

Distributed Multimedia Systems

VICTOR O. K. LI, FELLOW, IEEE, AND WANJUN LIAO, STUDENT MEMBER, IEEE

A distributed multimedia system (DMS) is an integrated communication, computing, and information system that enables the processing, management, delivery, and presentation of synchronized multimedia information with quality-of-service guarantees. Multimedia information may include discrete media data, such as text, data, and images, and continuous media data, such as video and audio. Such a system enhances human communications by exploiting both visual and aural senses and provides the ultimate flexibility in work and entertainment, allowing one to collaborate with remote participants, view movies on demand, access on-line digital libraries from the desktop, and so forth. In this paper, we present a technical survey of a DMS. We give an overview of distributed multimedia systems, examine the fundamental concept of digital media, identify the applications, and survey the important enabling technologies.

Keywords—Digital media, distributed systems, hypermedia, interactive TV, multimedia, multimedia communications, multimedia systems, video conferencing.

NOMENCLATURE

AAL	ATM adaptation layer.
A/D	Analog to digital.
ADPCM	Adaptive differential pulse code modulation.
ADSL	Asynchronous digital subscriber loop.
ANSI	American National Standard Institute.
API	Application-independent program interface.
ATM	Asynchronous transfer mode.
B-ISDN	Broad-band integrated services digital network.
BLOB	Binary large object.
CATV	Community antenna TV.
CBR	Constant bit rate.
CCITT	Comité Consultatif Internationale de Télégraphique et Téléphonique.
CD-ROM	Compact disc-read only memory.
CGI	Common gateway interface.
CIF	Common intermediate format.
CPU	Central processing unit.
CS	Convergence sublayer.
D/A	Digital to analog.
DAB	Digital audio broadcast.
DBS	Direct broadcast satellite.
DCT	Discrete cosine transform.
DDL	Data definition language.
DHRM	Dexter hypertext reference model.
DML	Data manipulation language.
DMS	Distributed multimedia systems.
DPCM	Differential pulse code modulation.
DTD	Document type declaration.
DVC	Digital video cassette.
DVD	Digital video disc.
DVI	Digital video interactive.
EDF	Earliest deadline first.
ETSI	European Telecommunication Standard Institute.
FM	Frequency modulation.
GDSS	Group-decision support system.
GSS	Grouped sweeping scheduling.
HAM	Hypertext abstract machine.
HDM	Hypermedia design model.
HDTV	High-definition TV.
HFC	Hybrid fiber coax.
HTML	Hypertext markup language.
HTTP	Hypertext transport protocol.
IETF	Internet Engineering Task Force.
IMA	Interactive Multimedia Association.
ISDN	Integrated services digital network.
ISO	International Standard Organization.
ITU-T	International Telecommunications Union-Telecommunication.
ITV	Interactive television.
JPEG	Joint Photographic Experts Group.
LAN	Local area network.
LMDS	Local multipoint distribution system.

Manuscript received June 5, 1996; revised April 5, 1997. This work is supported in part by the U.S. Department of Defense Focused Research Initiative Program under Grant F49620-95-1-0531 and in part by the Pacific Bell External Technology Program.

V. O. K. Li was with the University of Southern California, Communication Sciences Institute, Department of Electrical Engineering, Los Angeles, CA 90089-2565 USA. He is now with the Department of Electrical and Electronic Engineering, University of Hong Kong, China.

W. Liao was with the University of Southern California, Communication Sciences Institute, Department of Electrical Engineering, Los Angeles, CA 90089-2565 USA. She is now with the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan.

Publisher Item Identifier S 0018-9219(97)05311-5.

MAN	Metropolitan area network.
MBone	Multicast backbone.
MHEG	Multimedia/Hypermedia Expert Group.
MIME	Multipurpose Internet mail extensions.
MMDS	Multichannel multipoint distribution system.
MPEG	Motion Pictures Experts Group.
MTP	Multicast transport protocol.
NTP	Network time protocol.
NTSC	National Television Systems Committee.
OAM&P	Operation, administration, maintenance, and provision.
ODA	Office document architecture.
OS	Operating system.
OSPF	Open shortest path first.
PAL	Phase alternation line.
PC	Personal computer.
PSTN	Public switched telephone network.
QCIF	Quarter-CIF.
QoS	Quality of service.
RCAP	Real-time channel administration protocol.
RSVP	Reservation protocol.
RTP	Real-time transport protocol.
RTCP	RTP control protocol.
SAR	Segmentation and reassembly.
SECAM	Sequential couleur avec memoire.
SCSI	Small computer system interface.
SGML	Standard generalized markup language.
SMDS	Switched multimegabit data service.
SMPT	Simple mail transfer protocol.
SONET	Synchronous optical network.
ST-II	Stream protocol version 2.
TCP/IP	Transport control protocol/Internet protocol.
TDMA	Time division multiple access.
UDP	User data protocol.
VBR	Variable bit rate.
VC	Virtual channel.
VCI	Virtual channel identifier.
VOD	Video on demand.
VP	Virtual path.
VPI	Virtual path identifier.
WAN	Wide area network.
WWW (W3)	World Wide Web.

I. INTRODUCTION

DMS's will revolutionize current life-styles, especially those aspects associated with human communications. Such systems will create an electronic world in which people are able to shop, work, or learn at home, watch video programs on demand, access on-line digital libraries from the desktop, and so forth. Technological advances in computers, high-speed networks, data compression, and consumer electronics—coupled with the availability of multimedia resources, mechanisms, and manipulation functions, the development of the relevant standards, and the convergency

of the computer, telecommunication, and cable television industries—are accelerating the realization of such systems.

A multimedia system is an interactive digital multimedia system. It seamlessly integrates multimedia information via computers and allows users to interact with such information according to their preferences. Multimedia information includes video, audio, and other continuous media in addition to discrete media such as text, data, and still images. Discrete media data are often displayed as one presentation unit, while continuous media data are composed of a series of consecutive units of equal presentation duration [1]. The ability to accommodate continuous as well as discrete media in an integrated system is the distinguishing feature of multimedia systems. A DMS augments stand-alone multimedia systems with a real-time network; real time here refers to timely delivery with acceptable quality. According to application needs, the system provides services in an interactive (e.g., VOD) or broadcasting (e.g., video conferencing) mode and in a timely (e.g., video browsing) or messaging (e.g., multimedia e-mail) manner. An example of a DMS is a number of multimedia PC's and/or workstations interconnected with continuous media servers via the Internet that allow users to retrieve, browse, and manipulate video or audio. Such networked multimedia systems not only dramatically enhance the existing CD-ROM-based multimedia applications but also encourage newly emerging broad-band applications at the expense of more complexity due to the requirement of QoS guarantees. These include constraints on bit error rates, packet loss probabilities, and delivery delays required in a traditional point-to-point information delivery system, plus additional constraints introduced due to the orchestration of distributed media in a DMS, such as the synchronization among multiple media streams from distributed sources to achieve a meaningful presentation. We formally define a DMS as an integrated communication, computing, and information system that enables the processing, management, delivery, and presentation of synchronized multimedia information with QoS guarantees.

Fig. 1 summarizes a DMS. The inputs of the system consist of the important factors that drive a DMS from concept to reality, and the outputs consist of a wide range of distributed multimedia applications. The system inputs can be divided into three orthogonal dimensions. The inputs from the left-hand side are the major contribution industries, including the computer, telecommunication, cable TV, entertainment, and consumer electronics industries. The inputs from the right-hand side are the important issues in the development of a DMS, including the technical, standardization, regulation, copyright, market, and social and human factors. The inputs from the top are a collection of the enabling technologies of the information subsystem (for storage), the communication subsystem (for transmission), and the computing subsystem (for processing). The information subsystem consists of the multimedia servers, information archives, and multimedia database systems. It stores and retrieves multimedia information, serves a large

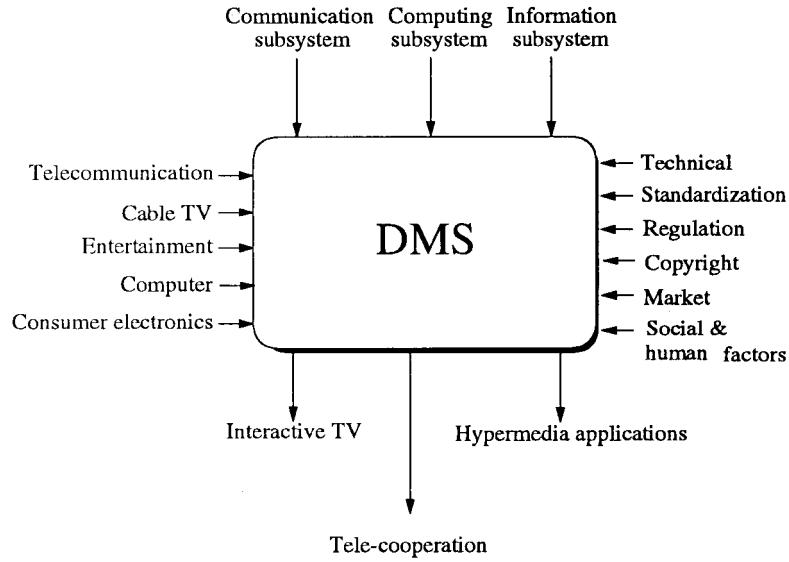


Fig. 1. A distributed multimedia system.

number of simultaneous user requests with QoS guarantees, and manages the data for consistency, security, and reliability. The communication subsystem consists of the transmission medium and transport protocols. It connects the users with distributed multimedia resources and delivers multimedia materials with QoS guarantees, such as real-time delivery for video or audio data and error-free delivery for text data. The computing subsystem consists of a multimedia platform (ranging from a high-end graphics workstation to a multimedia PC equipped with a CD-ROM drive, speaker, sound card, and video card), OS, presentation and authoring¹ tools, and multimedia manipulation software. It allows users to manipulate the multimedia data. The outputs of the system can be broadly classified² into three different types of distributed multimedia applications: ITV, telecooperation, and hypermedia. ITV allows subscribers to access video programs at the time they choose and to interact with them. Services include home shopping, interactive video games,³ financial transactions, VOD, news on demand, and CD on demand. Telecooperation overcomes time and location restrictions and allows remote participants to join a group activity. Services include remote learning, telecommuting, teleservicing, teleoperation, multimedia e-mail, videophone, desktop conferencing, electronic meeting rooms, joint editing, and group drawing. A hypermedia document is a multimedia document with “links” to other multimedia documents that allows users to browse multimedia information in a nonsequential manner. Services include digital libraries, electronic encyclopedias, multimedia magazines, multimedia documents, information

¹An authoring tool is a specialized software that allows a producer or designer to design and assemble multimedia elements for a multimedia presentation, either through scripting languages (such as Script-X, Perl Script, and JavaScript) or menu- and icon-based graphical interfaces (such as Director, Hypercard, and Apple Media Tool).

²Our criteria for classification is described in Section III.

³Interactive video games can also be classified as a hypermedia application.

kiosks, computer-aided learning tools, and WWW (or W3) surfing.

The main features of a DMS are summarized as follows [2]–[10].

- 1) *Technology integration*: integrates information, communication, and computing systems to form a unified digital processing environment.
- 2) *Multimedia integration*: accommodates discrete data as well as continuous data in an integrated environment.
- 3) *Real-time performance*: requires the storage systems, processing systems, and transmission systems to have real-time performance. Hence, huge storage volume, high I/O rate, high network transmission rate, and high CPU processing rate are required.
- 4) *System-wide QoS support*: supports diverse QoS requirements on an end-to-end basis along the data path from the sender, through the transport network, and to the receiver.
- 5) *Interactivity*: requires duplex communication between the user and the system and allows each user to control the information.
- 6) *Multimedia synchronization support*: preserves the playback continuity of media frames within a single continuous media stream and the temporal relationships among multiple related data objects.
- 7) *Standardization support*: allows interoperability despite heterogeneity in the information content, presentation format, user interfaces, network protocols, and consumer electronics.

The rest of this paper is organized as follows. Section II overviews digital media fundamentals. Section III presents three important types of distributed multimedia applications, including ITV, telecooperation, and hypermedia

systems. Section IV addresses the technical aspects of the information, communication, and computing subsystems. Section V presents our conclusions.

II. DIGITAL MEDIA FUNDAMENTALS

Just as the printing press, telegraph, telephone, and mass media have revolutionized human communications, the emerging digital media are expected to spawn another revolution in communications.

Electronic media and transducers are essential parts of multimedia presentation. A medium is something through which information travels. For example, natural media can be air, water, and space. Information travels in natural media in the form of waves, such as light waves and sound waves. Electronic media are distinguished from natural media in their abilities to record, duplicate, manipulate, and synthesize information [11]. Examples of electronic media include computer disks, CD-ROM's, videotapes, and cables. The transducer is a device capable of changing signals from one form to another. Microphones and video cameras are examples of transducers that convert sound waves and light intensity, respectively, into electrical signals. Conversely, loudspeakers and monitors convert electrical signals into sound and light, respectively.

Information can be represented as analog signals or digital signals. Analog signals exhibit a continuous tone of smooth fluctuations, while digital signals are composed of discrete values represented by numbers. The transformation between analog and digital signals is achieved by A/D conversions and D/A conversions. A/D conversion transforms an analog input (such as voltage) into a series of numeric representation by digitization. Digitization is composed of sampling and quantizing. Sampling takes snapshots of the analog waveform at certain points in time, and the sampling rate determines how often the analog signal is digitized. According to the Nyquist theorem, the sampling rate must be at least twice the highest frequency component of the analog waveform to reproduce the signal optimally [12]. Quantization determines the digital measured value of the analog waveform at the sampling time. The larger the range of numbers used for the quantization, the more gradations of the digitally sampled waveform can be represented. Fig. 2 illustrates sampling and quantization. In Fig. 2(a), the analog waveform is sampled at periodic points in time. The sampled values are then represented by the closest quantization levels. The difference between the value of the closest quantization level and the real analog value is known as the quantization error, or quantization noise. In Fig. 2(b), a 2-b binary number is used to represent the quantization level, resulting in four possible levels. In Fig. 2(c), three bits are used, allowing eight quantization levels. As expected, quantization error increases as the number of bits used to represent a sample decreases. D/A conversion is the reverse process of A/D conversion. It turns a sequence of discrete numbers back into the continuous analog signal. Fig. 3 shows the role of A/D and D/A

converters in the acquisition and playback of digital media systems.

Digital media take advantage of advances in computer-processing techniques and inherit their strengths from digital signals. The distinguishing features that make them superior to the analog media are listed as follows.

- 1) *Robustness*: the quality of digital media will not degrade as copies are made. They are more stable and more immune to the noises and errors that occur during processing and transmission. Analog signals, on the other hand, suffer from signal path attenuation and generation loss (as copies are made) and are influenced by the characteristics of the medium itself.
- 2) *Seamless integration*: the integration of different media via digital storage, processing, and transmission technologies, regardless of the particular media properties. Therefore, digital media eliminate device dependency in an integrated environment and allow easy data composition and nonlinear editing.
- 3) *Reusability and interchangeability*: with the development of standards for the common exchange formats, digital media have greater potential to be reused and shared by multiple users.
- 4) *Ease of distribution potential*: thousands of copies may be distributed electronically by a simple command.

A. Digital Image

Digital images are captured directly by a digital camera or indirectly by scanning a photograph with a scanner. They are usually compressed before being stored or transmitted and are displayed on the screen or printed.

1) *Image Display*: Digital images are composed of a collection of pixels that are arranged as a two-dimensional matrix. This two-dimensional (or spatial) representation is called the image resolution. As shown in Fig. 4, each pixel consists of three components: the red (*R*), the green (*G*), and the blue (*B*). On a screen, each component of a pixel corresponds to a phosphor. A phosphor glows when excited by an electron gun. Various combinations of different *RGB* intensities produce different colors. The number of bits to represent a pixel is called the color depth, which decides the actual number of colors available to represent a pixel. Color depth in turn is determined by the size of the video buffer in the display circuitry. For example, a screen with a 640-by-480 resolution and with 24-b color (i.e., eight bits each for the *R*, *G*, and *B* components) displays 640 pixels horizontally and 480 vertically, and each pixel can be represented by selecting from more than 2^{24} ($\simeq 16.7$ million) different colors.

2) *Digital Image Data*: The resolution and color depth determine the presentation quality and the size of image storage. The more pixels and the more colors, the better the quality and the larger the volume. For example, a 640-by-480 image with 24-b color requires $640 \times 480 \times 24 \text{ b} = 7.4 \text{ Mb}$ of storage. To reduce the storage requirement,

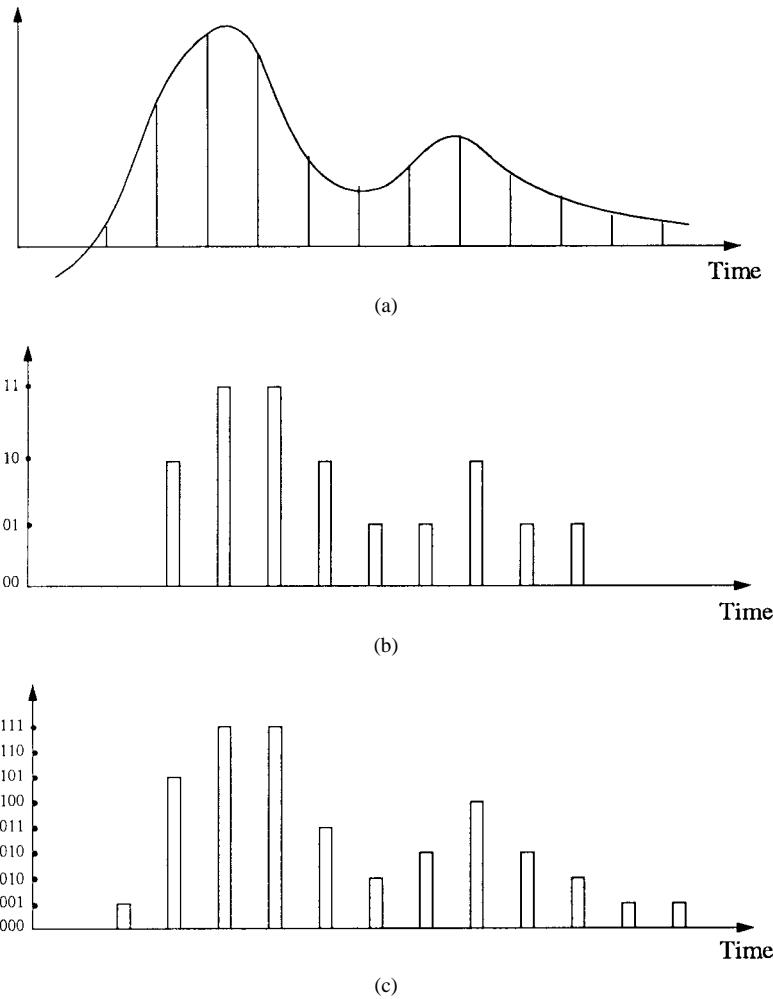


Fig. 2. (a) Sampling. (b) Quantization with four levels. (c) Quantization with eight levels.

three different approaches can be used: indexed color, color subsampling, and spatial reduction.

a) *Indexed color*: This approach reduces the storage size by using a limited number of bits with a color look-up table (or color palette) to represent a pixel. For example, a 256-color palette would limit each pixel value to an 8-b number. The value associated with the pixel, instead of direct color information, represents the index of the color table. This approach allows each of the 256 entries in the color table to reference a color of higher depth, a 12-b, 16-b, or 24-b value depending on the system, but limits the maximum number of different colors that can be displayed simultaneously to 256. Dithering, a technique taking advantage of the fact that the human brain perceives the median color when two different colors are adjacent to one another, can be applied to create additional colors by blending colors from the palette. Thus, with palette optimization and color dithering, the range of the overall color available is still considerable, and the storage size is reduced.

b) *Color subsampling*: Humans perceive color as the brightness, the hue, and the saturation rather than as *RGB* components. Human vision is more sensitive to variation

in the luminance (or brightness) than in the chrominance (or color difference). To take advantage of such deficiencies in the human eye, light can be separated into the luminance and chrominance components instead of the *RGB* components. For example, the *YUV* color space separates luminance from chrominance via the following linear transformation from the *RGB* system

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.30 & 0.59 & 0.11 \\ 0.62 & -0.52 & -0.10 \\ -0.15 & -0.29 & 0.44 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

where *Y* is the luminance component and (U, V) are the chrominance components.

The color subsampling approach shrinks the file size by downsampling the chrominance component (i.e., using less bits to represent the chrominance component) while leaving the luminance component unchanged. There are two different approaches to compressing color information.

- 1) Use less, say, half the number of bits used for *Y*, to represent *U* and *V* each. For example, 24-b color uses eight bits each for the *R*, *G*, and *B* components.

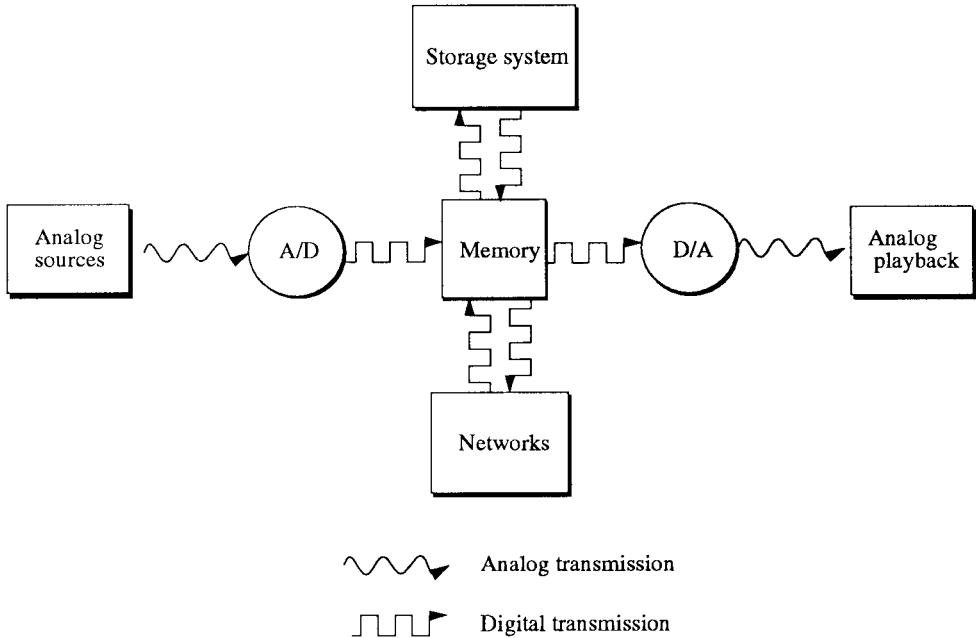


Fig. 3. A/D and D/A converters.

