda passante para transmissão, utilizando por isso técnicas de compressão de vídeo para diminuir o número de bits a serem transmitidos. Algoritmos e técnicas para compressão de vídeo (como por exemplo MPEG, JPEG, H.261) são discutidos no capítulo 2.

3.4 - SISTEMA OPERACIONAL

O sistema operacional controla todos os componentes de um sistema computadorizado, como por exemplo CPU, memória principal, armazenamento e todos os dispositivos de entrada e saída. Ele é fundamental para que um sistema multimídia tenha um bom desempenho. Sistemas operacionais convencionais foram projetados para oferecer um uso justo dos recursos do sistema entre os programas concorrentes, porém tais sistemas sofrem a falta de serviços de tempo real. Por exemplo, a abstração de processo UNIX [Tan93] é um conceito muito útil do ponto de vista

de programação, mas a troca de contexto dos processos demanda um custo de tempo muito alto para aplicações de mídia contínua. Esse alto custo leva a escolha de pacotes de *threads* do nível do usuário que possuem uma troca de contexto mais suave, contudo tais pacotes não suportam *threads* preemptivas úteis a sistemas multimídia [Tan93, Cou94].

3.4.1 - PROBLEMAS COM OS SISTEMAS OPERACIONAIS CONVENCIONAIS

A seguir são apresentados os problemas básicos para o suporte de aplicações de mídia contínua em sistemas operacionais convencionais [Buf94a].

Latência de Interrupção. A grande latência das interrupções é uma das razões pelas quais os sistemas operacionais convencionais não suportam muito bem atividades de tempo real. Aplicações de mídia contínua produzem um número freqüente de interrupções no *kernel* resultando em pesadas trocas de contexto.

Todo *kernel* não preemptivo provê latência de interrupção muito grande. O processamento dos eventos de tempo real que disparam uma nova *thread* (ou um novo processo) devem esperar até que o processamento atual do *kernel* termine, e isto demora vários milissegundos.

Existem três abordagens para reduzir a latência de interrupções no *kernel*. A primeira torna o *kernel* altamente preemptivo modificando sua estrutura interna. A segunda adiciona um conjunto de pontos de preempção no *kernel* existente. A terceira abordagem é converter o *kernel* atual em um programa do usuário e executá-lo no topo de um *microkernel*.

Gerenciamento de Inversão de Prioridades. Quando um programa tempo real compartilha os mesmos recursos com um programa não tempo real, ocorrem casos onde o primeiro deve esperar pelo término do segundo. Por exemplo, em aplicações que compartilham o mesmo servidor de rede, um pacote de alta prioridade deve esperar até que todos os pacotes de menor prioridade, que já estavam na fila, completem sua execução. Essa inversão de prioridade causa espera não previsível e problemas de *jitter*.

Gerenciamento de *Deadline* e Recuperação. Ao contrário de aplicações de tempo real, muitas aplicações de mídia contínua possuem um *deadline* inerentemente mais suave. Por exemplo, pode-se prosseguir uma vídeo-conferência mesmo que a imagem de vídeo mais recente

não possa ser processada a tempo. A perda de um *deadline* não leva ao caos total, porém a notificação da perda é uma informação importante, pois baseada nessa informação a aplicação pode querer mudar o nível de QoS.

Quando a aplicação perde um *deadline* devido à sobrecarga ou a erros de *hardware* ou *software*, o programa do usuário deve estar apto a tomar uma decisão. A decisão geralmente é uma recuperação que, freqüentemente, deve executar com uma prioridade maior que a atividade principal, logo necessita aumentar sua prioridade primeiro.

Gerenciamento de Qualidade de Serviço e Controle de Admissão. Sistemas operacionais convencionais não possuem um mecanismo para manter um nível de QoS para aplicações de mídia contínua ou esquemas de gerenciamento e prevenção de sobrecarga, o que pode acarretar problemas de espera (*delay*) não predizível e *jitter* [Buf94a, Ste95a].

3.4.2 - GERENCIAMENTO DE RECURSOS

Num sistema multimídia, as garantias de tempo para o processamento de mídia contínua devem considerar todo o sistema de *software* e *hardware* ao longo de toda a rota de comunicação. Os requerimentos dependem do tipo de mídia e da natureza das aplicações. Em um sistema multimídia distribuído, várias aplicações competem por recursos, os quais merecem uma alocação cuidadosa. O gerenciamento do sistema deve então empregar algoritmos de escalonamento apropriados para garantir que os recursos sejam devidamente alocados às aplicações de acordo com suas necessidades.

Uma possível realização de alocação e gerenciamento de recursos baseia-se na interação entre clientes e seus respectivos gerentes de recursos. O cliente seleciona o recurso e requisita a alocação do mesmo declarando suas necessidades através de uma especificação de QoS. Primeiramente, o gerente de recursos checa a utilização de seu recurso e decide se a solicitação de reserva pode ser atendida ou não. Todas as reservas existentes são mantidas e, caso seja necessário, há negociação com outros gerentes de recurso (através de protocolos de reserva de recurso, como ST-II) [Ste95].

3.4.3 - SUPORTE ARQUITETURAL

Existem dois modelos de arquitetura para o núcleo de um sistema operacional: o *kernel* monolítico e o *microkernel*. O *kernel* monolítico (como o UNIX) mantém todos os serviços oferecidos pelo sistema em seu interior. Um *microkernel* implementa apenas os serviços básicos como gerência de espaço de endereçamento, memória virtual, ciclos de processador, interrupções, etc. Todos os outros serviços, como serviço de arquivos, estão implementados fora do *kernel*, no espaço do usuário [Tan93, Cou94].

Uma característica chave no projeto de um sistema distribuído é a **flexibilidade** [Cou94, Mul93]. Um *microkernel* tem a vantagem de ser altamente modular: existe um interface bem definida para cada serviço, e todo serviço é igualmente acessível para qualquer cliente, independente de sua localização. Além disso, é fácil instalar, remover ou depurar novos serviços, sem a necessidade de parar o sistema para reinicializar o *kernel* (como é o caso do *kernel* monolítico). Essa habilidade de adicionar, retirar ou modificar serviços que provê ao *microkernel* flexibilidade [Tan93]. Essa flexibilidade, aliada à extensibilidade e facilidades em comunicação tornam a arquitetura *microkernel* apropriada para sistemas de tempo-real e especificamente multimídia.

Silberschatz ([Sil95]) detalha a arquitetura *microkernel* bem como sua aplicação no sistema operacional Mach e Coulson ([Cou92]) mostra extensões aplicadas ao *microkernel* do sistema operacional Chorus para dar suporte à multimídia.

3.5 - SISTEMAS DE COMUNICAÇÃO

As redes em sistemas multimídia distribuídos devem dar suporte a um conjunto de serviços e requisitos como: tráfego em tempo real, garantia de alto desempenho e difusão múltipla. Os protocolos existentes como o OSI e o TCP/IP [Com95] não satisfazem esses requisitos. Eles foram projetados para operar em sub-redes de baixa confiabilidade. Apesar da predominância dos protocolos da família TCP/IP, eles são inadequados para multimídia pois: o protocolo de transporte (TCP) não suporta nenhum parâmetro de qualidade de serviço, a verificação de erros por *checksum* e o controle de fluxo através de janelas deslizantes introduzem

um *overhead* muito grande na transmissão, existe a ausência de conexões multi-ponto, não há limite máximo garantido para atraso nem para o *jitter* e, não possui a previsibilidade necessária para operar em tempo real [Car95]. O uso de protocolos TCP/IP é viável para aplicações multimídia quando a sub-rede de comunicação assegura alto desempenho e confiabilidade [Car95], como a FDDI-II por exemplo. Também existe uma grande expectativa em torno das redes digitais de serviços integrados em banda larga (BISDN) usando tecnologia ATM [Com95].