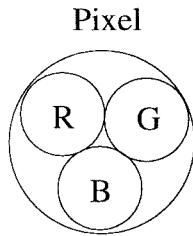


Fig. 4. A pixel composition.

When applying color subsampling, eight bits are used for Y and four bits each are used for U and V .

- 2) Alternately drop either chrominance component (i.e., $YUYVYUYVYU \dots$). For example, process the component of U while discarding the component of V , then process the component of V while discarding U and alternately discarding the U and V components. Each 24-b sample can thus be reduced to 16 b: eight bits for Y and eight bits for either U or V .

c) *Spatial reduction*: This approach, also known as data compression, reduces the size by throwing away the spatial redundancy within the images. Data compression will be described in detail in Section IV-B.

B. Digital Video

Video⁴ is composed of a series of still image frames and produces the illusion of movement by quickly displaying frames one after another. The human vision system accepts

⁴Television and video are usually distinguished. Television is often associated with the concept of broadcast or cable delivery of programs, whereas video allows more user interactivity, such as recording, editing, and viewing at a user-selected time. In this paper, we consider video with the broader meaning of motion pictures, including television.

anything above 20 frames per second (fps) as smooth motion.

1) *Video Display*: To paint an entire video image on a monitor, an electron gun illuminates every pixel as it sweeps across the screen from left to right and from top to bottom. This is called raster scanning. A complete line traveled by the electron gun from left to right is called a scanline. If the electron beams travel the scanlines sequentially from top to bottom, as shown in Fig. 5(a), it is called a progressive scan. If the electron beams travel the odd scanlines in one pass and the even scanlines in the next, it is called an interlaced scan, as shown in Fig. 5(b). For the interlaced scan, each pass is called a field, and two complete fields comprise a full display screen. When the electron beam travels to the bottom scanline, it retracts back to the top. The time of moving the electron gun from the bottom to the top is called the vertical blanking interval, while the time from the right of one scanline to the left of the next is the horizontal blanking interval. The vertical blanking intervals are often used to embed other signals, such as synchronization information, in broadcast television (TV) systems.

Two major technologies are adopted for video display: TV and computer monitors.

a) *TV monitor*: The input signals to TV are composite (to be described below). The TV monitor is an interlaced monitor. The refresh rate is 60 fields per second (or 30 fps).⁵ It uses the persistence of human vision as well as the persistence of the phosphors to reduce the refresh rate. Thus, a TV monitor costs less but flickers often due to the lower refresh rate. Video images on a TV monitor are overscanned, with the image area extending beyond

⁵The actual refresh rate depends on the type of TV system. For example, the rate for the NTSC TV system is 29.97 fps.

Table 1 Comparison of the Three Analog TV Signal Standards

Format	Scanlines	Interlaced frame rate (fps)	Adopted in
NTSC	525	29.97	US, Japan, and some Asian countries
PAL	625	25	Western Europe except France
SECAM	625	25	France, Eastern Europe, Russia

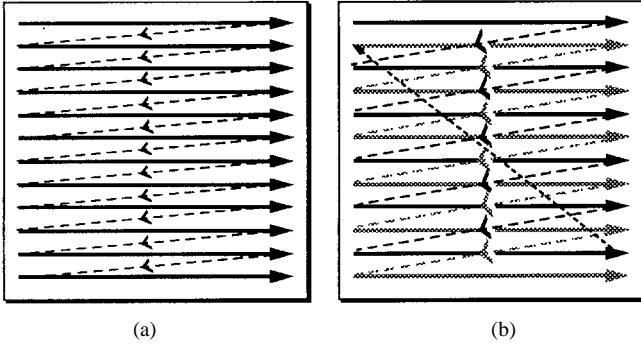


Fig. 5. (a) Progressive scan. (b) Interlaced scan.

the physical viewing area of the monitor. The aspect ratio of TV, which is the ratio of the horizontal to the vertical dimensions, is 4:3.

b) Computer monitor: The input signals to a computer monitor are *RGB* components. The computer monitor is a noninterlaced monitor. The refresh rate is 60 fps. The video images are underscanned on the computer monitor, with the image area smaller than the physical viewing area, and therefore there is a black border around the image. The aspect ratio of a computer monitor is also 4:3.

2) Analog Versus Digital Video Signals: For both analog and digital videos, there are two different types of video signals for storage and transmission.

- 1) A composite signal combines the luminance *Y* and chrominance *C* into a single stream to be stored and transmitted. Examples of analog composites include broadcast TV signals and VHS; examples of digital composites include the D-2 and D-3 formats.
- 2) A component signal separates the video into luminance *Y* and chrominance *C* streams. There are two variations of component signals: the *Y/C* component and the *RGB* component. The *Y/C* component uses two separate channels to transport *Y* and *C* information, whereas the *RGB* component uses three channels to carry *R*, *G*, and *B* colors. Component video signals have better quality than composite video. Examples of analog components include Hi8 (*Y/C* component) and Beta SP (*RGB* component), and those of digital components include the D-1, D-4, and D-5 formats.

a) Analog signal: The video waveform actually represents three signals at once: luminance, chrominance, and synchronization information. Three different types of analog video are used for TV broadcast [13], [14].

- 1) *NTSC:* provides an interlaced frame rate (or display rate) of 29.97 fps with 525 lines, where 41 lines are reserved for the vertical blanking interval to yield 484 lines of visual resolution. The luminance signal is called the *Y* signal. The chrominance signal is amplitude modulated onto the color subcarrier at two different phases: the *I* (in-phase) value at 0° and the *Q* (quadrature) value at 90°.
- 2) *PAL:* offers an interlaced frame rate of 25 fps with 625 scanlines. It provides more bandwidth for chrominance modulation, yielding better color resolution. The composite PAL signal operates on the *YUV* color space, where *Y* is the luminance component and (*U*, *V*) are the chrominance components.
- 3) *SECAM:* uses the same resolution and interlaced frame rate as PAL but uses FM to encode the chrominance signal. It also has better color resolution than NTSC.

The three video signals use the same bandwidth but a different number of scanlines and different refresh rates. Compared to NTSC, PAL and SECAM have better resolution but lower refresh rates and hence more flicker. Table 1 shows the comparison between these three video formats.

b) Digital signal: Digital video inherits the advantages of digital signals as described earlier. The biggest challenges posed by digital video are the massive volume of data involved and the need to meet the real-time constraints on retrieval, delivery, and display. The solution entails the following.

- 1) *Compromise in the presentation quality:* instead of video with full frame (i.e., images filling the complete screen), full fidelity (i.e., millions of colors at screen resolution), and full motion (i.e., ≥ 30 fps), one may reduce the image size, use less bits to represent colors, or reduce the frame rate. (For a frame rate less than 16 fps, the illusion of motion is replaced by the perception of a series of frames.)
- 2) *Video compression:* To reduce the massive volume of digital video data, compression techniques with high compression ratios are required. In addition to throwing away the spatial and color similarities of individual images, the temporal redundancies between adjacent video frames are eliminated.

C. Digital Audio

1) Audio Perception: Sound waves generate air pressure oscillations that stimulate the human auditory system. The human ear is an example of a transducer. It transforms sound waves to signals recognizable by brain neurons. As

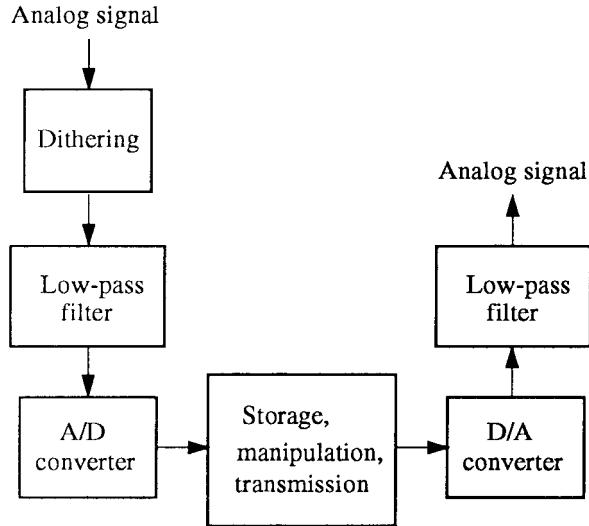


Fig. 6. The block diagram of digital audio processing.

with other audio transducers, two important considerations are frequency response and dynamic range.

- 1) *Frequency response*: refers to the range of frequencies that a medium can reproduce accurately. The frequency range of human hearing is between 20 Hz and 20 kHz, and humans are especially sensitive to low frequencies of 3–4 kHz.
- 2) *Dynamic range*: describes the spectrum of the softest to the loudest sound-amplitude levels that a medium can reproduce. Human hearing can accommodate a dynamic range over a factor of millions, and sound amplitudes are perceived in logarithmic ratio rather than linearly. To compare two signals with amplitudes A and B , the unit of decibel (dB) is used, defined as $20 \log(A/B)$ dB. Humans perceive sounds over the entire range of 120 dB, the upper limit of which will be painful to humans.

Sound waves are characterized in terms of frequency (Hz), amplitude (dB), and phase (degree), whereas frequencies and amplitudes are perceived as pitch and loudness, respectively. Pure tone is a sine wave. Sound waves are additive, and sounds in general are represented by a sum of sine waves. Phase refers to the relative delay between two waveforms, and distortion can result from phase shifts.

2) *Digital Audio Data*: Digital audio systems are designed to make use of the range of human hearing. The frequency response of a digital audio system is determined by the sampling rate, which in turn is determined by the Nyquist theorem. For example, the sampling rate of CD-quality audio is 44.1 kHz (which can accommodate the highest frequency of human hearing, namely, 20 kHz), while telephone-quality sound adopts an 8-kHz sampling rate (which can accommodate the most sensitive frequency of human hearing, up to 4 kHz). Digital audio aliasing is introduced when one attempts to record frequencies that exceed half the sampling rate. A solution is to use a low-pass filter to eliminate frequencies higher than the

Nyquist rate. The quantization interval, or the difference in value between two adjacent quantization levels, is a function of the number of bits per sample and determines the dynamic range. One bit yields 6 dB of dynamic range. For example, 16-b audio contributes 96 dB of the dynamic range found in CD-grade audio, nearly the dynamic range of human hearing. The quantized samples can be encoded in various formats, such as pulse code modulation (PCM), to be stored or transmitted. Quantization noise occurs when the bit number is too small. Dithering,⁶ which adds white noise to the input analog signals, may be used to reduce quantization noise [12]. In addition, a low-pass filter can be employed prior to the D/A stage to smooth the stair-step effect resulting from the combination of a low sampling rate and quantization. Fig. 6 summarizes the basic steps to process digital audio signals.

The quality of digital audio is characterized by the sampling rate, the quantization interval, and the number of channels (e.g., one channel for mono systems, two for stereo systems, and 5.1 for Dolby surround-sound systems). The higher the sampling rate, the more bits per sample, and the more channels, the higher the quality of the digital audio and the higher the storage and bandwidth requirements. For example, a 44.1-kHz sampling rate, 16-b quantization, and stereo audio reception produce CD-quality audio but require a bandwidth of $44.1 \times 16 \times 2 = 1.4$ Mb/s. Telephone-quality audio, with a sampling rate of 8 kHz, 8-b quantization, and mono audio reception, needs only a data throughput of $8000 \times 8 \times 1 = 64$ Kb/s. Again, digital audio compression or a compromise in quality can be applied to reduce the file size.

III. DISTRIBUTED MULTIMEDIA APPLICATIONS

Multimedia integration and real-time networking create a wide range of opportunities for multimedia applications. According to the different requirements imposed upon the

⁶Note that dithering here is different from the concept of dithering described in Section II-A.

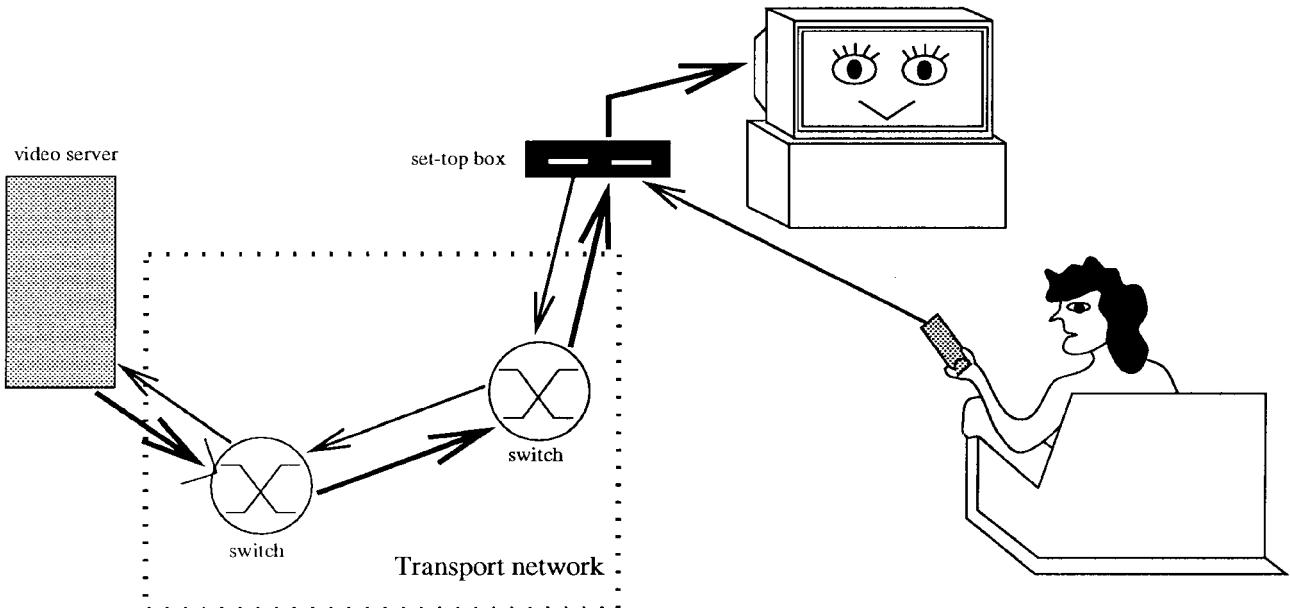


Fig. 7. A VOD system.

information, communication, and computing subsystems, distributed multimedia applications may be broadly classified into three types: ITV, telecooperation, and hypermedia. ITV requires a very high transmission rate and stringent QoS guarantees. It is therefore difficult to provide such broad-band services over a low-speed network, such as the current Internet, due to its low bandwidth and best-effort-only services. ITV typically demands point-to-point, switched connections as well as good customer services and excellent management for information sharing, billing, and security. Bandwidth requirement is asymmetric in that the bandwidth of the downstream channel that carries video programs from the server to the user is much higher than that of the upstream channel from the user to the server. In addition, the equipment on the customer's premises, such as a TV with a box on top, is much less powerful than the equipment used in telecooperation, such as video conferencing. Telecooperation requires multicast, multipoint, and multiservice network support for group distribution. In contrast to the strict requirement on the video quality of ITV, telecooperation, such as videophone and desktop conferencing, allows lower picture quality and therefore has a lower bandwidth requirement. It is possible to provide such services with the development of real-time transport protocols over the Internet. As with hypermedia applications, telecooperation requires powerful multimedia database systems rather than just continuous media servers, with the support of visual query and content-based indexing and retrieval. Hypermedia applications are retrieval services and require point- or multipoint-to-point and switched services. They also require friendly user interfaces and powerful authoring and presentation tools.

A. Interactive TV

An important ITV service is VOD. VOD provides electronic video-rental services over the broad-band network

[15]. Fig. 7 shows an example of a VOD system. Customers are allowed to select programs from remote massive video archives, view at the time they wish without leaving the comfort of their homes, and interact with the programs via VCR-like functions, such as fast-forward and rewind [16]–[22]. A VOD system that satisfies all of these requirements (any video, any time, any VCR-like user interaction) is called a true VOD system and is said to provide true VOD services; otherwise, it is called a near VOD system.

To be commercially viable, VOD service must be priced competitively with existing video rental services (the local video rental store). One way to allow true VOD services is to have a dedicated video stream for each customer. This is not only expensive but is wasteful of system resources because if multiple users are viewing the same video, the system has to deliver multiple identical copies at the same time. To reduce this cost, batching may be used. This allows multiple users accessing the same video to be served by the same video stream. Batching increases the system capability in terms of the number of customers the system can support. However, it does complicate the provision of user interactions. In staggered VOD, for example, multiple identical copies (streams) of the same video program are broadcast at staggered times (e.g., every five minutes). A user will be served by one of the streams, and user interactions are simulated by jumping to a different stream. Not all user interactions can be simulated in this fashion, however, and even for those that can be simulated, the effect is not exactly what the user requires. For example, one cannot issue fast-forward because there is no stream in the fast-forward mode. Also, one may pause for 5, 10, 15, etc. minutes but not for seven minutes because there is no stream that is offset in time by seven minutes from the original stream. An innovative protocol called the Split-and-Merge (SAM) [23] has been proposed to provide true VOD services while fully exploiting the benefit

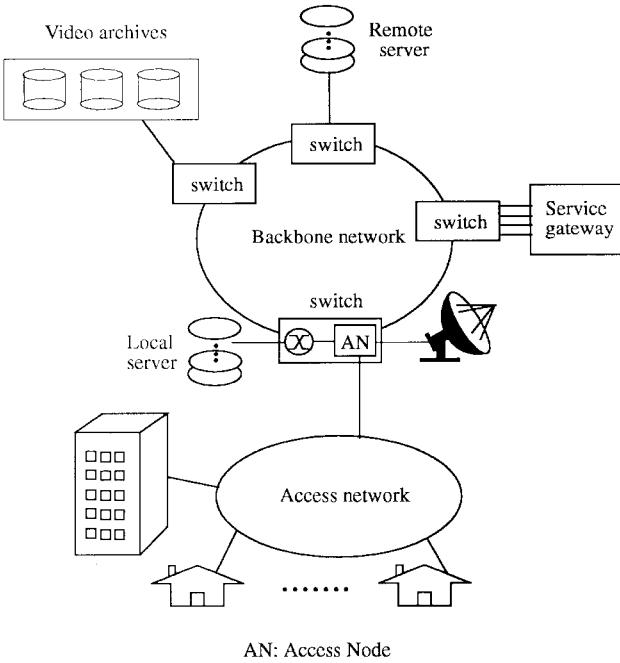


Fig. 8. VOD system architecture.

of batching, thereby reducing the per-user video delivery cost.

1) VOD Infrastructure: To implement the complete set of interactive services, a VOD system contains many components, including the video server, transport network, subscriber terminal unit (STU), and service gateway. Fig. 8 shows the typical architecture of a VOD system and Fig. 9 indicates the relationships among the components.

a) Video server: A video server, as shown in Fig. 10, consists of massive storage and media controllers. It stores a large number of digitized videos and serves a considerable number of simultaneous video requests to the same or to different videos on demand. The networked video jukebox [24] and the video library system [25] are two examples of video-server prototypes.

The storage media usually consist of magnetic disks, optical disks, and magnetic tape and are usually organized hierarchically for cost effectiveness. Under this configuration, popular videos are stored in the disks. Less popular ones are stored in tape devices with terabyte capacity and retrieved as necessary to the disk drive for processing. The video server may be located at the local or regional switch of the network provider or at remote information archives.

The basic functions supported by video servers include request handling, random access, and user interactions, in addition to admission control and QoS guarantees.

b) Transport network: The transport network delivers video programs from the video server to the customers. The network must have a very high data rate and satisfy the real-time delivery constraints of video traffic. It consists of two major components: the backbone network with high-speed switches and the local access network. The backbone network links remote video servers at geograph-

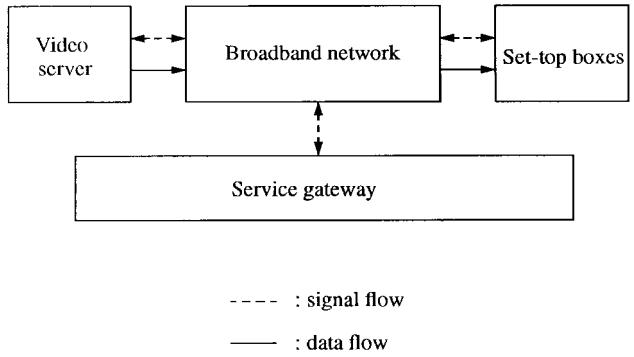


Fig. 9. The relationships between system components.

ically dispersed locations and the regional, national, or international information archives. The trend is toward a SONET backbone with ATM switching because of their low error rate, high data-transfer rate, bandwidth on demand, and seamless services.