Abaixo estão descritas algumas das características das principais tecnologias de rede em uso atualmente [Hac96]:

- Ethernet: é a tecnologia mais utilizada, trabalhando a uma taxa de transferência de 10 *Mbps*; com isso pode transmitir no máximo quatro vídeos comprimidos ao mesmo tempo. Utiliza o protocolo CSMA/CD, o que é uma desvantagem, pois esse protocolo é não determinístico. Em situações de alta carga não há controle sobre o *delay* de acesso. Em aplicações tradicionais, não há problema, mas em aplicações multimídia, onde existe a necessidade de sincronismo e tempo real, torna-se um problema sério, impossibilitando com isso a criação de um sistema multimídia satisfatório, a não ser que seja controlado o número de estações [Stu95].
- Token Ring: o protocolo de acesso é muito melhor que o da Ethernet para suportar dados multimídia. Uma vantagem é a taxa de transferência que é de 16 *Mbps*, ao invés de 10 *Mbps*. A característica mais importante é o nível MAC de prioridades. Ele pode ser usado para separar dados em tempo real (alta prioridade) e dados normais (baixa prioridade). Desta forma, as estações determinam que tipo de dado vai ser transmitido e se é de alta ou baixa prioridade, conseguindo com isso viabilizar comunicação multimídia.
- 100 Base-T: é a evolução da Ethernet (IEEE 802.3), e utiliza o mesmo protocolo de acesso CSMA/CD. Com isso, raramente excede 50 % de utilização da *bandwidth*, porém, como a taxa de transferência é de 100 *Mbps*, consegue uma taxa de transferência real de 50 *Mbps*, transmitindo um grande número de dados multimídia, mas não garantindo o *delay*. 100 Base-T é uma escolha satisfatória para pequenas e médias configurações.
- 100 VG-AnyLAN: outra tecnologia de rede a 100 *Mbps*, também conhecida por 100-*Mbps* Demanda de Prioridade, é uma evolução do padrão Ethernet e Token-Ring para 100 *Mbps* sobre par trançado categoria 3. Seu principal objetivo é aumentar a taxa de transferência, utilizando-se

dos cabos e interconexões existentes. Uma rede 100VG-AnyLAN consiste em um *hub* central ou repetidor, que gerencia o acesso da rede por execução contínua de um esquema de controle de acesso “*round robin*”. Isto garante que todas as requisições serão atendidas por ordem de chegada e por prioridade (nesse esquema não há colisões). Esse tipo de rede é melhor que a 100 Base-T, e é uma alternativa viável para comunicações multimídia, em particular para topologias que possuam um pequeno número de estações.

- FDDI: conceitualmente o FDDI é um superconjunto de *fast token ring*. A grande *bandwidth* pode suportar um grande número de estações multimídia. Para utilizar FDDI para comunicações multimídia necessita-se de um esquema de gerenciamento de taxa de transferência adicional, porém o seu custo é bastante alto.
- FDDI II: é uma rede baseada sobre um protocolo de *slotted-ring*, derivada da FDDI para suportar melhor o tráfego em tempo-real. Possui um complexo esquema de gerenciamento dinâmico da *bandwidth* entre dados assíncronos, síncronos e isócronos. FDDI II é parcialmente compatível com FDDI, já que toda a estação FDDI II suporta o tráfego de pacotes FDDI, mas o contrário não acontece.
- X.25: é baseado no protocolo *sliding windows* e no mecanismo de recuperação de erros, obtendo desta forma variações na entrega de pacotes e *delay* incontrolável. É utilizado em *links* com velocidades acima de 64 Kbps. Não possui *multicasting*. Por esses motivos, não é recomendado para redes multimídia.
- ISDN: foi desenvolvido para suportar uma grande variedade de serviços, sobre dados, fax e vídeo. É construído sobre um canal síncrono de 64 Kbps, que pode ser usado para *bit-streams* contínuos ou no tráfego de taxa de *bits* constante, como o H.261 vídeo, ou para comunicação por pacotes. ISDN é largamente utilizado na Europa. Pode ser utilizado na comunicação multimídia, tanto ponto a ponto quanto multi-ponto.
- ATM: na rede ATM, todas as informações são adicionadas dentro de um *slot* de tamanho fixo chamado de célula, que é identificada e comutada por um rótulo (*label*) no cabeçalho (*header*). A arquitetura ATM provê os ingredientes necessários para suportar requisições de aplicações multimídia. Devido a grande atenção dada à tecnologia ATM, ela será melhor explicada na próxima seção.

3.5.1 - A ABORDAGEM ATM (*ASYNCHRONOUS TRANSFER MODE*)

Esta seção discute a tecnologia ATM devido às facilidades que esta apresenta ao suporte de dados de mídia contínua, em especial. Porém o projeto do Sistema Tutor prevê a utilização de redes padrão Ethernet por esta já estar amplamente instalada, reduzindo assim os custos.

A rede ATM [Bou92] resulta da junção de dois conceitos: o de comutação de pacotes e o de multiplexação por divisão de tempo (*Time Division Multiplexing*, TDM). ATM é a tecnologia que está sendo utilizada na rede pública B-ISDN. O ITU define o modelo de referência do protocolo para B-ISDN, mostrado na figura 3.2.

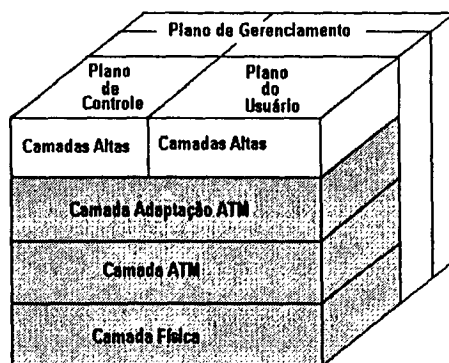


Figura 3.2 - Modelo de referência do protocolo para B-ISDN.

A evolução das redes de longa distância (WANs, *Wide Area Networks*) começa com a comutação de pacotes, passa pelo chaveamento de quadros (*Frame Relay*) e chega à comutação de células. A evolução foi ditada pelo aumento constante tanto da confiabilidade dos meios de transmissão quanto da velocidade dos equipamentos de comutação.

Tanto a comutação de pacotes quanto o chaveamento de quadros transmitem os dados em quadros de tamanho variável, tipicamente contendo centenas ou milhares de *bytes* de informação. A tecnologia de comutação de células emprega pacotes (denominados agora de células) de tamanho fixo igual a 53 *bytes*: cinco de cabeçalho e quarenta e oito de dados. A idéia é comutar rapidamente essas células durante seu trajeto. Essa filosofia é denominada banda sob demanda pelo fato da rede prover exatamente a banda requisitada pelas aplicações. ATM comuta as células

na velocidade em que são submetidas para transmissão (toda a comutação é feita por *hardware* dedicado a esse fim).

A camada ATM (figura 3.3) emprega células com o formato dado pela figura 3.4. Os campos VCI (*Virtual Channel Identifier*) de doze *bits* e VPI (*Virtual Path Identifier*) de dezesseis *bits* identificam uma conexão. Comer [Com95] explica a função de cada campo de uma célula ATM, bem como os processos de estabelecimento de conexão e de comutação.