The access nodes are located at the local switches and glue the backbone network to the access networks. Depending on the system implementation, an access node may be the head end in CATV networks, the central office in telephone networks, or the base station in mobile systems [2]. The access node may be equipped with satellite dishes to receive analog broadcast TV programs, have a local media server to store digital programs, and serve as an interface to the backbone network to access information from remote video servers. The major functions performed by access nodes include modulating (for passband systems) or multiplexing (for baseband systems) incoming signals to the desired outgoing formats, and switching and routing the streams for downstream transmissions. The access networks deliver video programs from the access node for the last few miles to the customers. This part of the network infrastructure demands the most investment from the service providers for the commercial deployment of services. Currently, one of the most active debates in residential broad-band deployment is the choice of access network architectures. Due to their different existing network infrastructures and operation policies, as well as other historical reasons, the telephone and cable TV industries propose different technologies. This is the so-called last-mile debate. Five alternatives of the access networks are presented below. The first three are wired approaches and the last two are wireless.

i) HFC: The HFC network (see Fig. 11) is proposed by the cable TV industry. It attempts to minimize initial investment by taking advantage of the existing CATV cable plants. HFC combines high-bandwidth, low-loss fiber optics with low-cost coaxial cables to provide more channels with better picture quality to the customers [26]. Fiber trunks are used from the head end to fiber nodes, where optoelectronic conversion is performed. Coaxial drops are used to connect to the homes in the neighborhood, similar to the traditional tree-and-branch structure except with shorter coaxial runs, fewer amplifiers, and lower impairment to

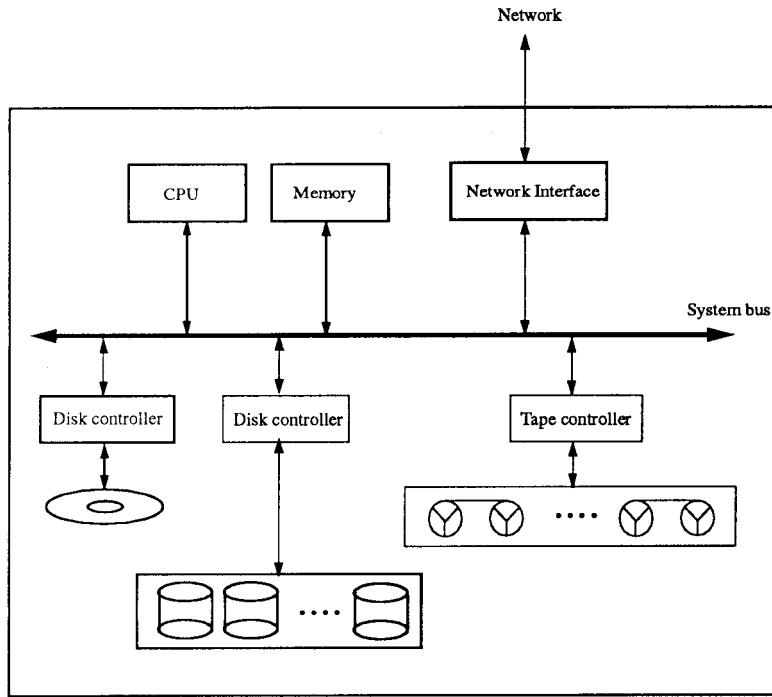


Fig. 10. Video server architecture.

picture quality. Each fiber node serves 500–2000 homes in the neighborhood clusters. The Time-Warner VOD field trial in Orlando, FL, uses an HFC architecture [27].

Fig. 12 shows a typical spectrum allocation⁷ of HFC [28], [29]. The downstream (server-to-customer) frequencies of 50–550 MHz are allocated for analog TV broadcast and 550–750 MHz for digital interactive services and telephone services. The upstream (customer-to-server) frequencies of 5–45 MHz are allocated for return messages and upstream telephone and data services. Since the upstream channels are shared by subscribers, multiple-access protocols, such as slotted ALOHA, TDMA, or polling, are required to resolve service contentions.

ii) ADSL: ADSL takes advantage of the high market penetration of telephone installation and its switched, duplex services. This technology places ADSL modems at the end of the copper loop to create a high-speed access link on existing telephone lines. It dramatically increases the data rate over standard twisted-pair copper cables [30], [31]. The Hong Kong Telecom Interactive TV services employ an ADSL architecture [32]. Fig. 13 shows a typical architecture of ADSL networks.

“Asymmetric” refers to the fact that the downstream bandwidth is much higher than the upstream one. There are different versions of ADSL. With ADSL-1, the data rate is up to 1.5 Mb/s downstream, enough to accommodate MPEG-1 (or VHS picture quality) video transmissions, and 16 Kb/s upstream for interactive control; with ADSL-2 and ADSL-3, the downstream rates can be up to 3 and 6 Mb/s, respectively, and the upstream can be 64 Kb/s.

⁷Note that the actual spectrum allocation is the choice of the service provider.

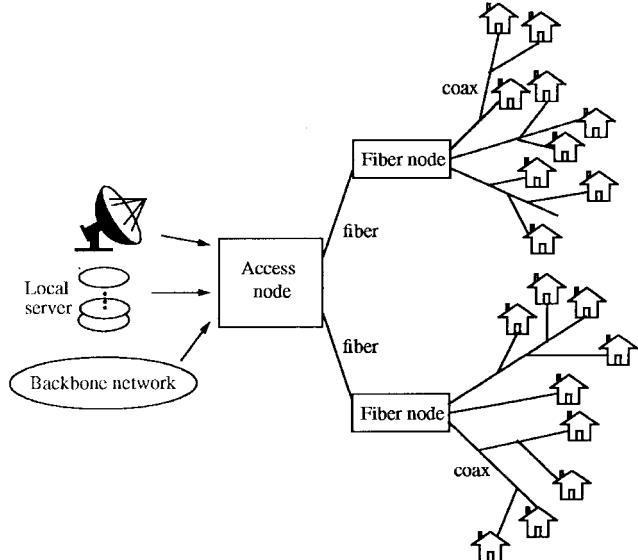


Fig. 11. HFC network architecture.

A variation called a very high bit rate digital subscriber line (VDSL) is planned to operate downstream at the OC1 (51.84 Mb/s) data rate. The inherently low bandwidth of twisted-pair cables, however, is the major obstacle for long-term deployment.

iii) Fiber to the curb (FTTC): FTTC takes advantage of the extremely high bandwidth of fiber optics and switched digital services. This technology uses a fiber-optical connection from the telephone central office to the optical network units (ONU), e.g., at the curbside or at the shared equipment within a building. Then, twisted-

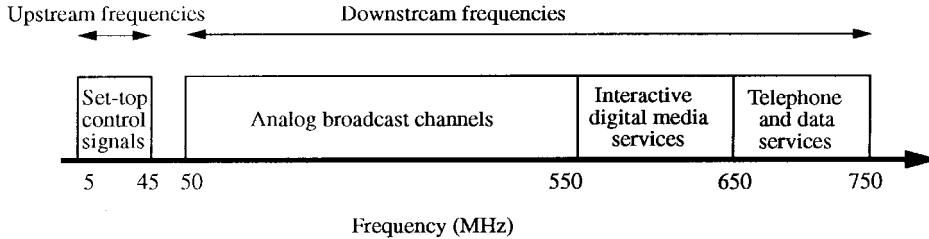


Fig. 12. A typical spectrum allocation for a multiservice cable network.

pair or coaxial cables are used for the final connection to the customer premises [33], as illustrated in Fig. 14. An ONU typically supports 8–24 homes and is located somewhere between several tens of and a few hundred meters from the customer premises. A variation called fiber to the home (FTTH) provides point-to-point fiber connection from the central office directly to homes. The high cost and immature all-optical technologies, however, have delayed the deployment of FTTH.

iv) *Wireless cable*: The term wireless cable is a misnomer. It refers to the broadcast of cable-TV-type programming directly to receivers on the outside of customer premises via microwave transmissions. Some telephone companies, e.g., Pacific Telesis, see this as a quick way to deliver video services to their customers. A large number of channels will be available, and some of the channels may be used to offer near VOD or true VOD services. Two technologies are available: MMDS and LMDS. MMDS operates around 2.5 GHz with a total bandwidth of 200 MHz (equivalent to 33 analog 6-MHz channels) and has been used by schools and archdiocese for about 20 years. With digitization, however, the bandwidth may be used to transport up to 150 digital channels. LMDS is similar but operates around 28 GHz with a much wider bandwidth of 1.3 GHz. Therefore, much more channels will be available.

v) *DBS*: In a DBS system, e.g., DirecTV, digital video and audio are sent from ground stations to geostationary satellites 23 000 miles above the earth's surface. Each satellite has many transponders. Each transponder receives a video program on one frequency and retransmits it on another. Each user has a small antenna dish (18 inches in diameter for DirecTV) that is pointed in the direction of one of the satellites and receives video signals directly from it. As in wireless cable, a large number of channels will be available, and some of the channels may be used to offer near VOD or true VOD services.

Note that these access networks may be used to offer high-speed Internet access to the customers. Due to the strong demand for Internet services, some service providers have postponed the deployment of VOD services in order to focus on Internet services. It is expected that as more customers acquire high-speed Internet connections, the high-speed network infrastructure required for VOD services will be developed automatically.

c) *STU*: An STU (or set-top box), along with the television monitor and the infrared controller (i.e., remote control), serves as the bridge between the subscribers

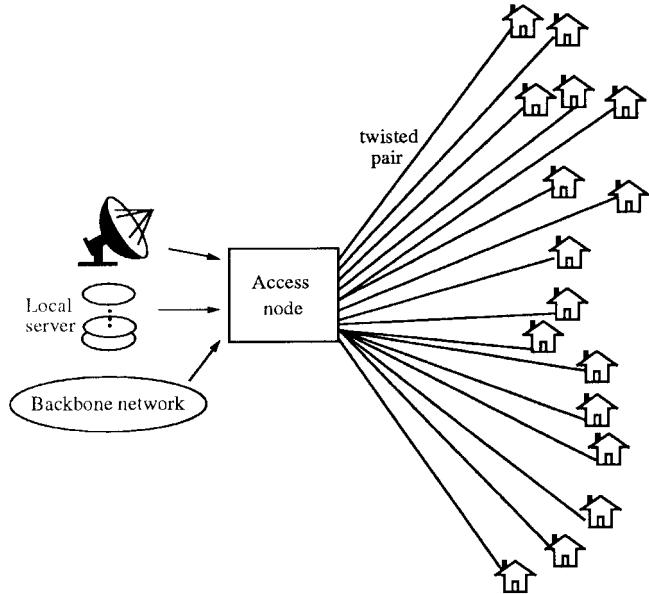


Fig. 13. ADSL network architecture.

and the system. With on-screen images and cursor-like devices, viewers are connected to the video servers and can browse the video selections and interact with the services via VCR-like functions [16]. The major functions of STU include receiving the incoming video streams; demodulating, demultiplexing, and decoding the signals; performing the necessary signal conversion, such as D/A transformation for playback on the TV monitor; and sending outgoing control messages.

STU must accommodate the heterogeneity of technologies and forms from various access networks and service providers and provide interactive controls to services [16], [34]. The ability of an STU to adapt to the diversity of access networks, service providers, applications, and user interfaces distinguishes it from the set-top box currently used in franchised CATV, which is dedicated to one cable company.

d) *Service gateway*: This component may be integrated with an access node or may be a separate element in the network. The main functions performed by the service gateway include [35]:

- directory services to provide menu browsing and program scheduling;
- mapping from service identity (ID) to corresponding location and program provider;

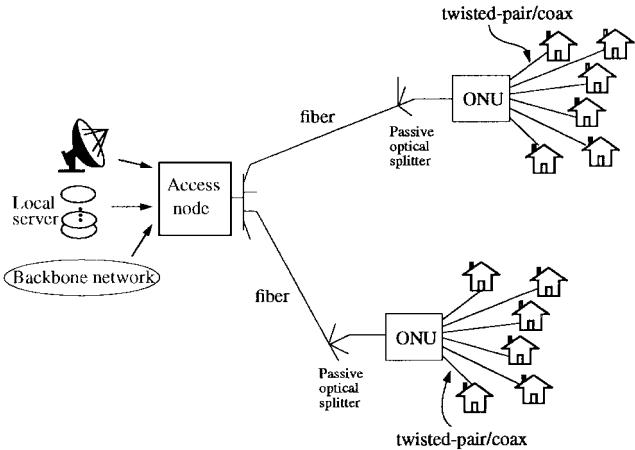


Fig. 14. FTTC network architecture.

- controlling, coordinating, and signaling for multimedia session establishment, maintenance, and disconnection;
- system management, including operation management, fault management, configuration, resource management, and performance management;
- subscriber profile maintenance and billing;
- secure communication to prevent theft of service or unauthorized access, including authentication, encryption, and signal scrambling.

2) *Server Placements*: Video server placement [15], [36]–[39] is an important design issue in VOD systems. The alternatives include centralized video servers, hierarchical video servers, and fully replicated distributed video servers. A centralized video server system is relatively simple to manage. All requests will be sent to and served at one site. Hierarchical server placement exploits the user access pattern and the nonuniform popularity of videos. Popular movies are placed near the customers at local switches, while unpopular ones are stored at a remote central archive. With a good prediction of video access and a good load-balancing mechanism, most requests are satisfied with the local servers, and a small portion of the requests goes to the remote central archive. The distributed server system distributes the video copies to many switches located closer to the users, thus alleviating the congestion in the network and the bottleneck due to the central server, but at the expense of higher cost. Managing distributed servers, however, is more complex. One has to decide which video and how many copies to maintain at each distributed server. In addition, due to varying rates of requests, the video offerings at each distributed server need to be changed periodically. Which alternative is preferable highly depends on the tradeoff between storage and communication costs, the application needs, the underlying infrastructure, and other factors. Li *et al.* [15] proposed a performance model that may be used to evaluate the requirements of network bandwidth and server storage, and hence the tradeoff between communication and storage costs, for various placement alternatives.

3) *Enterprise VOD*: This discussion has focused on large-scale VOD systems supporting tens and hundreds of thousands of users geographically distributed over a wide area and providing services such as movies on demand to the homes. The implementation of such VOD services, however, requires significant investment in the transport infrastructure. Although numerous field trials have taken place around the world [27], it is not expected that such large-scale VOD systems will be widely deployed in the near future. At the same time, smaller-scale VOD systems (also known as enterprise VOD) serving tens and hundreds of users in a small area have become very popular. These systems use technologies similar to those of large-scale, residential VOD systems but will typically be deployed on LAN's rather than MAN's and WAN's. Examples include movies-on-demand systems in hotels, cruise ships, and hospitals. In addition, some companies have deployed enterprise VOD systems for internal education and training.

B. Telecooperation

Telecooperation, also known as computer-supported cooperative work (CSCW) or groupware,⁸ refers to a system that provides an electronic shared workspace to geographically dispersed users with communication, collaboration, and coordination supports [40]–[45]. Group communication provides an electronic channel for the users to exchange messages either synchronously or asynchronously. It allows individuals to cooperate regardless of time and location constraints. Sharing information among groups is the key to effective collaboration. Group coordination manages the participants so as to enhance communication and collaboration. The development of a framework to support task and workflow automation is the key.

With respect to time and space taxonomy, CSCW may be classified into four different interactions [40]: centralized synchronous, distributed synchronous, centralized asynchronous, and distributed asynchronous, as shown in Fig. 15. Synchronous⁹ and asynchronous refer to the time dimension, while centralized and distributed refer to the space dimension. Synchronous exchanges demand real-time communication, and distributed interactions require broadcast or multicast support for group distribution. The centralized synchronous mode requires face-to-face interactions. Applications in the meeting room are examples. The distributed synchronous mode provides real-time interaction in groups dispersed at different locations. Examples include network chat, shared whiteboard (e.g., MBone's wb [46]), real-time joint editing, multimedia conferencing, and videophone. This type of application poses the greatest challenge in the design of group collaboration systems. The centralized asynchronous mode refers to those activities

⁸Strictly speaking, groupware often denotes the technology used to support a group of people working together, while CSCW generally refers to that technology as well as the psychological, social, and other issues. Here we focus on the technical issue.

⁹Synchronous interactions between collaborators require real-time or immediate communication while asynchronous interactions require only deferred communication.

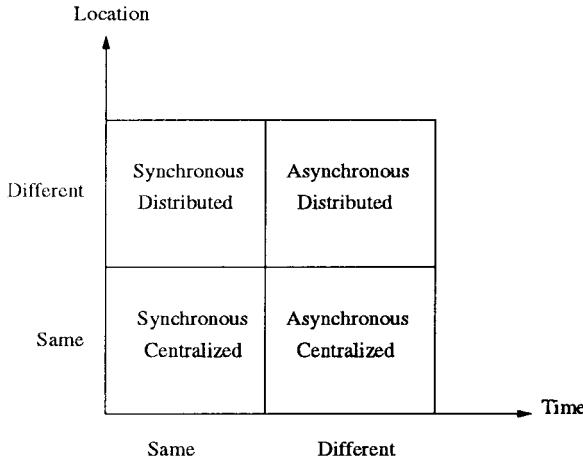


Fig. 15. Classification of interactions by time and space.

held at the same place but at different times. Newsgroups and electronic bulletin boards are such examples. The distributed asynchronous mode allows the exchange of messages within the group asynchronously. Electronic mail and voice mail are examples.

CSCW differs from traditional multiuser applications, such as database access and/or time-sharing systems, in that the latter give the user an illusion that he is the only one using the system by hiding the activities of other users, while in CSCW, users are made aware of the sharing and allowed to collaborate [47]. It is therefore important to model and conceptualize how people work. Following are three important telecooperative applications.

- 1) *Multimedia e-mail*: an electronic messaging system that allows all workers to exchange multimedia messages asynchronously. In fact, e-mail is the most widely used service over the Internet. MIME [48] is the current standard for Internet e-mail. Unlike the older SMPT standard, which understands only ASCII characters, MIME specifies a set of encoding rules and header extensions that describe new types of contents (such as image, video, or audio) and the structure of the multimedia message to embed multimedia information.
- 2) *Collaborative authorship applications*: the activity of collaboratively editing and composing a (multimedia) document by a group of people. This domain of applications can be either synchronous or asynchronous. Each member works on a part, and the completed product is the combination of all individual parts. Examples include Quilt [49], Collaborative Editing System [50], and Mercury [51].
- 3) *Multimedia conferencing*: supports participants with distributed, multiparty synchronous collaboration to simulate face-to-face interactions or real-time telephone conversations. The services may range from point-to-point videophone to multipoint conferencing.
 - a) *Videophone*: the hardware requirements include a telephone equipped with a video camera, which

transmits low-quality images at low frame rates through existing phone lines. An example of videophone is Intel Video Phone [52].

- b) *Desktop conferencing*: the hardware requirements include desktop PC's or workstations equipped with audiovisual facilities. Cornell University's CU-SeeMe [53] is a famous desktop conferencing system.
- c) *Electronic meeting room (or GDSS)*: uses several networked workstations, large computer-controlled public displays, and audio and video equipment to simulate a face-to-face meeting electronically [40], [54]. The PlexCenter Planning and Decision Support Laboratory at the University of Arizona [55] is one example.

There are two approaches to implementing CSCW systems: collaboration transparency and collaboration awareness [47], [56]. Collaboration transparency performs multiuser cooperation on existing single-user applications via an application-sharing system. No collaborative facilities are embedded in the application and thus single-user application programs remain unmodified for group use. One key feature of collaboration-transparent applications is that all the users have to use the same application-sharing system, and usually only one participant is able to access or control the shared windows [42].

Collaboration awareness performs multiuser access via the development of a new special-purpose application explicitly to handle collaboration. This approach embeds collaborative facilities within the applications and usually allows multiple simultaneous users access to the shared windows. Collaboration-awareness applications, however, need to design new multiuser applications from scratch [47].

1) *Telecooperation Infrastructure*: A telecooperation infrastructure provides a robust framework to facilitate group work and to share information. It typically consists of a network model, a system model, and a communication protocol model [57].

a) *Network model*: The network model defines the functional elements and their relationships in the telecooperation system. It includes a multipoint, multicast, and multiservice network and a collection of group collaboration agents. The group multipoint, multicast, and multiservice network connects and distributes multimedia materials to remote participants. A collaboration agent is the hardware and software that provides the necessary facilities and functionalities for cooperation and management of group work. Fig. 16 shows the typical architecture of the network model.

b) *System model*: The system model consists of five major modules, as depicted in Fig. 17: cooperation control, application sharing, conferencing, interface, and database.

The cooperation-control module administers a dynamic set of participants during cooperation sessions [45]. The major functions include access control, group dynamic control, and floor control. Access control validates mem-

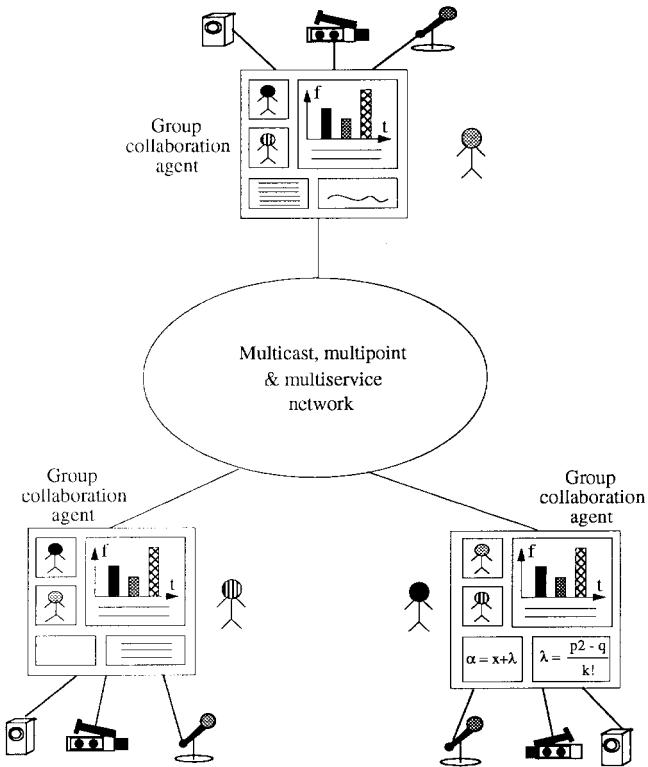


Fig. 16. The architecture of the network model.

bership in a group activity, registers a session, initiates or closes a session, and modifies the membership from one session to another. Group dynamic control allows participants dynamically to add in or drop out of a session. Floor control allows only one participant to own the floor at a time (i.e., only one user can interact with programs at a time) during the session. Interaction coordination is achieved via floor-control mechanisms to ensure that floor control is passed to another user before the user interacts with the session. Following are two variations of floor control policies.

- 1) *Centralized control*: the session chair is responsible for assigning the floor to a session contributor, who returns the right to the chair once he is done.
- 2) *Distributed control*: the current floor holder passes control to another waiting session contributor, who in turn passes the floor to the next waiting contributor after releasing it.

The application-sharing module handles the shared activities of the participants. It provides outputs to all participants with appropriate views while coordinating and controlling the inputs to satisfy the cooperation. There are two possible implementations of the application-sharing module [45], [47]: centralized or replicated. In the centralized approach (i.e., client-server architecture), shared applications are run in a dedicated server. This central server processes all user requests and distributes the results to all local machines for display. A three-tiered architecture can be used to facilitate

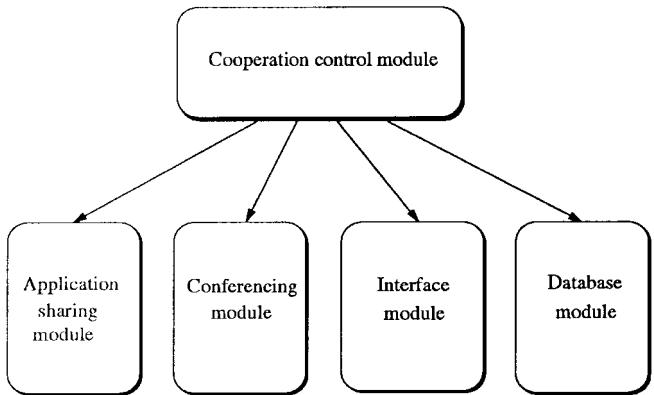


Fig. 17. Telecooperation system model.

group-task coordination for shared applications, including a groupware server, an application server, and a client agent [58]. The basic idea is to use one more level to separate the common, domain-independent functions from specific, domain-dependent tasks. The groupware server provides the coordination functions and necessary interactions across multiple applications, such as cooperation control and basic communication functions. The application server is built on top of the groupware server and handles the needs of the control and execution of a particular application. For example, the requirements of holding a conference and group joint drawing are different and call for two different application servers. The client agent provides user interfaces and other tools to perform group work. An advantage of the centralized approach is the inherent data consistency. It is therefore easy to manage and implement. Disadvantages include vulnerability to central node failures and central node congestion.

In the replicated approach, every local machine keeps and runs a copy of the shared applications, and the input and processing events of the floor holder are distributed to all other sites. This approach is more tolerant against machine failures and more scalable for large and heterogeneous user environments. It also facilitates the support of different views at the local displays. The major challenges with the replicated approach are the maintenance of data consistency and cooperation control among the participants.

The conferencing module supports asynchronous messaging and real-time audiovisual communication for distributed, multiparty collaboration. For video and audio transmissions, full-duplex communication channels and real-time transport services are required between the participants.

The interface module is concerned with the display of the shared data. There are two types of visualization for multiuser interface design: “What You See Is What I See” (WYSIWIS) and relaxed WYSIWIS. WYSIWIS ensures that the effect of each user’s action is completely shared and looks the same on the screens of all participants. Relaxed WYSIWIS relaxes WYSIWIS in four dimensions: display space, display time, subgroup population, and congruence of view [59]. Thus, it supports private and shared windows

for the participants to share only part of the information and allows the shared information to be presented to different users in different ways.

The database module stores shared data and knowledge. Concurrency control is required to resolve the conflicts in the shared data and shared operations between participants. In contrast to database concurrency control (e.g., locking shared data for exclusive access), group concurrency control needs to consider real-time and cooperation issues. Dependency detection [59] and reversible execution [60] are two approaches to concurrency control in group collaboration. Another important requirement, especially in collaborative authorship applications, is the support of multiple versions when changes are made to documents.

c) *Communication protocol model*: This model provides the protocols to exchange information within groups. Two kinds of protocols are required: user presentation and group work management. User-presentation protocols are concerned with client interactions, such as opening, closing, or dynamically joining and leaving a telecooperation session; presentation; and floor control. The group-work-management protocols perform the communication between clients and servers, such as registration of active sessions and inquiry on the current status of cooperative work.

C. Hypermedia Applications

Hypertext incorporates the notions of navigation, annotation, and tailored presentation to provide a flexible structure and access to computer-based digital information. It generalizes the concepts of the “footnote” and “cross reference” of traditional information (or document) organization and allows users to obtain information just by clicking on an anchor (a word, sentence, or paragraph linked to another document) within a document [61]–[63]. Hypertext implies using plain text as the information source. Hypermedia enhances hypertext with multimedia information [64]. Fig. 18 shows a typical structure of hypermedia documents, in which information units are stored in a network of nodes interconnected by links. Hypermedia allows nonsequential navigation within the information space according to individual needs. The author (or system designer) of hypermedia¹⁰ determines the actual structure and navigation options, and the readers decide the order of navigation according to their individual interests at run time.

Some applications are particularly suited for hypermedia, such as encyclopedias, dictionaries, and other reference books [65]. They are composed of a number of independent units that are seldom accessed sequentially but rather by selection and cross reference with other entries. On the Internet, even for technical papers and reports that are considered more suitable for linear reading, there is an in-

¹⁰The terms hypertext and hypermedia are usually distinguished. The former refers to a system with text-only information, while the latter refers to multimedia data. In this article, however, we use the term hypertext to refer to the architecture of the hypermedia system, and the term hypermedia to refer to the system itself.

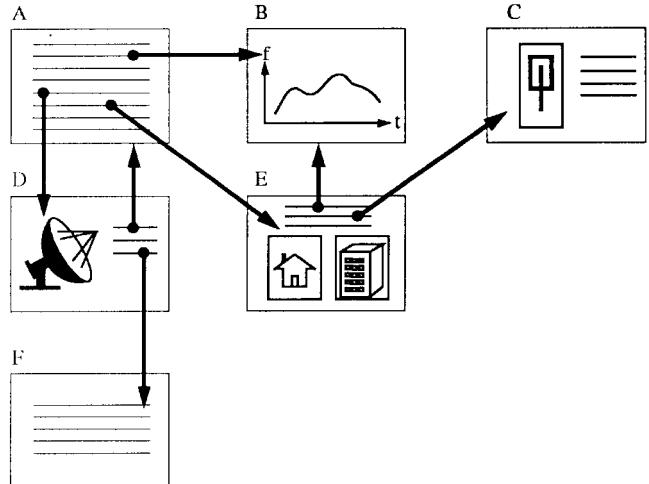


Fig. 18. A hypermedia structure.

creasing tendency to include HTML versions with hypertext links for easy references.

A hypermedia system may be treated as an application of database systems because it provides flexible access to multimedia information and a novel method to structure and manage data. It is different, however, from the conventional notion of database systems. A hypermedia system allows the user more freedom in assimilating and exploring information, while the conventional database system has well-defined structures and manipulation languages for data processing [66].

The basic features of a hypermedia system are summarized as follows [62], [67].

- Information may be divided into several smaller units (or nodes). Each node may contain single media data such as text, source code, graphics, video, audio, or animation, or combinations of two or more media types. The contents in different nodes may have different authors.
- The units of information are interconnected by links. Links can be bidirectional to facilitate backward traversals. Node contents are displayed by activating links.
- The system consists of a network of nodes and links, which may be distributed across multiple sites and remote servers in a computer network.
- Two navigation dimensions are provided. Linear reading is used to move up and down among the document pages at a node, and nonsequential browsing allows users to jump back and forth in the information space and to navigate the hypermedia network at will.
- With authoring tools, users can build their own information structure for various purposes via creation, manipulation, and linkage of shared information units.
- It is necessary to maintain a database system to manage the shared multimedia data. In addition to the traditional database functions, such as data-integrity guarantee, query processing, concurrency control, fail-

ure recovery, and security mechanism, the support of rich modeling, document hierarchy, and hypertext link should be included.

The major usability problem of hypermedia navigation is the disorientation of the readers while browsing [66]. Disorientation refers to the possibility that during browsing, users may be confused as to where they are or may miss the destination they are looking for. It has been shown that 56% of readers get confused about where they are when surfing in hyperspace [68]. Therefore, interface design to avoid user disorientation while reducing the cognitive load is an important design issue. Here, cognitive load refers to the fact that users must memorize the browsing paths they travel in order to get back to the last entry points. Solutions include backtracking facilities (return path), interaction history (e.g., using a bookmark [69]), guided tours, and the fish-eye and structural view of the traversing path (e.g., an overview diagram or a historical log of the reading path with the identification of the current location) [66], [70].

Due to proprietary storage mechanisms and document formats, most current hypermedia systems are not amenable to interoperability, and it is difficult to integrate the materials created in different systems. To make hypermedia systems completely open and interchangeable, various hypermedia models and document architectures have been developed.

1) *Hypermedia Model*: A number of hypermedia models have been developed, including HAM [71], HDM [72], and DHRM [73], [74]. In this paper, we will briefly describe DHRM.

DHRM is an important model designed to make the functionalities of hypermedia systems more generic and integrated into the desktop environment. The Dexter reference model¹¹ attempts to capture and generalize the important features and abstractions from existing and future hypermedia systems. It formalizes the common terminologies and semantics for the systems, provides a generic structure to compare and contrast different hypermedia systems, and serves as a standard to facilitate interoperation and interchanging among various hypermedia systems.

In the Dexter model, a hypermedia system is a collection of components corresponding to the traditional notions of nodes and links. The Dexter model separates a hypertext system into three layers and two interfaces (Fig. 19).

- 1) *Within-component layer*: specifies the structure and contents within the components of the hypertext network according to the individual applications. The Dexter model does not explicitly specify this layer in the model but leaves it to the applications. The standard document architectures such as ODA and SGML can be used for this purpose.
- 2) *Storage layer*: defines and models the components and operations. Each component is associated with a unique identifier (UID). The components are generic

¹¹The Dexter reference model was named after the workshop held at the Dexter Inn in New Hampshire in 1988.

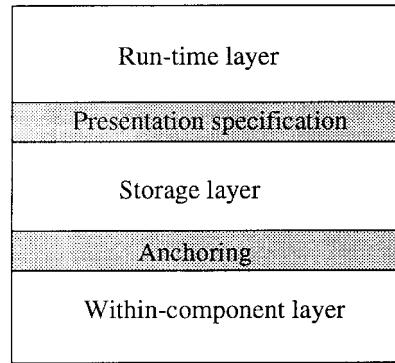


Fig. 19. The layer architecture of the Dexter model.

containers of data and may contain different media types such as text, graphics, images, and video. The storage layer also describes the functions to process the components. Two are the accessor and resolver functions, which jointly provide for the retrieval of components. Other functions support the creation, deletion, and modification of components (see [74] for details).

- 3) *Run-time layer*: provides a tool-independent environment for users to retrieve, view, edit, and manipulate the stored objects in the systems.
- 4) *Anchoring*: glues together the storage layer with the within-component layer. An anchor is defined as an ID referred to by link components in the storage layer and with values pointing to the anchored location of the material in the within-component layer.
- 5) *Presentation specification*: connects the storage layer with the run-time layer. It describes how to start and configure the presentation of components or hypermedia network to the users at run time and embeds such information into the components.

2) *Document Architecture*: The standardized document architectures include SGML/HyTime [75], ODA, and MHEG [76].

- a) *SGML/HyTime*: SGML/HyTime allows various kinds of information (such as multimedia, hypermedia, time-based, or space-based documents) to be packaged in the same document via standard “markups,” which consist primarily of start tags and end tags to separate each logical portion of a document [75]. SGML is a metalanguage that describes the logical structure of a textual document by using the tags to mark the boundaries. The architecture of an SGML document is expressed as the DTD, which formally specifies the element types, hierarchical relationships between the elements, and attributes associated with the elements. For example, the DTD of an article (which may consist of a set of sections, subsections, etc.) may differ from that of a book (which may consist of a set of chapters, subchapters, etc.). HTML is an application of SGML. HyTime adds multimedia and hypermedia support to the document, such as video, temporal-spatial synchronization, and hyperlink.

b) ODA: ODA mainly supports the exchange, processing, and presentation of office-oriented structured documents. Like SGML, ODA specifies the logical structure of a document. In addition, it specifies the layout structure of the document. The layout structure describes the spatial relationships among multimedia objects for presentation. The current ODA standard does not support multimedia because of the lack of support of the time domain in both the layout and logical structures. The integration of continuous media into ODA requires an extension of the ODA standard.

c) MHEG: MHEG¹² standardizes the structure of the interrelationships among different pieces of a multimedia presentation [76].

The scope of the MHEG standard is to define the representation and encoding of multimedia and hypermedia information objects that will be interchanged as a whole within or across applications or services, by any means of interchange including storage devices, telecommunications, or broadcast networks. [77]

MHEG adopts an object-oriented approach. The structure is defined as MHEG classes. The semantics of these classes specify the functionalities and requirements of an MHEG run-time environment. An interactive multimedia presentation is then composed by creating MHEG objects, which are the instances of MHEG classes. The basic elements of MHEG include content data, behavior, user interaction, and composition. Content data are the set of single media data objects in a multimedia presentation, such as text, images, and video clips. Behavior includes all activities concerned with the display of objects in a presentation. It can be divided into presentation action (e.g., set volume, set position, start, and stop) and composition control (e.g., temporal, spatial, and conditional links between objects). User interaction allows users to interact with a presentation. MHEG provides two types of interactions: simple selection and modification. Simple selection allows the user to interact with the playback using predefined sets of alternatives (e.g., push the play, pause, or stop buttons or adjust the volume from a menu), whereas modification is a more complex interaction that allows the user to enter or manipulate data during run time (e.g., type data into an input field).

Composition defines the skeleton of a presentation. From a logical view of a composite object, it packages and links a set of objects of presentation into containers, where objects can be single media objects or other composite objects. Composite objects can be interconnected via hyperlinks and synchronized with one another for presentation. For more details on MHEG, see [76] and [78].

3) WWW: The WWW is evolving toward a universal information hyperspace with a distributed, collaborative

¹²The ISO/IEC JTC1/SC29 standard committee defines the three exchange formats for multimedia systems: JPEG, MPEG, and MHEG. The JPEG and MPEG standards describe data compression for still image and motion picture, respectively. Both are concerned with the content of information. MHEG standardizes the exchange format of multimedia presentation. It is concerned with the structure of information.

infrastructure. In practice, the Web is a vast collection of digitized information documents stored in computers connected to the worldwide Internet. It is accessible through Web browsers such as Netscape and Mosaic and runs on a request-and-response paradigm known as HTTP to allow on-line service to information located in remote sites. HTML is the standard document representation format for Web pages. Fig. 20 shows an example of an HTML document. The output file on the left is generated from the HTML file on the right.

The Web documents created by HTML are static in nature. The author of a Web page determines its contents. Once it is stored in the Web server, each browser request for the document obtains the same response. In dynamic documents, in contrast, the contents are generated by application programs dynamically during run-time, depending on the current input from the user. Two approaches may be used to create such dynamic documents. The first is exemplified by the Common Gateway Interface (CGI), a widely used technology. As soon as a browser requests a document, the Web server invokes a CGI program that creates a document according to the input information. The server then forwards this document to the client browser. For example, one may want to record the number of visitors accessing one's home page, and such information can be reported by this method. This approach, however, renders a document static once it is sent to the client browser. If a continuous update of information is required, this approach may overload the server and decrease the throughput of the system. The other approach allows the dynamics of documents even after they have been loaded into the client browsers. Java, developed by Sun Microsystems, is a popular example of this approach. Rather than return a generated document whenever a browser requests a document, the server sends a copy of a computer program that the browser must run locally. As a result, the display of the document can change continuously and can also interact with user inputs, alleviating the burden of the servers. For example, one may use this approach to display animation of one's home page.

There is no central authority on the Web. Anyone can create a WWW document and reference other documents. The Uniform Resource Locator (URL) is the specification scheme to locate a document unambiguously on the Web. An example of a WWW URL is <http://commsci.usc.edu/faculty/li.html>, where "http" indicates the access method and "commsci.usc.edu" is the machine name. Other methods to retrieve information from the Web include ftp, telnet, gopher, and nntp. A nice property of WWW is that it shields the implementation details of both the formats and the access protocols and presents a uniform interface to users.

IV. OPERATION AND MANAGEMENT OF DMS

A DMS integrates and manages the information, communication, and computing subsystems to realize multimedia applications. In this section, we will examine the key

Victor O. K. Li, Professor, Electrical Engineering



Victor O.K. Li received his SB, SM, and ScD degrees in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology, Cambridge, Massachusetts, in 1977, 1979, and 1981, respectively. He is Professor of Electrical Engineering and Director of the Communication Sciences Institute (CSI) at the University of Southern California, Los Angeles, California, USA. He has published over 200 technical papers and has lectured and consulted extensively around the world. His research interests include high speed communication networks, personal communication networks, distributed multimedia systems, distributed databases, queueing theory, graph theory, and applied probability. Dr. Li chaired the Computer Communications Technical Committee of the IEEE Communications Society 1987-1989, and the Los Angeles Chapter of the IEEE Information Theory Group 1983-1985. He is the Steering Committee Chair and Co-Founder of the International Conference on Computer Communications and Networks (\$IC^3NS\$), General Chair of the 1st Annual \$IC^3NS\$, June 1992, Technical Program Chair of the Institution of Electrical Engineers (IEE) Personal Communication Services Symposium, June 1995, and Chair of the 4th IEEE Workshop on Computer Communications, October 1989. Dr. Li has served as an editor of IEEE Network and of Telecommunication Systems, guest editor of IEEE JSAC and of Computer Networks and ISDN Systems, and is now serving as an editor of ACM Wireless Networks. He serves on the International Advisory Board of IEEE TENCON'90, IEEE TENCON'94, IEEE SICON'91, IEEE SICON'93, IEEE SICON/ICIE'95, the International Conference on Microwaves and Communications '92, and the International Symposium on Communications '91. He was elected an IEEE Fellow in 1992.

Full Publications list:

Fig. 20. An example of an HTML file.

technologies in each subsystem, including the multimedia server and database system for the information subsystem, the operating system for the computing subsystem, and networking for the communication subsystem. We also describe the resource management that ensures end-to-end QoS guarantees, multimedia synchronization for meaningful presentation, and data compression for reducing the storage and bandwidth requirements for multimedia information.

A. Resource Management

Distributed multimedia applications require end-to-end QoS guarantees. Such guarantees have the following characteristics.

- 1) System-wide resource management and admission control to ensure desired QoS levels. The multimedia server, network, or host system alone cannot guarantee end-to-end QoS [79].
- 2) Quantitative specification (such as packet loss probabilities and delay jitter) rather than qualitative description as in the Internet TCP/IP. This allows the flexibility to accommodate a wide range of applications with diverse QoS requirements [80].
- 3) Dynamic management, i.e., QoS dynamically adjusted rather than statically maintained throughout the lifetime of the connection.

QoS, in the context of DMS's, is thus defined as the quantitative description of whether the services provided by the system satisfy the application needs, and is expressed

```
<Title> Victor O. K. Li's Home Page </Title>
<H2>Victor O. K. Li, <EM>Professor, Electrical Engineering</EM></H2>
<HR>
<P>

<P>
```

Victor O.K. Li received his SB, SM, and ScD degrees in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology, Cambridge, Massachusetts, in 1977, 1979, and 1981, respectively. He is Professor of Electrical Engineering and Director of the Communication Sciences Institute (CSI) at the University of Southern California, Los Angeles, California, USA. He has published over 200 technical papers and has lectured and consulted extensively around the world. His research interests include high speed communication networks, personal communication networks, distributed multimedia systems, distributed databases, queueing theory, graph theory, and applied probability. Dr. Li chaired the Computer Communications Technical Committee of the IEEE Communications Society 1987-1989, and the Los Angeles Chapter of the IEEE Information Theory Group 1983-1985. He is the Steering Committee Chair and Co-Founder of the International Conference on Computer Communications and Networks (\$IC^3NS\$), General Chair of the 1st Annual \$IC^3NS\$, June 1992, Technical Program Chair of the Institution of Electrical Engineers (IEE) Personal Communication Services Symposium, June 1995, and Chair of the 4th IEEE Workshop on Computer Communications, October 1989. Dr. Li has served as an editor of IEEE Network and of Telecommunication Systems, guest editor of IEEE JSAC and of Computer Networks and ISDN Systems, and is now serving as an editor of ACM Wireless Networks. He serves on the International Advisory Board of IEEE TENCON'90, IEEE SICON'91, IEEE SICON'93, IEEE SICON/ICIE'95, the International Conference on Microwaves and Communications '92, and the International Symposium on Communications '91. He was elected an IEEE Fellow in 1992. <p>

```
<STRONG><A HREF = "http://commsci.usc.edu/faculty/li-bio.html">
Full Publications list:</A>
```

as a set of parameter-value pairs [81]. Examples of such parameter-value pairs are (packet loss probability, 10^{-3}) and (packet delay, 10^{-6} s). The parametric definition of QoS requirements provides flexible and customized services to the diversity of performance requirements from different applications in a unified system. These QoS parameters are negotiated between the users and the service providers. The service agreement with the negotiated QoS and the call characteristics (such as mean data rate, peak data rate) is called the service contract. With QoS translation between different system layers, QoS requirements can be mapped into desired resources in system components and then can be managed by integrated resource managers to maintain QoS commitment according to the negotiated service contracts. There are three levels of QoS commitment supported by the system:

- 1) deterministic, which guarantees that the performance is the same as the negotiated service contract;
- 2) statistical, which guarantees the performance with some probability;
- 3) best effort, which does not offer service guarantees.