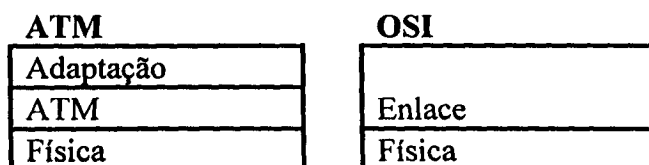


Figura 3.3 - Camadas de uma rede ATM comparadas com o modelo OSI.

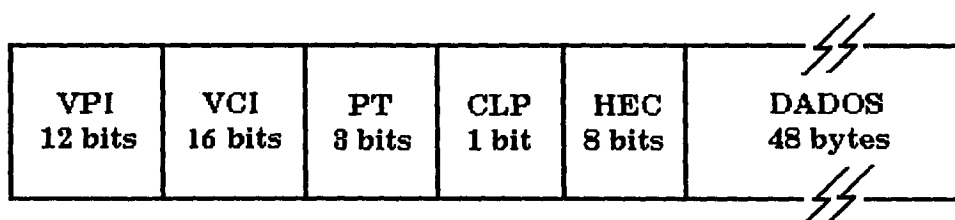


Figura 3.4 - Formato de uma célula ATM.

A camada de adaptação (figura 3.3) fornece serviços às camadas superiores abstraindo as particularidades da tecnologia ATM. A tabela 3.2 fornece os tipos de serviços oferecidos por essa camada. Interessa para aplicações multimídia distribuídas os serviços classe A e B. Esses serviços apresentam características de tráfego isócrono apesar da tecnologia ATM não prover esse tipo de taxa de transmissão. Serviços classe A são destinados a mídias que requerem taxas de transmissão constante como áudio e vídeo não compactados. Serviços classe B são destinados a mídia que demandam taxas variáveis como áudio e vídeo compactados (a taxa de compactação depende das características momentâneas da mídia). Em ambas as classes os dados chegam ao receptor em sincronismo com o emissor, isto é, a rede não altera a ordem temporal dos segmentos de mídia no fluxo.

Tabela 3.2 - Tipos de serviço oferecidos pela camada de adaptação.

Classe	A	B	C	D
Sincronismo fonte-destino	Garantido		Não garantido	
Taxa (bits/s)	Constante		Variável	

Serviços classe A e B são implementados através de uma base de tempo comum, ou, na inexistência desta, através de “bufferização”. Dada a velocidade de comutação das células ATM, o atraso de propagação e o *jitter* são muito baixos, o que propicia certa facilidade para a camada de adaptação prover a devida sincronização entre o emissor e o receptor.

3.6 - ARMAZENAMENTO E ACESSO

Dados de mídia contínua requerem um *throughput* maior do que o disponível com a tecnologia convencional de disco magnético [Lou93]; tal tecnologia não suporta a largura de banda necessária para múltiplas *streams* de serviço. A entrega de vídeo sob demanda requer uma largura de banda muito alta e, mesmo com redes velozes, pode ocorrer congestionamento no sistema de entrada/saída do servidor de arquivos.

As restrições de banda e de entrega em tempo real sugerem o uso de servidores localizados nos clientes para a execução de *playback* de dados multimídia. Por outro lado, o desejo de se compartilhar tais dados (como vídeo) e o custo efetivo dos dispositivos de armazenamento terciário, motivam a utilização de um repositório central. O que é necessário é manter o alto desempenho e escalabilidade de servidores distribuídos enquanto o custo do armazenamento centralizado é preservado. Assim, a solução pode ser a adoção de um sistema de armazenamento hierárquico distribuído [Fed94].

Na tentativa de satisfazer as necessidades de entrega em tempo real, alguns sistemas utilizam um servidor poderoso, empregando alguma técnica de *disk array* [Cha94] (para suprir a deficiência da tecnologia convencional de discos magnéticos) como a tecnologia RAID (*Redundant Array of Inexpensive Disks*) [Sil95, Lee94]. Uma outra abordagem poderia aplicar paralelismo também a nível de servidor, deixando que vários servidores realizem a tarefa de entrega dos dados [Fed94].

Muitos sistemas operacionais oferecem um mecanismo de armazenamento básico para arquivos, utilizado pelas aplicações. Aplicações modernas gerenciam estruturas de dados que não são muito bem modeladas como arquivos: planilhas, gráficos, hipertexto e objetos multimídia. Em vez de um modelo de uma única *stream* de *bytes* serial, documentos compostos por uma ou mais dessas estruturas são melhor modelados como uma coleção de objetos compostos interconectados [Dea95]. Esse novo modelo deve oferecer múltiplas facilidades de armazenamento para as aplicações e para documentos compostos, e também deve oferecer um mecanismo flexível para organização dos dados em disco e para gerenciamento de *cache*, de forma que diferentes modelos e diferentes aplicações utilizem a melhor política de disco e de *cache* para aumentar o desempenho [Dea95]. Sistemas operacionais como o Windows NT [Fed94] e o Mach [Fed94, Sil95], possuem interfaces para executar programas projetados para outros sistemas. Cada interface executa no topo de um *kernel* comum. Essa flexibilidade deveria estar presente também no armazenamento de dados, de forma a permitir que cada dado (texto, imagem, vídeo, etc.), fosse armazenado de acordo com o modelo mais conveniente, ao invés de todo dado ser armazenado conforme um único modelo [Dea95].

A habilidade para construir objetos compostos permite a um hiperdocumento ser composto por vários subobjetos, os quais podem ser compartilhados por outros hiperdocumentos. A decomposição da mídia em subobjetos compartilhados tem como vantagens: economia de espaço de armazenamento (um vídeo pode ser compartilhado por vários hiperdocumentos) e melhoria no desempenho (buscas podem retornar pequenas peças de vídeo ao invés do hiperdocumento inteiro) [Fed94].

3.7 - BUSCA E RECUPERAÇÃO

Uma questão crítica no projeto de sistemas de informação multimídia é a arquitetura de distribuição dos dados. Devido ao enorme volume de dados envolvido, é impraticável armazenar e gerenciar todas as informações de distribuição a partir de um único *site*. Além disso, a natureza interativa das pesquisas em uma base de dados centralizada, requer que o gerente da base de dados torne possível tanto o gerenciamento da base de dados quanto a funcionalidade na entrega de mídia contínua [Lit94].

Dados multimídia necessitam de novas técnicas para indexação e busca. Para simplificar o mecanismo de busca e aumentar a interatividade, muitos sistemas de base de dados empregam um mecanismo de **metadados**. Dados multimídia consistem de dois componentes: os dados crus em si e informação semântica contida nos dados [Gri96]. Diferente de dados alfanuméricos, tipicamente representados por valores ASCII, o conteúdo (valores de *pixels*, por exemplo) de dados multimídia não contém muita informação interessante. Nós gostaríamos de indexar a informação que os *pixels* representam, isto é, o conteúdo semântico dos dados ao invés de indexar os valores em si. Por exemplo, a pesquisa pela cadeia de caracteres alfanumérica 'abc' é direta. Porém, a pesquisa pela imagem de um carro ou pelo som de um leão em um *videoclip* requer a habilidade de acessar o conteúdo semântico. Esse conteúdo semântico é denominado **metadado** [Gri96].

Muitos dados multimídia contém dois tipos de metadados que podem ser indexados. Primeiro, a informação multimídia é armazenada como informação de registro. Exemplos de informações de registro incluem quando a imagem foi criada, onde o vídeo foi filmado, quem gravou o áudio, etc.. Segundo, informações multimídia contém informações semânticas que devem ser extraídas manual ou automaticamente. A identificação automática de metadados em multimídia é discutida por Griffioen [Gri96].