The goal of system-wide resource management is the coordination among system components to achieve end-to-end QoS guarantees. The major functions include 1) negotiate, control, and manage the service contracts of the users and 2) reserve, allocate, manage, adapt, and release system resources according to the negotiated values. Therefore, once the service contract has been negotiated, it will be preserved throughout the lifetime of the connection.

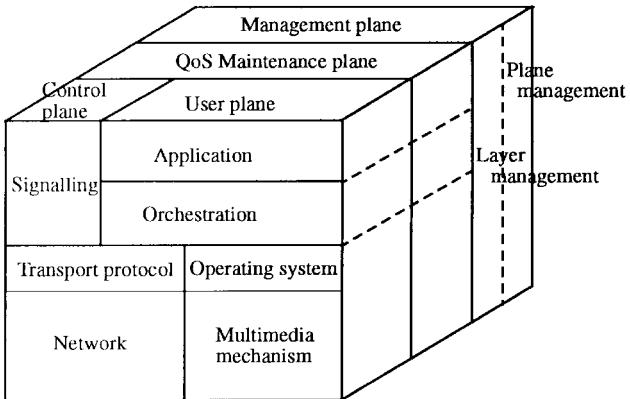


Fig. 21. QoS reference model.

It is also possible, through proper notification and renegotiation, to dynamically tune the QoS level [80], [82]. Admission control protects and maintains the performance of existing users in the system. The principle of admission control is that new requests can be accepted so long as the performance guarantees of existing connections are not violated [83]. Example prototypes of system-wide resource management include the QoS-A model of the University of Lancaster [80] and the QoS broker of the University of Pennsylvania [79].

1) *QoS Protocol Reference Model*: The QoS reference model provides a generic framework to integrate, coordinate, and manage system components to provide end-to-end, guaranteed QoS for a wide range of applications. Fig. 21 illustrates our proposed QoS protocol reference model. It consists of a number of layers and planes. The layer architecture consists of the application layer, orchestration layer, and communication layer.

- 1) The application layer corresponds to the generic application platform.
- 2) The orchestration layer is responsible for maintaining playback continuity in single streams and coordination across multiple related multimedia streams.
- 3) The communication layer provides real-time scheduling via the OS at the host system and real-time delivery via the network.

The communication layer is further divided into two different parts: network and multimedia host system.

The layered architecture hides the complexity of multimedia communications and the diversity of multimedia requirements from the upper layers. The lower layers provide services to the upper layers, and the upper layers specify service requirements. Fig. 22 depicts the relationship between the layered architecture and the ISO OSI protocol reference stack [84].

There are also four planes in the reference model: user plane, control plane, QoS maintenance plane, and management plane.

- 1) The user plane is for the transport of multimedia data from the source to the destination.

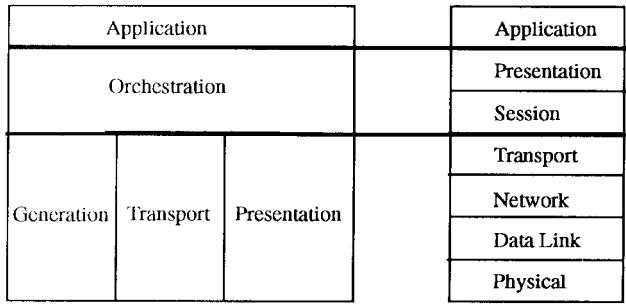


Fig. 22. Multimedia layered architecture versus OSI reference model [84].

- 2) The control plane is for the signaling of multimedia call establishment, maintenance, and disconnection. Such signaling can be transmitted out of band (i.e., in a different channel from the data) to decouple the different service requirements between the data and the control messages.
- 3) The QoS maintenance plane guarantees contracted QoS levels to multimedia applications on an end-to-end basis. The main functions performed on this plane include QoS management, connection management, and admission control. The QoS managers span all system layers, and each is responsible for QoS negotiation, QoS translation between the adjacent layers, and interaction with the corresponding layer (e.g., to gather status messages about QoS from that layer). The connection manager is responsible for the connection establishment, maintenance, and disconnection. The connection manager also has to adapt dynamically to QoS degradation (connection management will be discussed in detail below). Admission control is responsible for performing the admissibility test. Such tests include the schedulability test, bandwidth allocation test, and buffer allocation test. The connection manager is required to interact closely with the QoS manager and with admission control to achieve guaranteed performance.
- 4) The management plane, including layer management and plane management, is responsible for OAM&P services. Layer management and plane management provide intraplane and interplane management for seamless communication.

2) *Multimedia Call Processing*: Due to the real-time requirement, a multimedia application is generally connection oriented. Three phases, namely, establishment, maintenance, and disconnection, are necessary to make a multimedia connection (or call). Each call may be composed of multiple related sessions, and thus orchestration is needed to satisfy synchronization constraints. For example, a conferencing connection demands synchronized video and audio sessions. A call may be initiated by the sender (i.e., a sender-initiated call), as in teleconferencing, or by the receiver (i.e., a receiver-initiated call), as in VOD services. In either case, peer-to-peer and layer-to-layer negotiations will

take place [82]. Peer-to-peer negotiation is held between the caller and callee. For example, to retrieve a video from a remote server, a user needs to negotiate with the server to get the best possible service. If there are enough resources, such as disk transfer bandwidth and network bandwidth, the new request can be granted; otherwise, the user may be offered a lower quality of service. If the caller and callee are in groups, group-to-group negotiation may be necessary. Layer-to-layer negotiation is held between adjacent layers. For example, in video conferencing, the application layer needs to negotiate with the network layer to get multicast services for group distribution. We now describe the three phases in detail.

a) Establishment:

- *Service contract specification:* describes the required QoS constraints for the connection. It should include sufficient information to allow the system to make decisions on the granting or denial of the connection. It involves two steps: QoS negotiation and QoS translation between adjacent layers. QoS negotiation is described above. Once the negotiation is complete, QoS parameter translation maps the QoS between two adjacent layers. For example, application QoS and system QoS translation maps the application QoS parameters into the system QoS parameters. QoS parameter translation is essential to coordinate services in different layers. For example, packet loss parameters mean little to the user but translate into presentation glitches that the user cares about. Parameter mapping from distributed application platforms to the ATM network in the QoS-A project of Lancaster [80] is an example.
- *Admissibility test:* determines if the connection request is granted or denied. Based on the service contract negotiated, the admission test is performed. If the request is denied, a degraded QoS level of service may be offered or the user may be advised to attempt to get service later.
- *Resource reservation and allocation:* Resource reservation is fundamental to end-to-end QoS guarantees. According to the negotiated QoS level, deterministic guarantee necessitates reserving and allocating resources for the worst case, such as the peak rate bandwidth, while statistical guarantee is based on the average case and will provide satisfactory QoS with a certain probability. Best-effort service does not guarantee performance. It is based on the resource-sharing principle and thus no resources are reserved or allocated in advance.

Resource reservation may be sender initiated (e.g., ST-II [85]) or receiver initiated (e.g., RSVP¹³ [86]). The communication may be

point to point (e.g., RCAP [87]) or point to multipoint (e.g., ST-II and RSVP). With the sender-initiated approach, the sender transmits the reservation request toward the receiver(s) along with the path (for unicast) or the multicast tree that is rooted at the sender and extended to all receivers (for multicast). With the receiver-initiated approach, the receivers are responsible for making reservations.

- *Notification of acceptance:* Once the connection has been set up, the resource manager will inform the user of the acceptance.

b) Maintenance:

- *QoS maintenance:* A system-wide resource manager is required to manage the various system components and to perform admission control to guarantee the QoS of the existing connections. The packet loss model described in [83] accounts for packet losses at both the server and the network and can be used as the basis for system-wide admission control. QoS maintenance for each component may include congestion control and flow control in the network as well as real-time scheduling at the host systems and media servers.
- *QoS monitoring:* QoS monitoring performs traffic policing to prevent the user from violating the negotiated traffic characteristics. For example, traffic-shaping mechanisms such as the leaky bucket [88] and the (L, M, T) [89] schemes may be applied. If the user violates the service contract, penalties may be imposed, such as lowering the service priority or charging more.
- *QoS adaptation:* Since the traffic is dynamic, sometimes the system may be overloaded. If the negotiated QoS cannot be guaranteed or the user would like to change the negotiated parameter values during the connection, renegotiation between the user and the system will be required.

c) Disconnection: When a call terminates, the resources assigned to it are released and reallocated by the system.

B. Data Compression

The objective of compression technology is to reduce the amount of data necessary to reproduce the original data while maintaining required levels of coded signal quality, processing delay, and implementation complexity [90]. Data compression plays a critical role in the manipulation of multimedia data because continuous media data are typically large in volume and require real-time and continuous delivery and presentation. Without compression, existing computer platforms, storage devices, and networks are unable to satisfy the requirements of the massive storage space, high data-transfer rate, and huge transmission bandwidth typical of multimedia data. For example, the

¹³The RSVP protocol has been adopted by the IETF working group to standardize and enhance the QoS support of the current Internet architecture. This approach will be discussed in detail in Section IV-C.

transmission bandwidth requirement for digital video with a resolution of 640 by 480, 24-b colors, and 30 fps is $640 \times 480 \times 24 = 7.4 \text{ Mb/f}$, and $7.4 \text{ Mb/f} \times 30 \text{ fps} = 221 \text{ Mb/s}$. The bandwidth currently available is much smaller. For example, CD-ROM drives have 1.2–4.8 Mb/s, T1 lines 1.5 Mb/s, T3 lines around 45 Mb/s, coaxial cables around 40 Mb/s, and OC-3 optical fibers 155 Mb/s. Without data compression, current technologies are inadequate for supporting today's video, and certainly inadequate for the HDTV-quality video in the future.

Compression schemes exploit and eliminate any redundancies within each frame (i.e., spatial redundancy, including space and color similarities) and between sequential frames (i.e., temporal redundancy), resulting in the reduction of data being stored and transmitted. The former is called *intraframe* compression and the latter *interframe* compression. Note that squeezing the color similarities exploits the artifacts of human vision, which is more sensitive to luminance (i.e., brightness) than to chrominance (i.e., color changes). Compression techniques can be categorized based on the following considerations.

- 1) *Lossless or lossy*: If the decompressed data is identical to the original, it is referred to as lossless compression; otherwise, lossy compression. Lossy compression schemes usually have higher compression ratios but may introduce some compression artifacts such as aliasing, a distortion or stair-step effect in the image. The baseline of lossy compression is decided by the human eye.
- 2) *Entropy encoding or source encoding*: Entropy compression does not differentiate the types of data but analyzes the statistical properties of data to reduce the size. Source compression deals with the contents of the source material and makes use of their semantic and special characteristics to achieve data reduction. Entropy compression is usually lossless (such as run-length coding and Huffman coding) and source compression is usually lossy (such as discrete cosine transform plus quantization).
- 3) *Symmetrical or asymmetrical*: If the time required to compress and to decompress is roughly the same, it is referred to as symmetrical compression. In asymmetrical compression, the time taken for compression is usually much longer than decompression. Symmetric schemes are usually adopted in applications that must digitize and compress data in real time, such as video conferencing. Asymmetrical schemes, on the other hand, are often used when the data are compressed once and decompressed many times, as in most interactive TV applications.
- 4) *Software or hardware*: The compression may be performed by software or hardware. Hardware codecs (Coder and DECoder) are faster and are suitable for CPU-intensive processing. Existing hardware codecs often offer better quality and performance. However, they are available only on platforms with costly

extended hardware and dedicated hardware doomed to rapid obsolescence. Examples include JPEG, MPEG1, MPEG2,¹⁴ and DVI.¹⁵ Software codecs are suitable for less CPU-intensive processes and are limited to those processes capable of decoding an image within a frame period. Compatibility of the underlying hardware ensures wider consumer accessibility, lower costs, and upward compatibility. Examples includes Indeo of Intel, Cinepack of Radius, TrueMotion-S of Duck Corporation, Motion Pixels, and Video Beam.

1) *Basic Compression Techniques*: Some compression methods are based on a particular technique and some (especially those for multimedia applications) are hybrids of several techniques. Some basic techniques are examined first, and specific techniques applied to multimedia applications will then be described.

a) *Lossless compression*:

i) *Run-length coding*: In run-length coding, a string of repeated symbols is replaced with the symbol and the number of instances it is repeated. For example, "abaaaabbbbbbb" is expressed as "a1b1a4b7."

ii) *Variable-length coding*: Variable-length coding encodes data based on the frequency of occurrence, with the most frequently occurring data patterns replaced with shorter symbols. Huffman coding and arithmetic coding (in which data are encoded into numerical values) are examples.

b) *Lossy compression*:

i) *Predictive encoding*: Predictive encoding exploits the redundancy from sample to sample and derives the upcoming sample from the previous sample, where the sample may be a pixel, line, audio sample, or video frame. It encodes, stores, and transmits the differences between the prediction and the input sample value. The decoder produces the prediction from the previous sample and reproduces the sample from both the prediction and the difference value. Since the size of the difference between samples is usually smaller than that of the original sample, the file size may be reduced. Its success, however, highly depends on the prediction function and the ability to anticipate the changing signal [12]. This technique is most suitable for digital audio or video encoding, in which adjacent samples or frames are generally similar to each other. ADPCM is an example of predictive encoding.

ii) *Transform coding*: Transform coding converts data from one domain to another that is more convenient for compression. Of course, the inverse transform function must exist to allow conversion back to the original domain [91]. For example, DCT encoding transforms samples from the time domain to the frequency domain. The input of the DCT engine is a two-dimensional block, typically with 8×8 pixels. DCT transforms an 8×8 block of pixel color values into an 8×8 matrix of spatial frequencies. The

¹⁴There are software JPEG and MPEG coders also.

¹⁵In contrast to JPEG and MPEG, DVI is proprietary and requires the support of special hardware to perform compression and decompression.

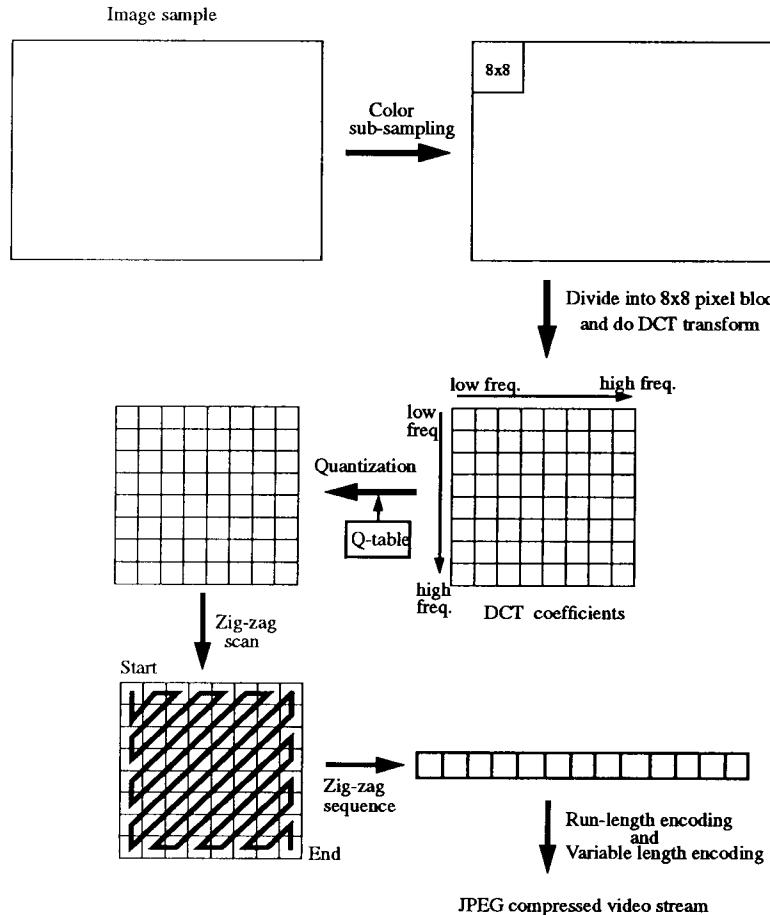


Fig. 23. The steps of lossy JPEG compression.

lower frequencies (i.e., general information) are positioned near the upper-left corner, and the higher frequencies (i.e., sharp edges) are near the lower-right corner. The coefficients of the output DCT matrix are divided by the corresponding entries of an 8×8 quantization table (Q -table). Each entry of the Q -table describes the quality factor of the corresponding DCT output coefficient. The lower the quality factor, the higher the magnitude of value. For example, for DCT compression, the lower frequencies are more important than the higher ones. Therefore, a Q -table with smaller numbers near the upper-left corner and larger ones near the lower-right corner is used. The division of the DCT output matrix by the Q -table causes the lower-right-corner portion of the resultant matrix to become mostly zeros. As a result, the greater portion of the high-frequency information is discarded. The modification of the Q -table is one way to change the degree of compression. The DCT decoder performs the reverse process and converts the 64 coefficients back to pixel values. Since quantization is a lossy process, it may not reproduce the originals exactly.

2) Multimedia Compression Techniques:

a) *Still image*: A still image compression scheme throws away the spatial and/or color redundancies. It can be further divided into lossless (with compression ratios of 2–5), such as run-length coding and variable-length coding,

and lossy (with compression ratios of 10–20), such as DCT, vector quantization, fractal, and wavelet coding.

The standard compression scheme for still images is JPEG [92], [93], which is a hybrid of several basic compression methods. JPEG, developed by the ISO and CCITT, is designed for arbitrary still images (i.e., continuous-tone images) and supports four modes of encoding.

- 1) *Lossless*: the image is encoded in such a way that it is guaranteed to be restorable.
- 2) *Sequential*: the image is encoded in the order in which it is scanned.
- 3) *Progressive*: the image is encoded in multiple passes, so that the entire content can be visualized in a rough-to-clear process (i.e., refined during succeeding passes).
- 4) *Hierarchical*: the image is encoded at multiple resolutions.

JPEG uses predictive encoding to achieve lossless compression and uses DCT encoding and quantization to perform the other three operations. Since quantization is a lossy algorithm, sequential, progressive, and hierarchical encoding are lossy compressions. The basic steps of lossy compression (shown in Fig. 23) are summarized as follows.

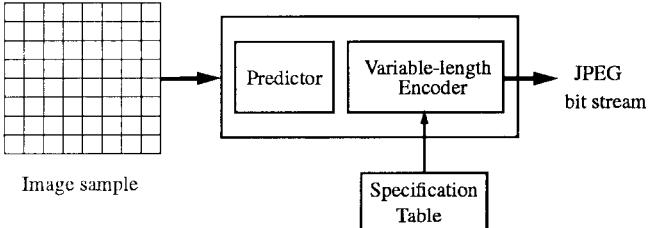


Fig. 24. The steps of lossless JPEG compression.

- Transform *RGB* to *YUV* color space and downsamples chrominance to exploit color redundancy.
- Apply DCT transform on pixel blocks to exploit spatial redundancy (images are divided into a series of blocks of 8×8 pixels before entering the DCT engine).
- Quantize the 64 frequency components in each block with higher quantization values on the high frequencies.
- Order the 2-D output of quantization in a zigzag sequence, creating a bit stream with zero value coefficients close to the end, since the low-frequency components (most likely nonzero) are placed in front of the high-frequency components (most likely zero).
- Apply run-length encoding to the zeros of the zigzag sequence.
- Further apply variable-length encoding on the run-length coded stream.

In lossless compression (shown in Fig. 24), images are compressed first by applying simple predictive coding schemes, and then variable-length coding is used to achieve data reduction.

b) Digital video: Digital video compression may either apply intraframe compression to each individual frame of the video (e.g., Motion-JPEG) or combine both intraframe and interframe compression (e.g., H.261, MPEG video standard). The former scheme yields a lower compression ratio than the latter because it does not exploit temporal redundancy. Differential encoding (e.g., DPCM) is one approach to squeeze out temporal redundancy. It encodes and sends the differences in the adjacent frames. Motion compensation is another technique for interframe compression. It compares the current frame with the previous one and encodes the motion vectors (i.e., the change in coordinate values due to the motion) and pixel differences after motion (see Fig. 25).

i) H.261 ($p \times 64$) and H.263: H.261 [94] by CCITT (the predecessor of the ITU-T) was developed for video phones and video conferencing over ISDN telephone lines. It provides audiovisual services at integral multiples of 64 Kb/s (i.e., $p \times 64$, $p = 1, 2, \dots, 30$) where 64 Kb/s is the baseline rate of ISDN systems.