A vantagem de se utilizar metadados é que essa abordagem permite ao usuário examinar o conteúdo da base de dados sem ter que recuperar objetos volumosos.

Existem vários requisitos que um esquema de gerenciamento de metadados deve atender para suportar aplicações multimídia interativas. Primeiro, são necessários serviços básicos para suportar identificação, localização, nomeação e distribuição de objetos. Segundo, modelos de aplicações específicas são necessários para suportar funcionalidade nas aplicações [Lit94].

O conceito de metadados é ideal para aplicações multimídia interativas nas quais vários objetos podem estar envolvidos em uma única apresentação. Por exemplo, um filme pode envolver a coordenação de *streams* de áudio e de vídeo. Um sistema de aprendizado à distância pode envolver a apresentação de áudio, vídeo, gráficos e texto. O servidor de metadados deve manter o relacionamento entre esses diversos objetos e informar sua presença para o cliente quando uma pesquisa é feita. No contexto de um servidor VOD, o sistema deve armazenar diferentes atributos de filmes. A identificação desses atributos é uma parte importante no projeto

de uma base de metadados pois são esses atributos que irão responder às pesquisas do usuário. Desse modo, a identificação dos atributos é feita através da identificação das pesquisas realizadas pelo usuário, provendo índices para responder a essas perguntas [Row94]. Índices podem ser: bibliográficos (título, assunto, gênero, diretor, produtor, elenco, etc.); estruturais (hierarquia: filme-segmento-cena); de conteúdo (para buscas baseadas em conteúdo ou em palavras-chave associadas a um conteúdo) [Row94, Ber95].

Para bases de dados armazenando vídeo, atributos de recursos incluem a carga paramétrica em subsistemas individuais de armazenamento e em sub-redes; e o número de usuários atualmente acessando o sistema VOD. O servidor de recursos mantém um registro da carga paramétrica e do tempo de uso do sistema através de uma tabela de estados. Desde que as decisões de alocação são baseadas nessas informações, é importante mantê-las atualizadas [Lit94].

Em sistemas de entrega de informação, onde a função primária da base de dados é a recuperação de dados, os servidores de recursos e de metadados devem “saber” quais são os filmes e recursos disponíveis para prover sua entrega. Quando os recursos estão sobrecarregados ou não estão disponíveis, o servidor de metadados deverá encontrar fontes alternativas de informação, se disponíveis, ou informar ao usuário a não disponibilidade. Essas funções devem ser transparentes para o usuário.

Para gerenciar seus recursos, o servidor de recursos deve coletar informações periódicas dos sistemas gerenciados. Isto pode incluir relatório de eventos como também respostas periódicas para perguntas de vários servidores de dados. O servidor de recursos também necessita manter informações sobre mudanças na configuração do sistema, as quais podem ser relocação de recursos, falha ou anúncio de novos recursos.

A principal atração do uso de servidores de recursos e de metadados é a flexibilidade desses servidores. Quando uma atualização de recurso ocorre, somente a entrada na base de dados necessita ser atualizada para a mudança se tornar visível para toda a comunidade de usuários. Finalmente, o servidor de metadados em si pode ser replicado, provendo um nível adicional de desempenho.

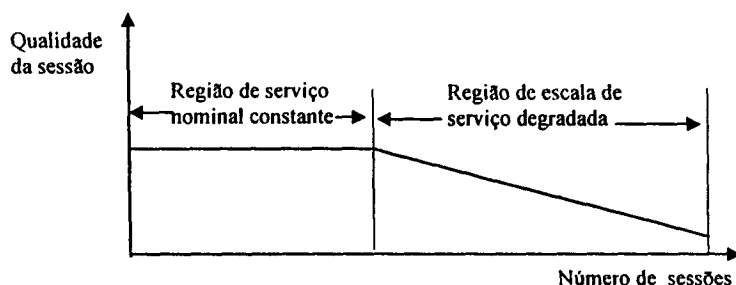


Figura 3.5 - Qualidade X Número de sessões em um sistema VOD [Lit94].

O servidor de recursos pode também ser usado para prover uma degradação suave dos serviços quando o sistema está sobrecarregado. Por exemplo, servidores de armazenamento multimídia tem uma capacidade limitada para suportar sessões concorrentes. Quando essa capacidade é excedida, a qualidade da sessão pode degradar abruptamente. Contudo, o conhecimento desses limites habilita o servidor de recursos a prover uma degradação suave (figura 3.5), ou a procurar fontes alternativas de dados replicados.

4.1 - INTRODUÇÃO

O movimento atrai o ser humano, estimulando o interesse, aumentando a compreensão e melhorando a retenção da informação. O movimento (em forma de vídeo) tem sido usado para: atrair a atenção; transportar o espectador a outros locais; transpor barreiras de cultura, linguagem e analfabetismo; treinar pessoas de forma ágil em novas ferramentas; dar credibilidade ao apresentador, à mensagem e à organização; entre outras. Quando usado de forma correta, o vídeo fornece ao público informações que ele não entenderia tão bem de outra forma, possuindo a capacidade de comunicar a maior quantidade de informação no menor tempo possível e com maior impacto.

Lindstrom (1995) cita que, em geral, a utilização de vídeo se destaca nas seguintes áreas de apresentação: descrever movimento; demonstrar procedimentos; transmitir emoção e simular situações. Além disso, o vídeo tem se tornado uma ferramenta imprescindível para a comunicação de informações, ao ponto de emissoras de TV colocarem alguns de seus programas na Internet (como a CNN, por exemplo).

Entre as vantagens da utilização de vídeo digital no desenvolvimento e realização de apresentações estão (Lindstrom, 1995):

- Economia de custo: software e hardware, mais baratos que os equipamentos analógicos profissionais, podem produzir vídeo com qualidade razoável, além de reduzir o tempo de edição.
- Interatividade ampliada: devido aos discos rígidos (rápidos) realizarem acessos aleatórios quase instantaneamente, podemos parar, retroceder ou avançar a apresentação de uma maneira bastante satisfatória.

- **Flexibilidade aumentada:** pelo fato do vídeo analógico ser baseado no tempo, ele somente pode ser armazenado no tamanho total da tela na qual foi capturado. Se o vídeo for reproduzido em uma pequena janela na tela, ainda assim ele deve conter toda a informação de sinais para uma tela inteira. O vídeo digital pode ser capturado e armazenado de acordo com o tamanho particular da apresentação final.
- **Maleabilidade:** o vídeo digital permite que se façam alterações de forma mais rápida e fácil, também permite a utilização de modelos.

4.2 - PRODUÇÃO DE VÍDEO PARA COMPUTADORES

Existem duas alternativas para a produção de vídeo para uso em computadores: a produção digital e a produção analógica.

Na produção digital um computador está presente durante todo o processo de captura. Como vantagem essa alternativa elimina o passo adicional da digitalização e evita possíveis degradações de qualidade vindas da utilização do *tape* analógico; porém necessita de muito espaço em disco.

A produção analógica tem como vantagens não necessitar de um computador para a captura de vídeo e o *tape* para armazenamento é barato. As desvantagens são que para obter os melhores resultados são necessários equipamentos de gravação profissionais (caros), além de necessitar passar pelo processo de digitalização (Luther, 1995).

A decisão de qual tipo de produção utilizar (digital ou analógica) depende de vários fatores: do nível de qualidade desejada para o vídeo final; de aspectos econômicos; da quantidade de editoração (pós-produção) desejada (Lindstrom, 1995).

O grande problema para a utilização de vídeo digital é o tamanho dos arquivos gerados. Uma produção simples com 1 minuto de duração, quadros com tamanho 640 x 480, taxa de captura de 15 quadros por segundo e com 16 milhões de cores (24 bits), necessita de uma taxa de transferência de 14MB³ por segundo, que é maior que as taxas de CD-ROMs ou mesmo discos rígidos. O tamanho desse arquivo (sem compactação) chega a 800MB. Dessa forma, a

³ MB: significa Megabytes, enquanto Mb significa Megabits.

compressão (capítulo 1) torna-se crucial para a utilização de vídeo em computadores, embora ela degrade a qualidade.

Para ser exibido, o vídeo deve ser descomprimido em tempo real e então apresentado. Existem três maneiras de se apresentar vídeo em um computador, que serão discutidas abaixo.

A primeira é utilizando um hardware especialista para realizar a descompressão e a apresentação, obtendo uma melhor qualidade. O padrão MPEG é uma das principais técnicas para esse tipo de apresentação.

Uma placa para descompressão MPEG custa entre 250 e 1500 dólares, oferecendo alta qualidade de apresentação. Também existem softwares para descompressão MPEG, porém a qualidade da apresentação é bastante degradada. Já as placas para captura (compressão) são bem mais caras, podendo chegar a 20000 dólares. Então, devido aos altos custos para aquisição de vídeo MPEG, normalmente captura-se o vídeo em algum outro formato (discutiremos alguns mais adiante) convertendo-o para MPEG (por meio de softwares de conversão disponíveis na Web) e, realiza-se a apresentação por hardware. O processo de conversão, na maioria das vezes, degrada um pouco a qualidade.

A segunda forma de apresentação de vídeo é utilizando softwares (conhecidos como *players* ou CODECs) para realizar a descompressão e a apresentação. Esta forma consome muita memória e requer muito trabalho do processador, então geralmente os vídeos apresentados por este método são exibidos em pequenas janelas na tela (algo como 160x120, isto equivale ao tamanho do quadro).

Um *player* utiliza um formato de arquivo de vídeo padrão, como MOV e AVI- que também são extensões dos nomes dos arquivos.

Arquivos MOV e AVI são muito parecidos no que diz respeito à forma de apresentação e plataformas. Ambos podem ser apresentados via *player* ou na Web via um *plug-in* instalado em um *Browsers*, ambos possuem taxas de compressão menores que as do MPEG, foram projetados para realizar apresentação sem auxílio de hardware e consomem muito processador e memória. Quando é necessária uma editoração mais apurada, utilizam-se softwares especiais, como o Adobe Premiére, que também realiza conversão AVI-MOV.

A terceira forma é utilizar software para a apresentação do vídeo e hardware para ajudar na descompressão, aumentando a qualidade. Este hardware é um acelerador gráfico, que têm como

vantagens: prover descompressão via hardware, o que aumenta a taxa de transferência de dados; capturar e enviar dados mais rapidamente para a tela, o que possibilita tamanhos de quadros maiores. Apesar dos aceleradores melhorarem o desempenho e a qualidade da apresentação, a apresentação via hardware ainda é superior.

A tabela 4.1 compara o mesmo vídeo (Luther, 1995a) de 15.4 segundos gravado em vários formatos.

Tabela 4.1: Comparação utilizando configurações diferentes

Arquivo	Áudio				Vídeo				Tama- nho
	Amos- tragem	Canal	Bit p/ Amost.	Compr.	Tam. Frame	FPS ⁴	Cor	Compr.	
MPEG	44.1 kHz	mono	16	MPEG1	352 X 240	30	YCbCr 4:2:0	MPEG1	3.1MB
MOV	44.1 kHz	mono	8	não compr.	320 X 240	25	24 bits	Apple Compact Video	3.6MB
AVI	22.05 kHz	mono	8	não compr.	320 X 240	25	24 bits	Cinepak	3.4MB
AVI	22.05 kHz	mono	8	não compr.	320 X 240	25	24 bits	Indeo	3.3MB

Os vídeos da tabela 4.1 possuem configurações diferentes, porém nota-se que mesmo possuindo taxas de amostragem e tamanho de quadro maiores, o arquivo MPEG é menor, resultado do algoritmo de compressão (MPEG-1) utilizado. O impacto do algoritmo de compressão também pode ser constatado nos dois arquivos AVI, eles possuem as mesmas configurações, a menos do algoritmo de compressão. Cada uma dessas diferentes configurações influi diretamente na qualidade de imagem e de apresentação (desempenho) do vídeo, quanto maiores a taxa de amostragem, a quantidade de bits por amostra, o número de *frames* por segundo, etc., melhor será a qualidade final do vídeo, porém o tamanho do arquivo será maior. Dentre esses quatro, o vídeo MPEG apresentou melhor qualidade de imagem e de apresentação.

O vídeo AVI com compactação Indeo obteve qualidade de imagem melhor que o vídeo MOV e que o vídeo AVI, ambos com compactação Cinepak, estes dois apresentando qualidade de imagem muito próximas. A apresentação do vídeo MOV foi melhor que a dos AVIs.

Utilizando o vídeo da tabela 4.1, mas agora aplicando o mesmo compressor (Indeo) e as mesmas configurações (tabela 4.2), o vídeo MOV apresentou melhores desempenho e qualidade de imagem que o vídeo AVI, tanto na forma comprimida quanto na forma não comprimida.

Tabela 4.2: Comparação utilizando o mesmo compressor

Arquivo	Áudio				Vídeo				Tama- nho
	Amos- tagem	Canal	Bit p/ Amost.	Compr.	Tam. Frame	FPS	Cor	Compr.	
MOV	22.05 kHz	mono	8	não compr.	320 X 240	25	24 bits	Indeo	3.81 MB
AVI	22.05 kHz	mono	8	não compr.	320 X 240	25	24 bits	Indeo	3.81 MB
MOV	22.05 kHz	mono	8	não compr.	320 X 240	25	24 bits	não compr.	120 MB
AVI	22.05 kHz	mono	8	não compr.	320 X 240	25	24 bits	não compr.	119.5 MB

4.3 - APRESENTAÇÃO DE VÍDEO EM REDES DE COMPUTADORES

Existem duas formas principais para a apresentação de vídeo em redes de computadores: o *downloading* e o *streaming video*.

O processo de *downloading*, bastante comum na Web, consiste em conectar um computador remoto e transferir um arquivo deste para ser apresentado em seu computador. A transferência ocorre através de protocolos de transmissão de dados pela rede (como FTP e

⁴ FPS: Frames Por Segundo (fps), é o número de quadros de vídeo apresentados em um segundo.

HTTP), utilizando um *browser* (como o Netscape Navigator ou o Internet Explorer). Nesse processo, o arquivo é primeiramente todo transferido para depois ser utilizado. A transferência é geralmente lenta devido às condições de tráfego na rede, das distâncias envolvidas, dos tamanhos dos arquivos (principalmente). Por isso os vídeos disponíveis na Web via *downloading* raramente possuem alta resolução (para reduzir o tamanho do arquivo). Uma vez que o arquivo tenha sido transferido, o vídeo pode ser apresentado em um *browser* através de *plug-ins* ou em um *player*. Um *plug-in* (assim como um *player*) é um módulo de software com uma finalidade específica, como tocar um arquivo de áudio ou apresentar um vídeo. O que o torna diferente é que ele é instalado como parte componente do *browser*, então após fazer o *downloading* com o auxílio de um *browser*, não é necessário acionar um outro programa (*player*) para assistir ao vídeo, assiste-se ao vídeo no próprio *browser*.