For interoperability, source video formats must be converted to a CIF. CIF defines the hierarchical data structure for compressed streams. It also specifies a maximum picture rate of 29.97 fps and the formats of the luminance and chrominance components. Two variations of CIF are

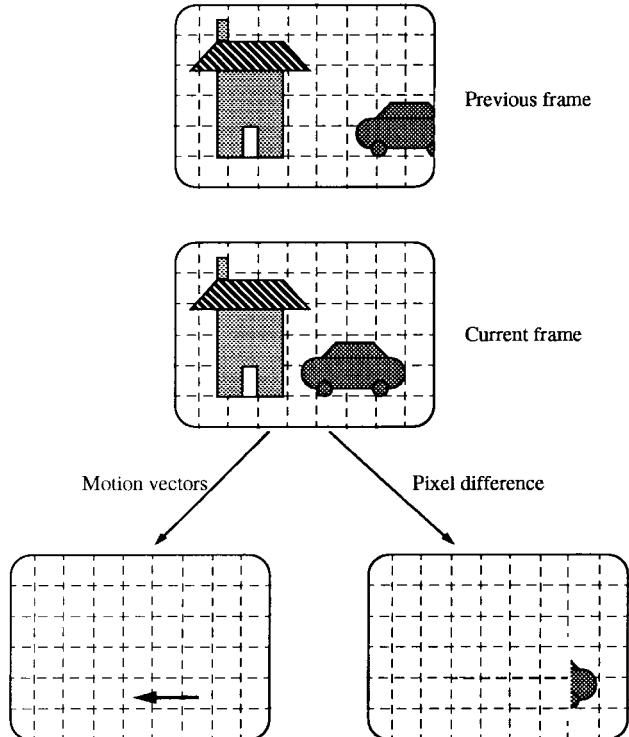


Fig. 25. Motion compensation.

supported by H.261: CIF and QCIF (at a quarter of the resolution of CIF). CIF uses 352×288 for luminance, i.e., 352 pixels per line and 288 lines, and 176×144 for the two chrominance components, whereas QCIF uses 176×144 for luminance and 88×72 for the two chrominance components. (Note that according to the H.261 standard, all codecs must be able to work with QCIF, and CIF is optional.)

H.261 adopts DCT for intraframe compression to eliminate spatial redundancy, and differential DPCM with motion estimation for interframe compression to reduce temporal redundancy. The basic idea is as follows. In the intraframe mode, every 8×8 block in a picture is DCT transformed, while in the interframe mode, motion compensation with DCT is applied. A comparison of the current frame with the previous one is made. If the difference is higher than a certain threshold, the difference is sent to be DCT transformed; otherwise, no data is sent.

H.263 is the revised version of H.261. It is a provisional ITU-T standard for a wide range of bit-rate video coding¹⁶ and is used for video conferencing over the PSTN. The coding algorithm of H.263 is similar to H.261 but with some improvements and changes. For example, negotiable options are supported to enhance performance, and some parts of the hierarchical structure of the data stream are optional to allow lower bit rates and better error recovery. Five resolutions are supported by H.263. In addition to CIF and QCIF, there are SQCIF, 4CIF, and 16CIF. The

¹⁶It was originally designed for very low bit rate, but now this restriction has been removed.

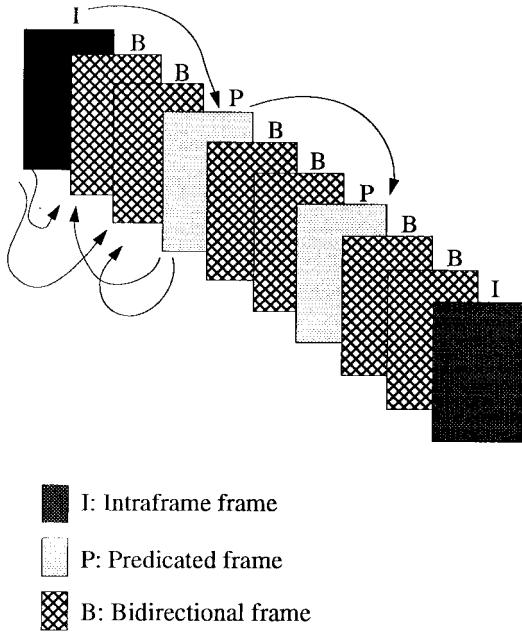


Fig. 26. Different types of MPEG frames.

resolution of SQCIF is about half that of QCIF, 4CIF four times that of CIF, and 16CIF 16 times that of CIF.

ii) MPEG video: Unlike H.261, which is optimized for teleconferencing where motion is naturally limited, the ISO MPEG video standard is devised for a wide range of video and motion pictures¹⁷ [95], [96]. It defines the structure of coded video bit streams transmitted to the decoder and the decoder architecture but leaves the encoder¹⁸ architecture undefined, so that some encoders may produce higher quality compression, some will be real-time, and some will require manual intervention. MPEG compression is lossy and asymmetric, with the encoding more expensive than the decoding.

MPEG uses both intraframe and interframe techniques for data compression. There are three types of encoded frames: I (intraframe), P (predicted), and B (bidirectional), as shown in Fig. 26. An I-frame contains all of the necessary information to reproduce a complete frame and thus is independent of any other frames in the video sequence. It basically applies the intraframe compression of the basic JPEG DCT algorithm. A P-frame is predicted from the differences from the previous I- or P-frame in the sequence. A B-frame is bidirectional interpolated by the preceding and the subsequent I- or P-frames. The interframe dependency of P- and B-frames implies that a current P-frame can be decoded only if the preceding I- or P-frame has been presented to be decoded. Similarly, without both the preceding and subsequent I- or P-frames in the decoder, it is impossible to generate a B-frame. For example, if the presentation order of a video sequence is IBBPBBBBI ... then the actual input sequence of MPEG frames to a decoder should be

¹⁷The goal of MPEG is to define a standard for coded representation of moving pictures, associated audio, and the system.

¹⁸The basic requirement of MPEG encoders is to produce MPEG-compatible video.

IPBBPBBIBB.... Therefore, the patterns of video frame sequence in storage and during transmission are different from that for presentation. The basic compression steps are:

- 1) preprocess and color subsample;
- 2) for I-frame, skip this step; for P-frame and B-frame, do interframe motion compensation;
- 3) transform 8×8 pixel blocks to an 8×8 frequency coefficient matrix, and send 8×8 blocks (either intrablocks or motion-compensated blocks) to the DCT engine and quantizer;
- 4) apply zigzag ordering on the 8×8 quantized coefficients;
- 5) apply run-length encoder to the zeros of the zigzag sequence;
- 6) apply variable-length encoder on the run-length coded stream.

There are different versions of MPEG, denoted by MPEG-x. MPEG-1 (i.e., the first version) was targeted at CD-ROM and applications at a bit rate of about 1.5 Mb/s. The video is strictly progressive (i.e., noninterlaced) and picture quality is equivalent to VHS. MPEG-2 addresses high-quality coding for all digital transmissions of broadcast-TV-quality video at data rates of 2–15 Mb/s. The major applications include digital storage media, digital television (including HDTV)¹⁹, broadcasting over cable, satellite, and other broadcast channels, and other communications applications [97]. MPEG-2 is backward compatible to MPEG-1. The primary enhancement is centered around the addition of interlaced video. Thus, there are two coding structures (field and frame) in MPEG-2. Other improvements include enhancements on coding efficiency. MPEG-4 is now in the application-identification stage. It targets applications with very low bit rates of 4.8–64 Kb/s, such as videophone over the PSTN and visual transmission over mobile facilities.

c) Digital audio: Compression of entertainment-quality audio signals has been widely studied, from μ -law companding and ADPCM to the more recent MPEG audio standard and Dolby AC-3 with high fidelity and surround-sound effect. PCM produces the highest audio quality with no compression loss. Compared to PCM, μ -law is of low complexity but often produces poor quality compressed audio; ADPCM is of medium complexity and produces average quality; and MPEG audio and Dolby AC-3 produce the highest quality compressed audio with no perceivable loss compared to PCM.

i) μ -law companding: μ -law is widely used to encode speech over telephone lines in the United States and Japan. A similar approach called A-law is used in Europe. The basic idea is to use quantization steps of different sizes that increase logarithmically with the signal levels to compress the signal, and to use the inverse function to expand the signal. As a result, the quantization levels are spaced far apart when the signal is louder and closer

¹⁹Originally, HDTV applications were addressed by MPEG-3, but eventually they were folded into MPEG-2.

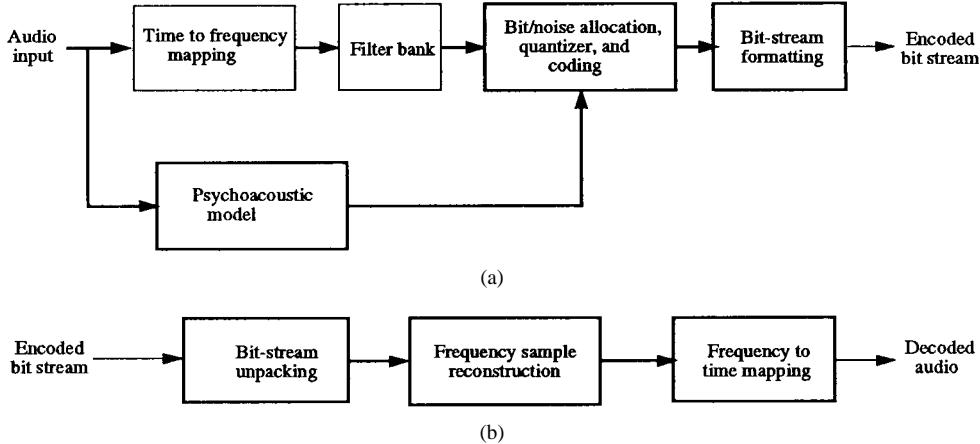


Fig. 27. The block diagram of MPEG/audio encoder/decoder. (a) MPEG/audio encoder. (b) MPEG/audio decoder.

together when the signal is quieter [12]. The logarithmic characteristic of μ -law is

$$Y = \frac{\log(1 + \mu x)}{\log(1 + \mu)}, \quad \text{for } 0 \leq x \leq 1$$

where Y is the magnitude of the output, x is the magnitude of the input, and μ is a positive parameter used to define the desired compression level (μ is often set to 255 for standard telephone lines).

ii) ADPCM: ADPCM takes advantage of the fact that consecutive audio samples are generally similar and employs adaptive quantization with fixed prediction. Instead of quantizing each audio sample independently, as in PCM, an ADPCM encoder calculates the difference between each audio sample and its predicted value and outputs the quantized difference (i.e., the PCM value of the differential). The performance is improved by employing adaptive prediction and quantization so that the predictor and difference quantizers adapt to the characteristics of the audio signal by changing the step size. The decoder reconstructs the audio sample by using the identical quantization step size to obtain a linear difference plus possibly adding 1/2 for the correction of truncation errors, and then adding the quantized difference to its predicted value.

The method of calculating the predicted value and the way of adapting the audio signal samples vary among different ADPCM coding systems. For example, the ADPCM algorithm proposed by the IMA takes 16-b linear PCM samples and outputs 4-b ADPCM samples, yielding a compression factor of 4:1.

iii) MPEG audio: MPEG audio compression employs perceptual coding that takes advantage of the psychoacoustic phenomena and exploits human auditory masking to squeeze out the acoustically irrelevant parts of audio signals [98]. Audio masking is a perceptual weakness of human ears. Whenever a strong audio signal is present, the spectral neighborhood of weaker audio signals becomes imperceptible. Measurements of human hearing sensitivity

have shown that the ear integrates audio signals that are very close in frequency and treats them as a group. Only when the signals are sufficiently different in frequency will the ear perceive the loudness of each signal individually. In perceptual coding, the audio spectrum is divided into a set of narrow frequency bands, called critical bands, to reflect the frequency selectivity of human hearing. Then it is possible to filter coding noise sharply to force it to stay close to the frequency of the frequency components of the audio signal being coded, thereby masking out the noise. By reducing the coding noise when no audio is present and allowing strong audio to mask the noise at other times, the sound fidelity of the original signals can be perceptually preserved.

Fig. 27 shows the block diagram of MPEG/audio encoding and decoding [98]. The audio input is transformed from the time domain to the frequency domain. It then passes through a filter bank, which divides the resultant spectrum into multiple subbands. Simultaneously, the input stream goes through a psychoacoustic model, which exploits auditory masking and determines the signal-to-noise ratio of each subband. The bit/noise allocation block determines the number of code bits allocated to quantize each subband so that the audibility of quantization noise is minimized. The quantized samples are then formatted into a series of bit streams. The decoder performs the inverse operation of the encoder and reconstructs the bit stream to time-domain audio signals. To provide different quality and different bit rates, the standard supports three distinct layers for compression: layer I is the most basic and simplest coding algorithm, and layers II and III are enhancements of layer I.

MPEG-2 audio is being developed for low-bit-rate coding of multichannel audio. It has 5.1 channels corresponding to five full range channels of audio, including three front channels (i.e., left, center, right) plus two surrounds, and a low-frequency bass-effect channel called the subwoofer. Since the subwoofer has only limited frequency response (about 100 Hz), it is sometimes referred to as the ".1" channel. In addition to two main channels (left and right)

backward compatible to MPEG-1 audio, MPEG-2 audio uses new coding methods and syntax for the surround channels.

iv) *Dolby AC-3*: Dolby AC-3 is a multichannel perceptual coding technology used in Dolby Digital film soundtracks and the new Dolby Digital (Surround AC-3) home-theater equipment, such as laser discs [99]. Contrary to MPEG2 audio, Dolby AC-3 is not backward compatible to MPEG1 audio. Using sophisticated audio masking and a shared bit-pool arrangement for audio-noise reduction, AC-3 achieves very effective data reduction and offers sound quality extremely close to the original. It enables high fidelity, 5.1-channel surround sound to be encoded in less space than that required by a single channel on a CD.

Dolby AC-3 has been officially adopted by the Grand Alliance²⁰ to provide digital surround sound in the forthcoming U.S. HDTV audio system and is being used for DBS in two-channel form. In addition, Digital Cable TV and DVD have adopted AC-3, and DVC and DAB may do so as well.

C. Networking

The communication network, encompassing the transmission media and the transport protocols, transports multimedia traffic to satisfy QoS guarantees on an end-to-end basis. According to application requirements, the transmission media may be wired (twisted pair, coaxial cable, fiber optics) or wireless (radio channels, satellite channels) and the transport protocols may provide connection-oriented or connectionless services and best-effort, statistical, or deterministic performance guarantees. With respect to the taxonomy of the geographical coverage area, the network can be a LAN, a MAN, or a WAN. A LAN (e.g., Ethernet, token ring, token bus) may cover an area within a building, a campus, or an organization; a MAN (e.g., FDDI, DQDB) may cover a metropolitan area, such as a small city; a WAN (e.g., TCP/IP, ATM) is a nationwide or an international network.

To support multimedia applications, the network must satisfy the requirements for multimedia data transmission. Multimedia traffic has diverse characteristics and various QoS requirements. Discrete media traffic (e.g., file transfer, image retrieval) requires error-free services but is tolerant of delay. Continuous media traffic (e.g., video or audio playback), on the other hand, requires real-time, high-speed transmission and is connection oriented. It is sensitive to delay and delay jitter, or the packet delay variations between consecutive packets, but is tolerant of occasional packet losses. In addition, the network has to support application-specific requirements. For example, video conferencing needs multicast service for group distribution, and ITV requires switched, point-to-point services and asymmetric

²⁰The Digital HDTV Grand Alliance, announced on May 24, 1993, represents the merging of technologies from the three groups that had developed all-digital HDTV systems for consideration as the U.S. standard: AT&T and Zenith Electronics Corporation, General Instrument Corporation and the Massachusetts Institute of Technology, and a consortium composed of Thomson Consumer Electronics, Phillips Consumer Electronics, and the David Sarnoff Research Center.

bandwidth for the downstream (video server to user) and the upstream (user to video server) directions.

The Internet has rapidly evolved into a significant network infrastructure for communication. It runs the Internet protocol (IP) with its best-effort delivery service and enjoys a large user base. Another promising technique, ATM, is rapidly appearing in the market due to the extensive support of telephone companies. It allows bandwidth on demand and guaranteed QoS and is expected to be the best candidate for high-quality media delivery. We will examine the Internet effort on the support of multimedia transmission, the elegant properties of ATM that make it especially suited for distributed multimedia applications, and the issues of integrating ATM networks with the emerging integrated services Internet.

1) *IP*: Internet refers to the global network to which a large percentage of existing networks are now interconnected by routers or gateways, running the TCP/IP protocol suite. IP is the key to success for the Internet. It provides datagram delivery (best effort, connectionless) service and leaves the reliability issues (delay, out-of-order delivery, packet losses, misdelivery) to the end systems. It uses a global addressing scheme for a vast range of services. Each datagram, in addition to data, carries routing information in the header so as to be independently forwarded to the destination.

The primitive service model of the Internet provides only point-to-point and best-effort services. Such services are particularly suited for traditional applications such as file transfer and remote login. It performs well for real-time media traffic (e.g., video, audio) only under lightly loaded networks. The support of IP multicast, resource reservation, and higher level RTP's are the major development efforts on the Internet geared toward multimedia applications. We will examine these three issues next.

a) *IP multicast*: Multicast refers to the ability to send a single packet to multiple destinations. Applications that require multicast include teleconferencing, e-mail multicast, Usenet news, remote learning, and group communication. Traditional protocols such as TCP or UDP provide only unicast transmission. The provision of multicast with unicast services requires the delivery of replicated unicast packets to each recipient. To avoid sending multiple replicated unicast copies, thereby increasing network utilization, multicast transport service is required. With IP multicast, each data source sends a packet to a multicast address and lets the network forward a copy of the packet to each of a group of hosts. Fig. 28 shows the difference between unicast and multicast transport services. The basic transport service of IP multicast is unreliable multicast transmission of datagrams, which is suitable for applications geared more toward performance than reliability. A reliable protocol for IP multicast called MTP [100] has been developed to ensure atomicity and reliability.

Multicast routing protocols provide a mechanism that enables routers to propagate multicast datagrams in such a way as to minimize the number of excess copies transmitted to any particular subnetwork. In addition, they must be

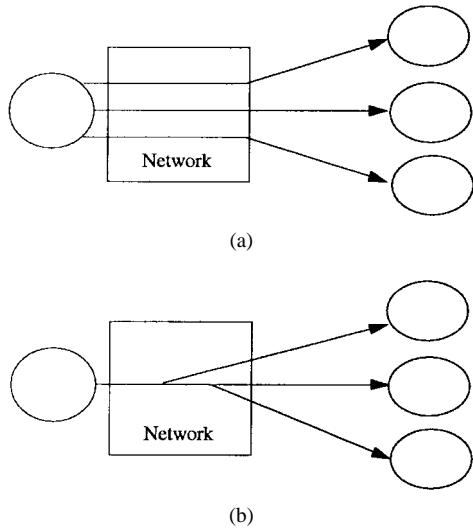


Fig. 28. Unicast and multicast services for multipoint communication. (a) Unicast service. (b) Multicast service.

flexible to allow participants to join and leave the system without affecting others. The key is to determine which hosts on which subnetwork will participate in the group. Two important multicast routing protocols are link state multicast and distance vector multicast, which are extensions from the corresponding unicast routing algorithms.

i) Link state multicast: In a link state protocol, such as OSPF, each router monitors the state (or metric) of its adjacent links and sends this information to all other routers whenever the state changes. The metric may be the reliability, delay, cost, etc. of the link. Thus, each router has a picture of the entire network. Suppose a router A wants to multicast to a group of users. First, it needs to identify the subset of routers these users are connected to. This is accomplished by adding the ID's of the groups that have members on a particular link to the state of the link. Then it computes the shortest distance tree rooted at A and spanning this subset of routers based on the link state metrics. The multicast is then performed on this shortest distance tree.

ii) Distance vector multicast: In a distance vector protocol, each router maintains a routing table with entries {destination, cost, next node} indicating for each destination the least cost to reach it and the next node on the shortest path to it. For each destination it can reach, a node sends {destination, cost} to its directly connected neighbors, informing them of the cost of reaching the destination if this node is used as a relay. Each node will update its routing table if it finds that a neighbor can provide a shorter path to a destination. To support multicasting, each node will also send the ID of the groups that can be reached via this node to each of its directly connected neighbors. This information can be included in the routing table and each router then knows over which links it should forward multicast packets for each group. To avoid sending duplicate multicast packets when multiple routers are connected to a router or link, we designate one router as the “parent” router of each router or link.

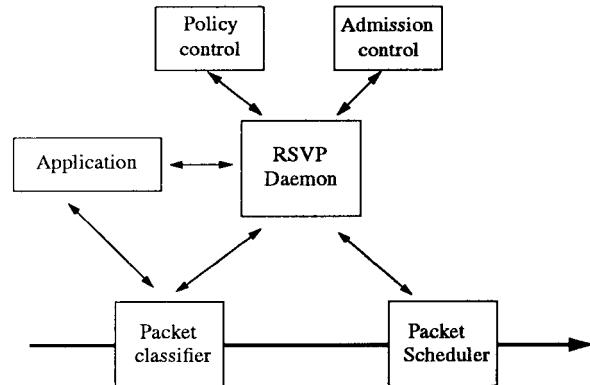


Fig. 29. RSVP protocol architecture.

The ID of this parent is a function of the source of the multicast. The router with the shortest path to the source is selected as the parent, and only the parent will forward a multicast packet.

IP multicast services on the Internet are being realized with the development of MBone.²¹ MBone is a virtual network implemented on top of the physical Internet [46], [101], [102]. The network consists of subnetworks that can directly support IP multicast, such as Ethernet or FDDI, connected by virtual point-to-point tunnels. Tunnel-connected endpoints are multicast-enhanced routers (i.e., mrouters) that encapsulate multicast IP packets within regular IP datagrams for transmission through the existing Internet. Since they look like normal unicast packets, no special treatment by any regular routers and subnets is necessary. When an mrouter receives a packet through a tunnel, it strips off the header and multicasts the packet as appropriate. MBone supports many applications for multiparty video conferencing. The most popular MBone tools include network video (nv), visual audio tool (vat), and shared drawing whiteboard (wb).

b) IETF RSVP: RSVP attempts to provide guaranteed QoS services for heterogeneous receivers across the Internet with multipoint-to-multipoint communications. It runs on top of IP, either IPv4 or IPv6. Fig. 29 summarizes the RSVP protocol architecture [103]. To reserve resources at a router, the RSVP daemon communicates with the admission-control and policy-control modules. Admission control determines if there are sufficient resources to satisfy the QoS of the new request while guaranteeing the QoS requirements for existing connections. Policy control determines if the user has the administrative permission for resource reservation. If both check out, the RSVP daemon sets parameters in the packet classifier, which determines the QoS for each packet, and the packet scheduler, which orders packet transmissions; otherwise, the reservation will be denied.