No processo de *streaming video*, assim que os primeiros dados chegam ao seu destino eles são apresentados, enquanto a fonte continua a transmitir os dados pela rede na forma de pacotes UDP. Existem diversos fatores a serem considerados na transmissão *streaming video*. O vídeo, como todo dado de mídia contínua (mídias que variam com o tempo), é gerado a uma certa taxa e deve ser apresentado nessa mesma taxa, de forma que o vídeo apresenta características de tráfego contínuo, exigindo uma taxa contínua de transmissão. Cada quadro tem um tempo limite para a sua apresentação (*deadline*), se esse tempo for ultrapassado, ocorrerão tremores na imagem (*jitter*) (Buford, 1994). Explicando de forma mais detalhada, imagine um vídeo com 10 minutos de duração sendo transmitido de uma máquina à outra através da rede. O quadro número 15 (hipotético) deve ser apresentado no tempo 1'00", o quadro 16 em 1'03", e assim por diante. Se o quadro 16 sofre um atraso na sua transmissão, ele poderá chegar no tempo 1'06", que é o *deadline* para a apresentação do quadro 17. Então, ou o quadro 16 será descartado ou será apresentado, causando tremores ou congelando a imagem por alguns instantes.

Se o propósito é a transmissão do vídeo em uma pequena rede local, os problemas podem desaparecer, dependendo da tecnologia de rede utilizada:

- Ethernet: apesar de ser uma tecnologia barata e bastante disseminada, não é adequada à transmissão de vídeo com alta qualidade (mesmo em redes locais) devido às suas baixas velocidades de transmissão (10 Mbps), a não ser determinística e a ocorrência muitas colisões

provocando atrasos, principalmente em situações de altas cargas (como no caso da transmissão de vídeo) (Comer, 1995).

- FDDI: é uma tecnologia mais cara que a Ethernet, porém transmite a 100 Mbps, o que a torna uma boa candidata para pequenas e médias produções multimídia. (Comer, 1995).
- ATM: a tecnologia ATM apresenta diversas facilidades ao suporte de dados multimídia, transmitindo a velocidades entre 155 e 600 Mbps. É atualmente a tecnologia mais apropriada para a transmissão de dados multimídia em geral, contudo ainda é uma tecnologia cara. (Comer, 1995; Tanenbaum, 1996).

Streaming Video é o modo de transmissão adotado para a apresentação de programas ao vivo pela Web e para a realização de videoconferências. Apesar de todos os problemas discutidos, existem tecnologias já bem desenvolvidas para suportar vídeo em tempo real através da Internet. VDO e VXtreme são duas dessas tecnologias. Ambas utilizam arquivos AVI sem compressão, convertem e comprimem o mesmo para seus próprios formatos. O VDO utiliza o formato vdo e o VXtreme utiliza o formato vxi. As taxas de compressão chegam a 500:1 com o VXtreme. A qualidade de apresentação é impressionante, em se tratando de transmissão de vídeo pela Internet. Dependendo das condições de tráfego da rede, o desempenho é igual ao de uma apresentação local. O VXtreme possui ferramentas para a captura, codificação e editoração de vídeo, podendo inclusive construir páginas HTML que interagem com o vídeo. O VDO não possui ferramenta para interação com HTML e nem ferramenta própria para editoração de vídeos, mas é compatível com várias delas, como o Adobe Premiere por exemplo. Para visualizar um vídeo VDO ou VXtreme em um computador, basta instalar o respectivo *plug-in* no *Browser*. Para transmissão ao vivo, os requisitos para máquinas clientes estão na tabela 4.3.

Tabela 4.3: Requisitos para clientes VDO e VXtreme em computadores.

	Processador	Memória	Espaço em Disco	Plataforma	Browser
Vxtreme	Pentium	16MB	6MB free	Win95 ou WinNT 4.0	Navigator 2.0 Iexplorer 3.0
VDO	Pentium 166	32MB	-----	Win95	Navigator 2.0 Iexplorer 3.0

Os vídeos apresentados pelos clientes estão armazenados em máquinas remotas que executam um programa servidor. O servidor recebe pedidos de transmissão de vídeo dos clientes e, envia o vídeo através da rede. Os requisitos de máquina para a instalação de servidores estão na tabela 4.4.

Tabela 4.4: Requisitos para servidores VDO e VXtreme em computadores.

	Processador	Memória	Espaço em Disco	Plataforma
VXtreme	Pentium	64MB	20MB free	WinNT 4.0
VDO	Pentium 166	32MB	-----	WinNT3.51/4.0

A interface do VXtreme, tanto de apresentação quanto de gerenciamento, é mais atraente e amigável que a do VDO, porém a instalação é mais complicada.

GLOSSÁRIO DE TERMOS

ADC - Analógico para digital. Refere-se ao processo de transformação de um sinal analógico em um sinal digital.

ADPCM - *Adaptive Differential Pulse Code Modulation*. É uma técnica de codificação predictiva derivada do PCM, que somente codifica a diferença entre o valor atual da amostra e um valor predictivo que resulta de alguma forma de extrapolação do valor anterior, usando uma função de predição variável. Exemplo: esquema ITU-TS G.723 para telefone.

Amplitude - No processamento de sinais, o valor obtido de um sinal. Em processamento digital, amplitude refere-se ao valor obtido de amostras individuais.

ATM - Modo de Transferência Assíncrono. Padrão ITU destinado a altas taxas de *bits* para redes *WAN* com estrutura BISDN, desenvolvido e baseado no princípio de multiplexação por divisão de tempo assíncrono. A tecnologia também é destinada para redes locais. Baseado no transporte de pequenas unidades de dados de tamanho fixo (53 *bytes*) que recebem o nome de células.

BISDN / B-ISDN - Rede digital de serviços integrados em banda larga. Termo genérico estudado e padronizado pela ITU e ISO para tecnologia de alta largura de banda suportando todos os tipos de serviços digitais em um caminho integrado.

Bandwidth - Largura de Banda. Em transmissão de dados, o número de frequências que um sistema pode transmitir sem excessiva redução. Em redes de computadores, termo usado para se referir à capacidade de uma rede, expressado em *bits* por segundo (bps).

CCITT - Comitê que faz parte da ITU. Produz padrões para telefone e para sistemas de comunicação de dados. Mudou o nome para ITU-TS.

CELP - *Code Excited Linear Prediction*. Um esquema de compressão para voz baseado em quantização vetorial.

CIF - *Common Intermediate Format*. Troca de Formatos para imagens de vídeo em movimento.

Comutação de Pacotes - Técnica de multiplexação por divisão de tempo usada em comunicação de dados, onde o fluxo de bits é dividido em pequenas unidades de dados enviadas uma a uma em um intervalo de tempo definido. As unidades de dados são também chamadas de pacotes, *frames*, ou células.

Crominância - Termo usado para se referir às diferenças de cor. Em analogia a TV *broadcast*, sinais de crominância são construídos pela combinação linear de sinais de diferentes cores.

DAC - Digital para Analógico. Refere-se ao processo de transformação do sinal digital para um sinal analógico e, daí para um fenômeno físico que o ser humano pode sentir (ver, ouvir).

DCT - Transformação Co-seno Discreta, é um método de codificação utilizado em técnicas de compressão de imagens.

Deadline - Em sistemas multimídia refere-se ao prazo de entrega ou de apresentação de um determinado dado. Exemplo: em uma transmissão de vídeo a 30 fps, cada *frame* tem um *deadline* de 0,3 segundos para ser entregue.

Delay - Em computação é o tempo de espera pela resposta de um estímulo.

Descompressão - Em computação refere-se ao processo pelo qual as informações comprimidas são expandidas pela adição de informações redundantes eliminadas no estágio de compressão.

DPCM - Modulação por Código de Pulso Diferencial. Uma técnica de codificação predictiva derivada do PCM, onde somente a diferença entre o valor atual e o valor da amostra é codificada. DPCM usa uma função de predição constante.

fps - *frames per second* (quadros por segundo), taxa de transmissão de *frames* em redes de computadores.

HDTV - *High-Definition Television*. Refere-se à tecnologia de TV com qualidade superior a da TV *broadcast* convencional. 1920 *pixels*/linha, 1080 linhas/*frame* a alta taxa de 60 *fps*.

ISO - *International Standardization Organization*, organização internacional que visa o estabelecimento de padrões.

ITU - *International Telecommunications Union*, é uma organização internacional na qual setores privados e governamentais coordenam redes e serviços globais TELECOM. Suas atividades incluem coordenação, regulamentação e padronizações de telecomunicações e a organização de eventos regionais e mundiais da TELECOM.

Jitter - Em comunicação de dados é a variação sobre o tempo de atraso do tráfego de dados na rede, provocando oscilações no sinal.

Kernel - Em sistemas operacionais é o núcleo do sistema operacional. Fornece os serviços necessários para a utilização de um computador.

Latência - Período de inatividade entre um estímulo e a resposta por ele provocada.

LPC - *Linear Predictive Coding*. Esquema de compressão de voz baseado em quantização vetorial.

Microkernel - Em sistemas operacionais é um núcleo de sistema operacional enxuto, provendo apenas os serviços básicos como comunicação. Os serviços não básicos são implementados no espaço do usuário.

Mídia discreta - Em sistemas multimídia refere-se a mídias que não mudam com o tempo, como texto, gráficos e imagens.

Mídia contínua - Em sistemas multimídia refere-se a mídias que mudam com o tempo, como áudio, vídeo e animações.

MUSICAM - Empresa que implementou um padrão de áudio e comercializa produtos baseados nesse padrão.

PAL - *Phase Alternating Line*. Nome do padrão de TV a cores *broadcast* usado na Alemanha, parte da África e parte da Ásia. 625 linhas e 25 *fps*.

PCM - *Pulse Code Modulation*. Técnica de digitalização que consiste de um passo de amostragem seguido por uma quantização e geração de palavra código, onde cada amostra é quantizada independentemente da outra. Particularmente usado para digitalização de voz.

Playback - Em sistemas multimídia refere-se à execução de apresentações multimídia.

Preempção - Em sistemas operacionais é a capacidade de se retirar um processo, que está atualmente executando, da CPU em detrimento de outro processo de maior prioridade.

QoS - *Quality of Service*. Em redes de computadores e sistemas operacionais, um conceito que as aplicações usam para especificar suas necessidades para o sistema, antes da transmissão ou processamento inicial.

SDK - *Software Development Kit*

SDM - Servidor de Dados Multimídia. Sistema em implementação no laboratório Intermídia do ICMSC-USP, por Hachisuca [Hac96].

Seek - Em computação é o movimento realizado pelo braço de leitura de um disco rígido para o seu posicionamento no local de início da leitura/escrita.

SGBD - Sistema de Gerenciamento de Base de Dados.

Stream - Em computação refere-se a um fluxo de dados.

Thread - Em sistemas operacionais, também é conhecida como processo leve. *Threads* compartilham o mesmo espaço de endereçamento e o mesmo conjunto de registradores permitindo uma troca de contexto mais suave que a dos processos convencionais.

YUV - Em televisão colorida refere-se a um sistema de três sinais: o sinal de luminância Y e dois sinais de crominância U e V. Sistema usado pelo padrão PAL.

REFERÊNCIAS BIBLIOGRÁFICAS

- [App96] Apple Computer Inc. **"QuickTime File Format Specification"**. Maio, 1996.
- [Bas92] Bastos, T. L. P.; Soares, L. F. G.. **"Análise de Algoritmos para Reprodução em Tempo Real de Voz em Redes de Pacotes"**. 10^o SBRC - Simpósio Brasileiro de Redes de Computadores, p. 445-463, Recife, Abril de 1992.
- [Ber94] Berners-Lee, Tim; Cailliau, Robert; Luotonen, Ari; Nielsen Henrik; Secret, Arthur. **"The World Wide Web"**. Communications of the ACM, no. 8, vol. 37 (1994), pp. 76-82.
- [Ber95] Berger, David. **"Video-on-Demand Metadata Query Interfaces"**. Tese de metrado, Universidade da Califórnia Berkeley, 1995.
- [Bou92] Le Boudec, J-Y.. **"The Asynchronous Transfer Mode: a Tutorial"**. Computer Network and ISDN Systems, n.24, p. 279-309, 1992.
- [Bro89] Brown, Heather. **"Standards for Structured Documents"**. The Computer Journal, no. 6, vol. 32, (1989), pp. 505-514.
- [Buf94] Buford, John; Rutledge, Lloyd; Keskin, C.. **"HyOctane: a HyTime Engine for an MMIS"**. Multimedia Systems, no. 4, vol. 1 (1994), pp. 173-185.
- [Buf94a] Buford, John. **"Uses of Multimedia Information in Multimedia Systems"**. Addison-Wesley, 1994.
- [Car95] Cardozo, E.; Magalhães, M. F.; Mendes, L.; Ricarte, I. L. M.. **"Sistemas Multimídia"**. FEE/Unicamp, 1995.
- [Cha94] Chang, Ed; Zakhor, A.. **"Scalable Video Data Placement on Parallel Disk Arrays"**. Proceedings of the SPIE Symposium on Storage and Retrieval for Image and Video Databases II. San Jose, California. Fevereiro de 1994, vpl. 2185 pp. 208-223.
- [Com95] Comer, D. E.. **"Internetworking Whit TCP/IP"**. Vol. 1, 3rd. Edition, Prentice-Hall, 1995.
- [Cou92] Coulson, G.; Blair, G. S.; Robin, P.. **"Micro-Kernel support for Continuous Media in Ditrributed Systems"**. Intrernal Report MPG-93-04; Lancaster University, Nov/1992.
- [Cou94] Couloris, G. F. ; Dollimore, J.. **"Distributed Systems"**. Addison-Wesley Publishing Company, 1994.
- [Cro95] Crowcroft, J.. **"Distributed Multimedia"**. Página HTML, Endereço: <http://www.cs.ucl.ac.uk/staff/jon/dummy/>, 1995.
- [Dea95] Dean, D.; Zippel, R.. **"Matching Data Storage To Application Needs"**. Operanting Systems Review, vol. 29, number 1. ACM Press, January, 1995.