RSVP is receiver initiated, and two types of messages are used to reserve resources: PATH and RESV. Each data source sends a PATH message, which contains a flow specification (e.g., bandwidth) to the destination multicast

²¹ MBone was first used to deliver live video and audio sessions of an IETF meeting on the Internet in 1992.

address. When a router receives a PATH message, it records the relevant information (e.g., IP multicast address, flow specification, source ID, etc.). As a result, not only are the receivers informed of the flow specification of the data traffic but the intermediate nodes also obtain the PATH state information. Based on information obtained from PATH messages and from higher layer protocols, each receiver can determine its QoS requirements and initiate a RESV message to reserve specific resources along the reserved route of the PATH message. Multiple receivers may send RESV messages toward the same data source. These RESV messages are merged as they travel up the multicast tree, and only the merged RESV is forwarded toward the source. Thus, RSVP scales well and accommodates heterogeneous receiver requirements. Resources will then be reserved along the route from the source toward the receiver.

For reliability and robustness, RSVP periodically updates the soft states at intermediate routers to allow for users joining and leaving a group, where the soft state refers to “a state maintained at network switches which, when lost, will be automatically reinstalled by RSVP soon thereafter” [86].

Presently, three reservation styles have been defined: fixed filter, wildcard filter, and shared explicit [104]. Fixed filter allows explicit sender selection and distinct reservation and is suitable for applications such as video conferencing, in which each video stream has its own RSVP flow. Explicit sender selection refers to the reservation made only for senders explicitly listed in the RESV message, while wildcard sender selection includes any sender. Distinct reservation means that each sender has its own RSVP flow, while shared reservation uses the same RSVP flow for multiple senders.Wildcard filter allows wildcard sender selection and shared reservation. It is suitable for applications such as audio conferencing, in which participants typically take turns speaking. Shared explicit allows explicit sender selection and shared reservation. It is similar to wildcard filter but with better security.

c) RTP: RTP provides end-to-end data-transport services for the delivery of real-time data (e.g., video, audio) over unicast or multicast networks [105]. The data transport is augmented by the RTCP to monitor QoS and provide minimal control, management, and identification. RTP and RTCP are independent of the transport protocols and the underlying network, although they are typically run on top of UDP/IP and IP multicast. Except for framing, RTP makes no specific assumptions on the capabilities of the lower layers. It does not perform resource reservation, and no QoS guarantees are made. In addition, no mechanisms are defined for flow control and reliability. Timely delivery relies only on the support of lower-layer services such as RSVP that have control over the resources in networks. The functionalities of RTP include time stamping, sequence numbering, timing reconstruction, loss detection, and channel and content identification. Based on the timing information and sequence number carried in the RTP header, receivers are allowed to reconstruct and to

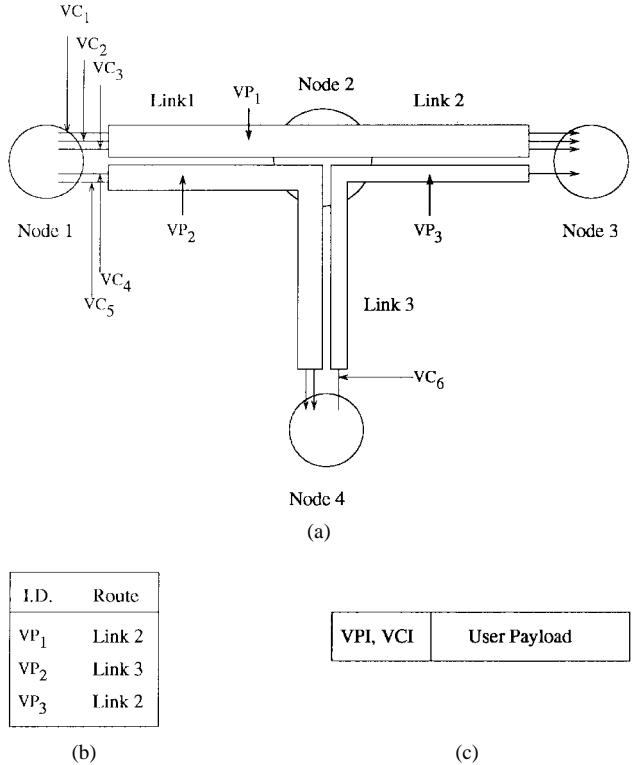


Fig. 30. The ATM virtual path concept. (a) Virtual paths, virtual channels, links, and nodes. (b) Routing table at node 2. (c) VP, VC identifiers in a cell.

play audio and video sequences despite occasional loss or misdelivery of packets.

2) ATM: ATM technology was selected to support B-ISDN by international standards bodies such as the ITU-T, ETSI, and ANSI. Although originally designed for WAN's, it is also suited for LAN's and MAN's and therefore is the perfect vehicle for their seamless integration. ATM is basically virtual circuit switching enhanced by the virtual path concept. It uses small, fixed-size cells (53 bytes, five for header and 48 for payload) and simple communication protocols to reduce per-cell processing and to speed up switching.

In virtual circuit switching, a virtual circuit routing table is required at each intermediate node of the path to direct each packet (known as a cell in ATM terminology) of a virtual circuit (or virtual channel) to the appropriate outgoing link. Since there may be a large number of virtual channels, and many of them may have the same origin and destination, one may reduce the size of routing tables in the transit nodes by grouping those virtual channels sharing the same origin-destination pairs into a virtual path. For example, in Fig. 30, virtual channels VC1, VC2, and VC3 are grouped into virtual path VP1; VC4 and VC5 are grouped into VP2; and VP3 contains only one virtual channel, namely, VC6. The packet header includes a VPI and a VCI. A connection or call is assigned a virtual channel. The end node uses the VCI of the cell to direct it to the corresponding terminal. At the transit nodes, the VPI of the cell provides enough information to direct this cell to the corresponding path in the network.

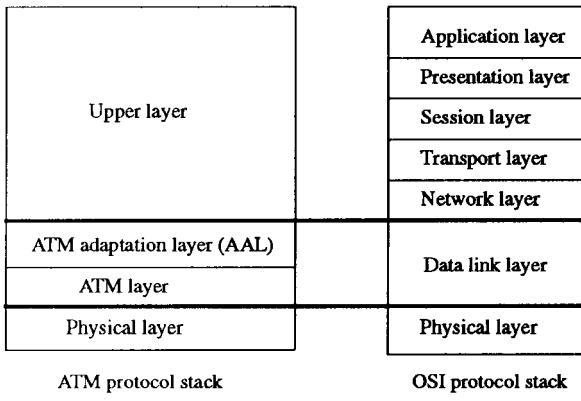


Fig. 31. ATM protocol stacks versus OSI protocol stacks.

The ATM network has a layered structure allowing multimedia traffic to be mixed in the network. It includes the upper-layer, AAL, the ATM layer, and the physical layer. Fig. 31 shows the relationship between the ATM layer structure and the OSI protocol stacks, and Fig. 32 shows the role of the ATM layered structure in the ATM network.

- The upper layer includes higher layer protocols such as TCP, RTP, and Xpress Transport Protocol [106].
- The AAL layer adapts the upper layer protocols to the ATM format. It inserts or extracts user information as 48-byte payloads. The AAL layer consists of two sublayers: the CS and the SAR. The CS converges different types of user traffic and encapsulates/decapsulates data flow to and from the SAR sublayer. The SAR sublayer in the sender segments data into a series of 48-byte cells, and in the receiver reassembles cell sequences back to the original data.
- The ATM layer adds or strips the 5-byte header to or from the payload. For the sender, it takes 48-byte data from the AAL and adds the 5-byte header that contains routing information to ensure that the cell is sent on the right connection; for the receiver, it strips the 5-byte header and passes the 48-byte payload to the corresponding AAL.
- The physical layer defines the electrical characteristics and network interfaces and places ATM cells into the transmission medium. The ATM network is not tied to a specific type of physical medium but is optimized to work with the SONET.

CCITT and other standards bodies have identified five classes of ATM services.

- 1) *Class A*: supports CBR traffic, in which the source generates a constant number of bits per unit time. Examples include continuous media traffic such as voice and uncompressed video. This type of traffic is tolerant of some packet losses but very sensitive to delay jitter. Bandwidth required is moderate for CBR voice and very high for uncompressed video.

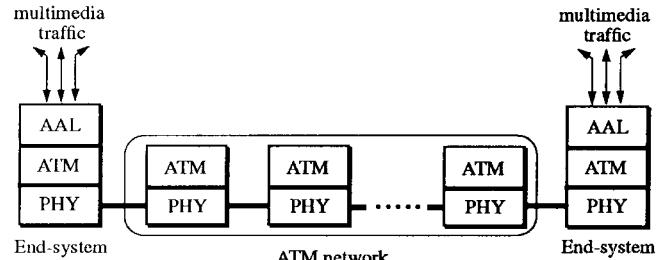


Fig. 32. ATM network.

- 2) *Class B*: supports VBR stream traffic, in which the source generates a variable number of bits per unit time. This is exemplified by compressed video²² such as MPEG. Like CBR, it is sensitive to packet delay variations. In addition, since there are less redundancies in compressed data, it is not tolerant of packet losses. Bandwidth required for video is very high. The traffic is highly bursty and exhibits multiple time scale variation, with the bit-rate requirements varying over milliseconds (within one frame of the video), over seconds (between frames), and over minutes (between scene changes).
- 3) *Class C*: supports connection-oriented framed data.
- 4) *Class D*: supports connectionless (datagram) framed data (e.g., SMDS).
- 5) *Class X*: supports data traffic requiring raw cell delivery using proprietary, nonstandardized protocols.

Note that for Classes A and B, a timing relation is required between the sender and the receiver, while for the other classes, a strict timing relation is not required. Classes C, D, and X correspond to VBR services. To support these five classes of services, four types of AAL formats have been developed. AAL Type 1 supports Class A, Type 2 supports Class B, Type 3/4 supports Classes C and D, and Type 5 supports Class X.

Even with an ATM network, it is difficult to provide true VBR service. One way to satisfy the QoS requirement of VBR traffic is to allocate bandwidth equal to the peak traffic rate. This will give a hard QoS guarantee but is very inefficient. One can exploit the statistical multiplexing of ATM networks and allocate a bandwidth equal to the mean traffic rate, but this will only give a statistical guarantee of the QoS. Another solution is to use buffers to smooth or shape the traffic so that it approximates CBR traffic [107]. In particular, to accommodate the multiple time scale burstiness exhibited by compressed video, a quasi-static bandwidth allocation scheme [108] may be used. In this scheme, developed for VOD systems, certain bandwidth is initially allocated to the connection. As the traffic fluctuates, the amount of video in the video buffer changes. The video buffer is monitored, and if buffer overflow or underflow is

²²Compressed video may also be sent as CBR traffic with less resolution for a complex scene and higher resolution for simple scenes, but this is less desirable than VBR video.

imminent, a request will be made to change the bandwidth allocated to the connection. In addition, since VBR traffic is sensitive to packet losses, one needs to have an admission control scheme to limit the packet losses or employ error-recovery schemes. For example, the receiver may attempt to reconstruct the missing data through temporal or spatial interpolation [109].

3) The Integration of IP and ATM: The two major contenders as the protocol for global internetworking are IP and ATM. There are major obstacles to either's success. The strength of IP is the large installed base. It is the protocol used on the Internet, which is the biggest and fastest-growing computer network in the world. The Internet is a packet-switched network and basically provides connectionless and best-effort transport services. The major problem is that the best-effort delivery of messages does not support the various QoS required in a true integrated services network. In addition, the existing Internet has very limited bandwidth, although there are plans to upgrade it to higher bandwidth. The development of IP multicast, resource reservation protocols, and real-time transport protocols allows the Internet to provide limited forms of multimedia services. Even with the upgraded bandwidth, the Internet will be inadequate for many multimedia applications, such as VOD.

On the other hand, the ATM network is connection oriented. The strength of ATM lies in the possibility of setting up connections with a class of service that matches the requirements of the application involved. The planned bit rates and bit error rates also meet the requirements of real-time transmission of audio and video streams. ATM works best in an environment where everyone uses ATM, however, and the major obstacle is the large installed base of non-ATM equipment. Whether ATM or IP or another protocol will eventually emerge as the global internetworking protocol, only time will tell. In the meantime, the Internet community is developing resource reservation protocols such as RSVP [86] that give QoS guarantees. In addition, IP over ATM has been successfully deployed [104]. The future global internetworking protocol will likely be one that supports QoS guarantees and may include both ATM and IP technologies.

D. Multimedia Operating System

OS's manage computer resources (e.g., CPU, memory, I/O devices, etc.), hide the physical characteristics of the underlying hardware, and provide an efficient and convenient environment for end users. A multimedia OS extends the functionalities of traditional OS's to accommodate multimedia data manipulation so as to provide an environment for real-time support while simultaneously running traditional applications efficiently. The major concerns include real-time processing and QoS-based resource management. Together they ensure that the QoS requirements of multimedia applications are met and provide intelligent overload control for transient misses of playback deadlines. A multimedia OS may be developed as an extension of a traditional OS or constructed from scratch using the microkernel architecture

(e.g., Chorus [110], Real-Time Mach [111]). In any case, it should provide CPU management, memory management, I/O management, and file system management.

1) CPU Management: Real-time processing can be achieved through efficient real-time scheduling. Real-time schedulers schedule tasks to make their deadlines. In the context of continuous media, a deadline can be the acceptable playback time of each frame, and therefore it is a soft deadline and appears periodically. The challenges of multimedia scheduling are due to two conflicting goals [112].

- 1) Nonreal-time processes should not suffer from the execution of real-time processes, since multimedia applications equally depend on discrete and continuous media data. Thus, real-time processes cannot monopolize all resources.
- 2) Real-time processes should be allowed to preempt nonreal-time processes or other real-time processes with lower priorities.

The most important real-time scheduling approaches include EDF and rate-monotonic scheduling [113]. With EDF, each task is preemptive and is assigned a priority according to the deadline. The highest priority is assigned to the job with the earliest deadline, and tasks are executed in priority order. When a new task arrives, the scheduler recomputes the priorities of all pending tasks and then reorganizes the order of processing. If the new arrival has the highest priority, the task being executed is preempted and the new task gets served immediately. The interrupted process is resumed later from the interruption point. Otherwise, the new task will be put in an appropriate position.

With rate-monotonic scheduling, each task is preemptive and is assigned a priority according to the request rate. The highest priority is assigned to the job with the highest rate. In contrast to EDF, such assignment is performed only at the connection establishment time and is maintained throughout the lifetime of the connection. For preemptive, periodic tasks, rate-monotonic scheduling is optimal in the sense that "no other static algorithm can schedule a task set that the rate-monotonic algorithm cannot also schedule" [112].

Comparing these two algorithms, EDF is more dynamic. It has to be executed frequently and thus incurs higher scheduling overhead. The advantage is that it can achieve processor utilization up to 100%. A rate-monotonic algorithm, on the other hand, is static since the priority assignment is only calculated once. Since the priorities are assigned according to the request rate, more context switches occur in rate-monotonic scheduling than in EDF. The worst case upper bound of the process utilization is about 69% (or $\ln 2$) even though, on the average, the utilization far exceeds 80%. However, because it has no scheduling overhead and is optimal for periodic jobs, rate-monotonic scheduling is suitable for continuous media applications.

2) *Memory Management*: The memory manager allocates memory to processes. Continuous media data are typically very large in size and require stringent timing requirements. Traditional virtual memory with demand paging is inappropriate for such real-time processes since page faults incur too much swapping time between main memory and disk and may affect the real-time performance. One solution is to avoid swapping and to lock continuous media data in memory during the process [112]. This approach, however, may affect resource utilization. Another method is to prefetch such data in a timely fashion [114], taking advantage of their periodicity. Other important practical implementation techniques include using scatter buffers and passing pointers. With scatter buffers (or scatter loading), the address space of a process is loaded into possibly discontiguous regions of memory. This tends to be more space efficient than loading into a single contiguous region (since it may not be available) but may result in fragmentation. With passing pointers, objects are passed by reference rather than having to pass the objects themselves. Again, this may result in more efficient usage of memory space.

3) *I/O Management*: The main function of the I/O subsystem is to transfer multimedia information between the main memory and the network adapter or multimedia peripherals (camera, loudspeaker, CD-ROM drive, microphone, disk, keyboard, monitor, etc.). The important issues include device management, interrupt latency, and real-time transmission. Device management integrates all hardware components and provides a uniform interface for the control and management of these devices. It shields the machine dependency by presenting the physical devices as abstract device drivers. Multimedia applications are I/O intensive. The continuous media frames will interrupt the kernel frequently and lower the system throughput. There are three strategies to alleviate this problem: changing the internal structure of the kernel to make it highly preemptable, including a set of safe preemption points to the existing kernel, or converting the current kernel to a user program and running it on top of a microkernel [114]. Real-time I/O is necessary for most multimedia applications to ensure the continuity of each stream and the synchronization of multiple related streams. The problem is centered around the I/O bus bandwidth. Compared to the performance improvements of processor and memory, predicted to double every one to two years, the improvements in I/O subsystems lag far behind [115]. In addition, data must traverse a long path (e.g., SCSI bus and system bus) before reaching their destination. This may even incur bus contention [116]. With advances in networking and communication technologies, it is possible to achieve network bandwidth well above a gigabit per second, and the network I/O becomes the bottleneck that limits overall system performance. The focus, therefore, is to improve I/O system throughput.

4) *File System Management*: The file manager is responsible for making efficient use of the storage capacity and providing file access and control of stored data to the users. Multimedia file systems demand additional real-time

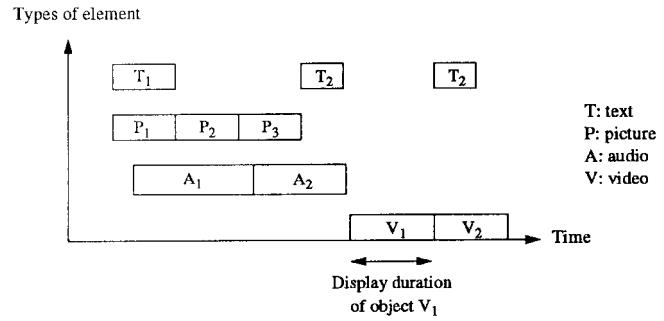


Fig. 33. The temporal relationships between objects in a multimedia scenario.

requirements to handle continuous media streams. We will examine this issue in more detail in Section IV-F.

E. Synchronization

Multimedia synchronization²³ attempts to ensure the desired temporal ordering among a set of multimedia objects in a multimedia scenario, where “multimedia scenario” denotes the temporal semantics of a multimedia session [117]. Fig. 33 shows an example of the temporal relationships in a multimedia presentation scenario. The goals of multimedia synchronization mechanisms are to preserve the playback continuity of media frames within single continuous media and the desired temporal dependencies among multiple related data objects, given that user interactions are allowed [118]–[120]. Temporal dependency may be either *implicit* or *explicit*. For implicit dependency, the timing relationship is captured at the time of media acquisition. One such example is lip synchronization, which refers to the requirement that the voice should match the movement of the lips of the speaker. Explicit dependency means that the time relationship is created explicitly. A slide show with pictures and annotated text is an example.

Multimedia synchronization differs from traditional synchronization mechanisms, such as monitor and semaphore, in that it imposes real-time constraints (i.e., deadlines) on the event, and differs from real-time systems in that such deadlines are predetermined, periodic, and soft (i.e., no disaster is caused if the deadlines are missed). Missing deadlines in multimedia presentation may annoy users with synchronization anomalies, such as hiccups in audio or jerkiness in video. The correction of such synchronization anomalies includes dropping, blocking, duplicating, or resampling frames at the presentation site to resynchronize the playback [1]. For example, restricted blocking [119] repeats the display of the last frame or takes alternative action (such as playing music) to fill up the time in order to resynchronize the presentation. It has been shown that human beings are less tolerant of audio errors than of video errors. Therefore, in an integrated services network in which the audio stream is competing with the video stream for

²³The broader meaning of multimedia synchronization includes the temporal (i.e., timing relationship), spatial (i.e., the layout structure), and logical relationships in a multimedia scenario [1]. In this section, we focus on the temporal aspect. Readers can refer to [75] and [76] for standardization efforts on all three aspects.

network resources, higher priority should be accorded the audio stream.

In a distributed system, the enforcement of such coordinated presentation becomes more complicated because for each stream, random delays may be incurred along the data path: data retrieval in the storage system, data transmission in the network, and data processing in the host computer. Media sources are widely distributed, and different media located in remote sites may be required to be played back in a coordinated fashion at the same or at multiple sites on the network. Without proper orchestration, the existence of random delays among multiple related streams may result in synchronization anomalies, such as the loss of lip synchronization. Another associated problem is the synchronization accuracy between the clocks of the sender and the receiver when global timing knowledge is used. Currently, the Internet uses the NTP [121] to solve this problem. Ramanathan and Rangan [122] proposed a protocol that employs adaptive feedback techniques to synchronize multimedia presentation of distributed sources in the absence of a global clock.

The possible communication configurations may be point to point, multipoint to point, point to multipoint, and multipoint to multipoint, corresponding to a single source to a single destination, multiple sources to a single destination, a single source to multiple destinations, and multiple sources to multiple destinations, respectively [2], [123], [124]. In any case, synchronization is best performed at a point near the destination to eliminate the random delays due to the transmission and the interstream coordination [125]. Synchronization messages can be transmitted in several different ways [1], including transmitting a complete synchronization message prior to starting multimedia data delivery, using separate transmission channels to transport data and synchronization messages [126], or time stamping each data frame of related streams.

According to the synchronization granularity, there are three levels of synchronization.