- [Fed94] Federighi, C.; Rowe, L. A.. **"A Distributed Hierarchical Manager for a Video-On-Demand System"**. Symp. On Elec. Imaging Sci. & Tech., San Jose, C. A. , February 1994.
- [Flu95] Fluckiger, F.. **"Understanding Networked Multimedia : applications and technology"**. Editora Prentice Hall, 1995.
- [Fuj95] Fujitsu Open Systems Solutions; TechnoTeacher, Inc. **"HyTime Application Development Guide"**. Versão 1.2, Maio, 1995.
- [Fur95] Furht, B.. **"A Survey of Multimedia Compression Techniques and Standards - Part I : JPEG Standard"**. Editora Academic Press, Revista Real-Time Imaging, v. 1, n.1, p. 49-67, 1995.
- [Gad92] Gadegast, F.. **"MPEG I : Standard for Compressed Video and Audio"**. Página HTML, Endereço:
<http://cuiwww.unige.ch/OSG/MultimediaInfo/mmsurvey/standards.html>, 1992.
- [Gil91] Gilge, M.; Gusella, R.. **"Motion Video Coding for Packet-Switching Networks - An Integrated Approach"**. International Computer Science Institute, Dezembro de 1991.
- [Gri96] Griffioen, J.; Yavatkar, R.; Adams, R.. **"Automatic and Dynamic Identification of Metadata in Multimedia"**. 1st IEEE Metadata Conference, Silver Spring - Maryland, Abril de 1996.
- [Gru85] Gruber, J. G.. **"Subjective Effects of Variable Delay and Speech Loss Dinamically Managed Voice Systems"**. IEEE Transactions on Communications, v. COM-33, n. 8, p. 801-808, 1985.
- [Hac96] Hachisuca, A. M. M.. **"Desenvolvimento de um Sistema Servidor de Aplicações Multimídia"**. Mini-dissertação de Mestrado, ICMSC-USP, Abril de 1996.
- [Her94] Herwijnen, Eric van. **"Pratical SGML"**. Kluwer Academic Publishers, Massachusetts, 2^a ed., 1994.
- [ISO86] International Standards Organization. ISO/IEC IS 8879. **"Information Processing - Text and Office Systems - Standard Generalized Markup Language (SGML)"**. 1986.
- [ISO92] International Standards Organization. ISO/IEC IS 88744. **"Hypermedia/Time-Based Structuring Language (HyTime)"**. 1992.
- [Jos95] Josefh, R.; Rosengren, J. **"MHEG-5: An Overview"**. Página HTML, disponível pelo endereço:
<http://www.fokus.gmd.de/ovms/mug/archives/documents/mheg-reader/rd1206.html>
- [Lee94] Lee, E. K.; Chen, P. M.; Hartman, J. H.; Drapeau, A. L. C.; Miller, E. L.; Katz, R. H.; Gibson, G. A.; Patterson, D. A.. **"RAID-II: A Scalable Storage Architecture for High-Bandwidth Network File Service"**. 21st International Symposium on Computer Architcture, Chicago, Illinois, Abril, 1994.

- [Lit94] Little, T.D.C.; Venkatesh, D. **"Client Server Metadata Management for the Delivery of Movies in a Video-On-Demand System"**. 1st Intl. Workshop on Services in Distributed and Networked Environments, Prague, Czech Republic, June 1994.
- [Lou93] Lougher, P.; Shepherd, D. **"The Design of a Storage Server for Continuous Media"**. The Computer Journal, vol. 36, number 1, 1993.
- [Lut91] Luther, A. C.. **"Digital Video in the PC Environment"**. Segunda Edição. McGraw-hill, 1991.
- [Lut95] Luther, A. C.. **"Using Digital Video"**. AP Professional, 1995.
- [Mat94] Mattison, P.. **"Practical Digital Video with Programming Examples in C"**. Editora Wiley Professional Computing, 1994.
- [Mor95] Moreira, Edson dos S.; Nunes, Maria das Graças V.; Pimentel, Maria da Graça C.. **"Design Issues for a Distributed Hypermedia-based Tutoring System (HyDTS)"**. International Conference on Computer Application in Industry. Cairo, Egito, Dezembro, 1995.
- [Mul93] Mullender, Sape. **"Kernel Support for Distributed Systems in Distributed Systems"**. Second edition. Addison-Wesley, 1993.
- [Nah95] Nahrstedt, K.; Steinmetz, R. **"Resource Management in Networked Multimedia Systems"**. IEEE Computer, May 1995.
- [New91] Newcomb, Steven; Kipp, Neil; Newcomb, Victoria. **"The "HyTime" Hypermedia/Time-Based Document Structuring Language"**. Communications of the ACM, no. 11, vol. 34 (1991), 67-83.
- [Pan93] Pan, D. Y.. **"Digital Audio Compression"**. Digital Technical Journal, v. 5, n. 2, p. 1-14, 1993.
- [Pat93] Patel, Ketan; Smith, B. C.; Rowe, L. A.. **"Performance of a Software MPEG Video Decoder"**. Página HTML, Endereço: <http://mm-ftp.cs.berkeley.edu/pub/multimedia/papers/Mpeg93.ps.gz>, 1993.
- [Pim95] Pimentel, Maria G.C.. **"O HyTime no contexto do projeto SMmD"**. Anais do I Workshop em Sistemas Hipermedia Distribuídos, São Carlos, Julho, 1995, pp. 67-77.
- [Pim96] Pimentel, Maria G.C.; Trindade, Ciro C.; Leiva, Willie D.. **"Hyperlinking Segundo HyTime: Overview e Suporte"**. Anais do II Workshop em Sistemas Hipermedia. Fortaleza, Maio (1996), 1-10.
- [Ret95] Rettinger, L. A.. **"Desktop Video Conferencing : Technology and Use for Remote Seminar Delivery"**. Página Html, Endereço: http://www2.ncsu.edu/eos/service/ece/project/succeed_info/larettin/thesis/tit.html, 1995.
- [Row94] Rowe, L. A.; Boreczky, J. S.; Eads, C. A. **"Indexes for User Access to Large Video Databases"**. Symp. On Elec. Imaging Sci. & Tech., San Jose, C. A., February 1994.
- [Rut93] Rutledge, Lloyd. **"A HyTime Engine for Hypermedia Document Presentation"**. Dissertação de Mestrado, Universidade de Massachusetts Lowell, 1993.

- [Rut95] Rutledge, Lloyd; Buford, John F.; Rutledge, John L.. **"Applying HyTime to HTML"**. Proceedings of IASTED/ISMM Distributed Multimedia Systems and Applications, Agosto, 1995.
- [Sil95] Silberschatz, A.; Galvin, P. B.. **"Operating Systems Concepts"**. Addison-Wesley, 1995.
- [Sim96] Simon, Richard J.. **"Windows 95 Multimedia & ODBC API Bible"**. Book 3, Waite Group Press, 1996.
- [Sis96] Siscoutto, R. A.. **"Sistema de Videoconferência em Redes de Computadores"**. Mini-dissertação de Mestrado, ICMSC-USP, Abril de 1996.
- [Soa92] Soares, L. F. G. et. al.. **"Fundamentos de Sistemas Multimídia"**. VIII Escola Computação, Gramado, 1992.
- [Ste95] Steinmetz, R. **"Analyzing the Multimedia Operating System"**. IEEE Multimedia, p. 68-84, Spring, 1995.
- [Ste95a] Steinmetz, R.; Narstedt, K.. **"Multimedia : Computing, Communications and Applications"**. Prentice-Hall, 1995.
- [Stu95] Stüttgen, H. **"Network Evolution and Multimedia Communication"**. IEEE Multimedia, p.42-59, Fall, 1995.
- [Tan93] Tanenbaum, A. S.. **"Modern Operating Systems"**. Prentice-Hall, 1993
- [Taw93] Tawbi, W.; Horn, F.; Horlait, E.; Stefani, J. B. **"Video Compression Standards and Quality of Service"**. The Computer Journal, vol. 36, Number 1, 1993.
- [Tri96] Trindade, Ciro C.. **"Minimal Hyperlinking HyTime Documents: Autoria e Apresentação"**. Mini-dissertação de Mestrado, ICMSC-USP, Abril de 1996.