- 1) *Intramedia synchronization* [124], [127]: preserves the temporal relationship between consecutive frames and the continuity of playback within a single media stream. The basic idea is to satisfy the latency and throughput constraints of the media and to smooth the delay jitters experienced in the storage system, network, and host system by using real-time protocols (including real-time retrieval, transport, and processing) and providing sufficient buffers along the data path.
- 2) *Intermedia synchronization* [122], [128]–[135]: coordinates the different media streams to achieve the desired temporal relationships among them. It can further be categorized as *continuous synchronization* (or *live synchronization*), which preserves the implicit temporal relationships, and *event-driven synchronization* (or *synthetic synchronization*), which maintains the explicit specified temporal relationships. Intermedia synchronization is provided by the orchestration

layer (shown in Fig. 22) of the QoS reference model, which performs end-to-end orchestration to achieve interstream coordination for a meaningful presentation. The communication configuration is multipoint to point, such as multimedia document retrieval with different types of media stored in different databases.

- 3) *Interparty synchronization* [136]: maintains the intramedia and/or intermedia synchronizations among different participants at distributed locations. It is a point- or multipoint-to-multipoint configuration. Examples include remote learning (point to multipoint) or video conferencing (multipoint to multipoint). To provide this synchronization, multicast operations must be supported by the underlying network.

General multimedia synchronization can be divided into two levels: temporal specification and temporal synchronization [117], [137], [138]. *Temporal specification* is the upper level. It models the temporal relationships among the set of related abstract multimedia objects of a presentation scenario, where an abstract object denotes a logical unit of media data such as a video object or a text object. *Temporal synchronization* is the lower level. It translates the temporal specification of a presentation scenario into the desired presentation scheduling sequences and enforces the playback deadlines despite the indeterministic delays due to the server, the network, and the host system. We now describe these two levels in more detail.

1) *Temporal Specification*: Temporal specification captures the temporal relationships among multimedia objects in a multimedia presentation scenario. A good temporal-specification mechanism must be flexible enough to deal with the problems of human perception of synchronization in a wide range of applications. In particular, it must reflect:

- required real-time semantics and the precedence relations among objects;
- seamless integration of discrete and continuous media objects;
- absolute and relative timing relationships. The absolute temporal relationship indicates the precise timing relationships among the objects, while the relative temporal relationship refers to the incomplete or fuzzy timing relations among the objects;
- user interactions such as random access, fast-forward, pause, etc. in a multimedia presentation, and nonsequential navigation in hypermedia.

Temporal relationships among different objects can be represented by a graphical-based (e.g., Timed Petri net²⁴ [131], [142]–[144], flow graph [137], [145]), language-based [146]–[150], or hybrid [151] description. Graphical-based specification takes advantage of pictorial visualiza-

²⁴Timed Petri net [139]–[141] is a bipartite directed graph with two sets of nodes: places (usually denoted by circles) and transitions (usually denoted by bars). A transition fires if all input places contain a nonblocking token, at which time a token is removed from each input place and deposited to each output place. A token added to an input place is blocked for a duration assigned to this place.

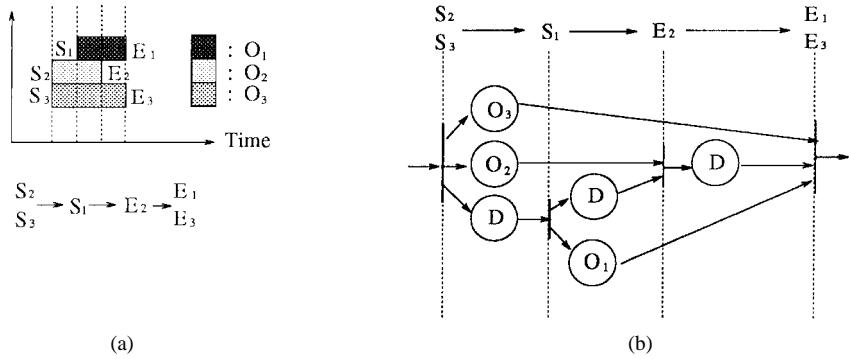


Fig. 34. An example of synchronization-point-based modeling.

tion. Language-based methodology exploits the power of flexible expression of language, which enables ease of inclusion of user interactions. Note that temporal specification focuses on the semantics of the temporal relationships among the objects without explicit reference to time. Temporal operations, such as scaling (i.e., extending the time duration), inverting (i.e., exchanging the start and end relations), or shifting (i.e., moving an offset in the time domain), can be applied when mapping from the virtual-time coordinates back to the real-time axis. The semantics of temporal relationships can be specified via several approaches.

a) *Hierarchy-based modeling*: The hierarchical approach [126] organizes a multimedia scenario as a tree structure, where nodes represent the temporal relations and leaves are the corresponding media objects, user input, or delays. This model is simple and easy to implement. The main temporal relations, however, are based on serial and parallel relationships, and objects are only synchronized at the beginning or end points. This restricts the usability of this model since some scenarios cannot be represented, such as “annotating text during the display of a piece of music.”

b) *Interval-based modeling*: The conceptual model of time may be based on instants or intervals [117], [131], [152]. A time instant is a certain point in time with zero length, such as 10:15 a.m. An interval of time is denoted by $[a, b]$ with length $b - a$, where a is the start point and b is the end point. The playback duration of an object can be represented as a time interval. Interval-based modeling models the binary (or n -ary) relationships between two (or n) time intervals. There are 13 distinct ways to describe the relations between any two intervals [153]. Little and Ghafoor [131], [154] used Object Composition Petri Net, which is an extended timed Petri net, and Li *et al.* [137] used Time-Flow Graph. Wahl and Rothermel [155] extended the 13 basic binary relations to 29 and defined 10 operators to handle these interval relations.

c) *Synchronization-point-based modeling*: Instead of relying on the playback intervals between two objects, the synchronization-point approach [143], [156] directly captures the timing relationship among a set of objects and synchronizes multimedia objects based on synchronization points, which are the start and end points of objects. The

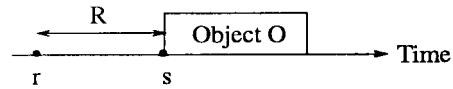


Fig. 35. Timing relationships among s , r , and R .

ordering of synchronization points is the minimal record of the temporal relationships among these objects because removal of any information from this ordering results in a loss of the corresponding relationship [156]. In addition, there is a one-to-one correspondence between a multimedia scenario and the partial ordering of the synchronization points. Let S_i denote the start point of object i and E_i the end point of object i . Assume that we have three multimedia objects O_1, O_2 , and O_3 and a multimedia scenario shown in Fig. 34(a). The ordering of the synchronization points of O_1, O_2 , and O_3 is

$$S_2 \rightarrow S_1 \rightarrow E_2 \rightarrow E_3.$$

If one takes away the arrow between S_1 and E_2 , there is no way to distinguish the cases of O_1 starting before, after, or at the same time as O_2 ends. In Fig. 34(b) this ordering of the synchronization points is represented as a Petri net.

2) *Temporal Synchronization*: Temporal synchronization renders the temporal specification into a set of playback schedules and retrieval schedules²⁵ and enforces the playback deadlines upon the flow transport mechanism in the communication system. To display an object O at time s (i.e., *the playback deadline*), the presentation site sends a request to the corresponding media server at *request time* r , which is R time units prior to s . R is the *response time* to account for possible experienced end-to-end delays and packet losses. The timing relationships among the playback deadline s , the request time r , and the response time R are given by $r = s - R$ and are depicted in Fig. 35. The possible experienced end-to-end delays are contributed by:

- 1) server site (D_{ser}), including queuing delay and retrieval delay (i.e., seek, latency, and data transfer);

²⁵A playback schedule is the set of playback deadlines of multimedia objects in a multimedia scenario, where playback deadline is the latest starting time for a meaningful presentation. Retrieval schedule is the set of request deadlines to the server. It accounts for the end-to-end delays and ensures that the playback deadline is not missed.

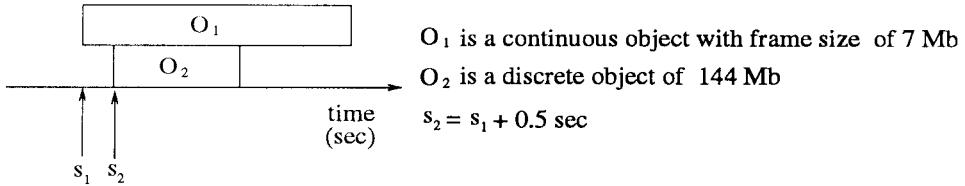


Fig. 36. An example of the request ordering of O_1 and O_2 .

- 2) network (D_{net}), including round-trip queuing delays, propagation delays, and transmission delays;
- 3) request site (D_{req}), including queuing delay and processing delay. Note that for continuous media objects, the intramedia synchronization delay d_{syn} (given below) should also be included, i.e., $D_{\text{req}} = d_{\text{que}} + d_{\text{proc}} + d_{\text{syn}}$ for continuous media objects and $D_{\text{req}} = d_{\text{que}} + d_{\text{proc}}$ for discrete objects, where d_{que} and d_{proc} are the queuing and processing delays, respectively.

For a continuous media object, which is a stream of media frames, finer-grained synchronization is required to keep the continuity of the stream. In an ideal system, no delay variations are incurred by the transport networks and processing systems, and no frame is lost during the transmission. This means that each media frame experiences the same delay and is ready to be played according to the specified playback schedule. However, in practical systems, each media frame may experience different delays. For media stream i , intrastream synchronization can be achieved by delaying the playback deadline of the first media frame by τ_1^i , where [124]

$$\tau_1^i \geq d_{\text{max}}^i - d_{\text{min}}^i$$

and d_{max}^i and d_{min}^i are the maximum and minimum network delays of stream i , respectively. Note that $d_{\text{max}}^i - d_{\text{min}}^i$ is the *intramedia synchronization delay* (d_{syn}) of the continuous media object. Thus, the playback deadline for an object can be met so long as its response time R satisfies the relationship

$$R \geq D_T$$

where D_T , the end-to-end delay defined as the time from when the request for a particular object is issued to when the object can be played at the request site, is given by

$$D_T = D_{\text{ser}} + D_{\text{net}} + D_{\text{req}}.$$

Research on the calculation, measurement, and estimation of the end-to-end delay can be found in [122], [129], [134], [157], and [158].

The response time of each media object is determined by the characteristics of the transport networks, the processing systems, and the media object itself. Thus, even if, say, two media objects start simultaneously, the corresponding request time may be different. Moreover, the request ordering of a set of objects may not be the same as the playback ordering of the objects. For example, in Fig. 36, assume that the channel capacity is 155 Mb/s, the object O_1 is a

continuous media object with a frame size of 7 Mb, and O_2 is a discrete media object of 144 Mb, and that these two media objects experience the same delays of d seconds except for the transmission delay. We also assume that the intramedia synchronization delay for O_1 is 5×10^{-6} s. The response time for O_1 is $(d + 7/155 + 5 \times 10^{-6})$ s, and that for O_2 is $(d + 144/155)$ s. Note that the number of frames in O_1 is irrelevant. So long as we delay the first frame by the intramedia synchronization delay and we have the first frame in buffer, the subsequent frames will arrive in time to be played back. Therefore, for a playback deadline of O_1 0.5 s before that of O_2 , the request times are

$$\begin{aligned} r_1 &= s_1 - (d + 0.045 + 5 \times 10^{-6}) \text{ s} \\ &= (s_1 - d - 0.045005) \text{ s for } O_1 \end{aligned}$$

and

$$\begin{aligned} r_2 &= (s_1 + 0.5) - (d + 0.93) \text{ s} \\ &= (s_1 - d - 0.43) \text{ s for } O_2. \end{aligned}$$

Since $r_2 < r_1$, the request site should request O_2 ahead of O_1 by $r_1 - r_2 \simeq 0.395$ s. Hence, from the playback schedule $S = \{s_1, s_2, \dots, s_n\}$ and the corresponding response times $\mathcal{R} = \{R_1, R_2, \dots, R_n\}$, the retrieval schedule $Q = \{r_1, r_2, \dots, r_n\}$ can be determined by $Q = S - \mathcal{R} = \{s_1 - R_1, s_2 - R_2, \dots, s_n - R_n\}$. Without loss of generality, assume the objects are labeled such that $s_1 \leq s_2 \leq \dots \leq s_n$. The request ordering can be determined by sorting in a nondescending order the request schedule Q to yield an ordered retrieval schedule $Q_{\text{ordered}} = (r'_1, r'_2, \dots)$. For example, in Fig. 36, $Q_{\text{ordered}} = (r'_1, r'_2) = (r_2, r_1)$.

The time duration from when a user initiates a presentation request to when the presentation begins is called the *set-up time*. Fig. 37 shows the set-up time in a presentation. The set-up time T must be long enough to compensate for the system delays and to achieve intermedia synchronization. The set-up time therefore is not necessarily equal to $\max\{R_1, R_2, \dots, R_n\}$ where O_1, \dots, O_n are the start objects. T is largely dependent on the request time of the first request objects in the request ordering. Let the playback schedule $S = (s_1, s_2, \dots)$ and the ordered request schedule $Q_{\text{ordered}} = (r'_1, r'_2, \dots)$. The set-up time T is then $T = s_1 - r'_1$.

The set-up time determines the earliest playback time of the corresponding scenario, which in turn determines the absolute playback deadlines of the subsequent objects. The earliest playback time is the earliest time a presentation can start after it is initiated, and it is also the actual playback deadline of the start objects. Combining the earliest playback time with the relative playback schedule

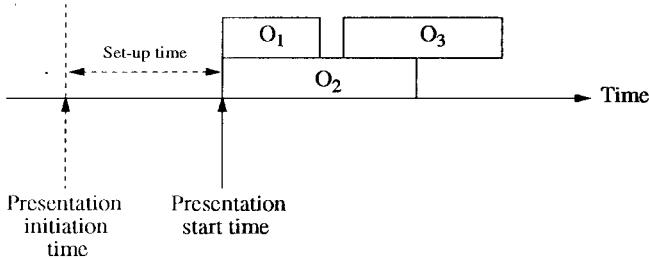


Fig. 37. The set-up time.

gives the absolute playback schedule. In particular, given that the presentation is initiated at t_0 , the earliest playback time is $t_0 + T$. Then the absolute playback schedule S_{abs} can be obtained by substituting s_1 with $t_0 + T$ in the relative schedule S . Note that there are n response times and one set-up time for a presentation with n media objects.

F. Multimedia Server

Distributed multimedia applications impose major challenges in the design of multimedia servers,²⁶ including:

- massive storage for a huge amount of multimedia information;
- real-time storage and retrieval performance to satisfy the continuous retrieval of a media stream at specified QoS guarantees;
- high data throughput to serve a large number of simultaneous user requests efficiently.

A storage hierarchy may be used to increase the cost effectiveness of the storage. To ensure playback continuity, the simplest approach is to employ a dedicated disk head for each stream [159]. The disk transfer rate is typically greater than the playback rate of an individual video stream. This approach, therefore, wastes system resources, is not cost effective, and limits the number of user requests to the number of disk heads. Other promising approaches allow a disk head to be time shared among multiple streams, employ real-time storage and retrieval techniques to satisfy the real-time constraint, and adopt admission-control mechanisms [83], [160], [161] to ensure that the QoS guarantees of the existing streams will not be degraded by accepted new requests. Because any given storage system has a limit as to the number of files it can store and the number of user requests it can serve, a server can effectively increase effective data throughput and storage space with the use of an array of disk drives.

Other issues associated with server design include dynamic load balancing [162] and fault tolerance [163]. Dynamic load balancing attempts to balance the load among multiple disks due to the different popularity of objects (e.g., a new movie may be accessed much more often than a documentary), user access patterns (e.g., for VOD services, subscribers tend to watch programs at night rather than during the day), and device utilization. The goal is to avoid load imbalance and to have optimal data placement across

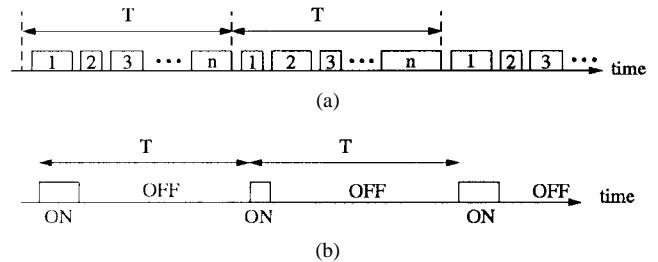


Fig. 38. The on/off process of an individual stream. (a) Service process of the disk head. (b) Request process of video stream $i, i = 1, 2, \dots, n$.

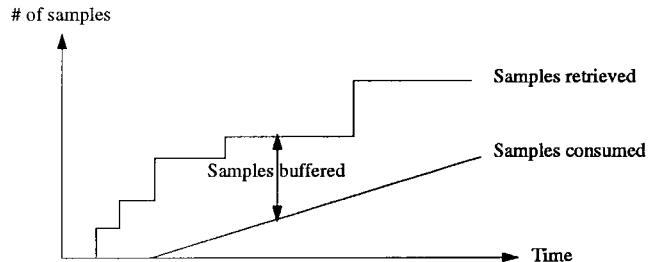


Fig. 39. The relationship between data retrieval, consumption, and buffering.

the storage hierarchy (e.g., a “hot” video is stored in a disk drive while an unpopular video is archived in tertiary storage) and across multiple disks (e.g., popular movies may be replicated in multiple disks while unpopular movies are stored singly) to achieve the maximum utilization of both space and bandwidth in the storage system. A fault-tolerant design prevents playback interruption due to disk failures and improves the reliability and availability of the storage system [163]. In a continuous media server, disk failure does not cause data loss because data is archived in tertiary storage. Instead, it interrupts the retrieval progress and results in discontinuity in playback. Solutions include mirroring [164] and parity [165] approaches, in which a part of the disk space is allocated to storing redundant information.

1) *Disk Behavior*: Basic disk behavior in the server is as follows. A disk incurs a seek and latency time to locate the data of a particular media stream and transfers a batch of data before serving the next stream. Therefore, the disk is multiplexed over a set of concurrent streams, as shown in Fig. 38(a). Here T , called the period time, denotes the time taken to serve all the streams in one round. Each individual stream is served in an alternating on/off process, as shown in Fig. 38(b): it is “on” when data is transferred by the disk and remains “off” until its next turn [83]. A buffer is allocated for each stream at the user site to overcome the mismatch between the data transfer rate and the playback rate of the media stream. Fig. 39 shows the relationship between frame retrieval, consumption (i.e., playback), and buffering [166].

To avoid starvation during playback, i.e., a lack of data available in the buffer to be played, the disk head must transfer another batch of data to the buffer before

²⁶Here we focus on the continuous media server.

the previous batch has been consumed. This constraint implies that there exists an upper bound on the number of concurrent requests a disk can handle. Since the disk transfer rate is higher than the playback rate, the disk head is not allowed to transfer another batch too soon; otherwise, the stream buffer will overflow. These two constraints and the desire to minimize the buffer requirement place a great challenge on the design of the server.

2) *Real-Time Storage and Retrieval*: Continuous media streams demand timely data retrievals for playback continuity. There are three major approaches to provide such real-time support of continuous media: frame grouping, data placement, and real-time disk scheduling. Frame grouping reduces the access latency per frame by allowing a group of frames to be retrieved as an atomic unit. Data placement exploits the characteristic of continuous media and employs a specific disk layout structure to reduce seek and latency overhead and to gain timely access [167]. Disk scheduling places no specific requirement on data organization but develops real-time disk scheduling with data buffers for real-time retrieval.

a) *Frame grouping*: From a storage and retrieval viewpoint, it is advantageous to group video frames into segments. A video segment is an atomic unit of retrieval from a disk. Since the disk retrieval overhead (seek and rotational latency) does not depend on the size of the video segment, a large segment improves disk retrieval efficiency. However, a large segment warrants large buffers. Thus, there exists a tradeoff between disk retrieval efficiency and buffer requirement. Depending on the application, a certain segment length (defined as the number of frames per segment) may be preferred over others. When interframe coding is employed and when segment skipping is resorted to during fast playback, the process of suitably grouping video frames into segments and that of skipping video segments during fast playback is a challenging one. Sengodan and Li [168], [169] describe grouping and retrieval schemes for MPEG frames. These schemes describe how MPEG frames need to be grouped into segments of any desired segment length and which segments need to be skipped to achieve any desired fast playback rate. Several properties and closed-form equations have been provided.

b) *Data placement*: Digitization of continuous media yields a sequence of frames. In the storage system, the continuous server groups frames into segments and stores them on a disk. The goal of data placement is to optimize the disk layout structure by exploiting the characteristics of continuous media, such as continuity, write-once-read-many, and sequential retrievals, so that periodic and timely playback constraints can be satisfied. Example approaches include contiguous and constrained placement.

Contiguous placement stores the successive segments of a media stream consecutively on a disk. This placement strategy takes advantage of sequential stream retrieval and therefore requires only one seek time to locate a particular stream. Increasing the segment size further improves the retrieval efficiency. Contiguous placement, however, suffers from fragmentation due to insertion and deletion. Thus, it

is not suitable for read-write applications but works well with read-only ones.

Constrained placement allows the successive segments of a media stream to be scattered so long as the intersegment separation is bounded by the constraint of playback continuity [170], [171]. The storage pattern of this placement strategy denoted (M, G) (where M is the size of a data segment and G is the intersegment separation) is therefore determined by the playback rate of a stored stream and must satisfy the condition that the playback duration of a segment should exceed the time to skip over a gap and to retrieve the next segment [161], [172], i.e.

$$D_{\text{play}} \geq \frac{M + G}{R_{dt}}$$

where D_{play} is the playback duration and R_{dt} is the disk transfer rate. This placement strategy requires the disk to be sparsely occupied to begin with so that data can be stored in accordance to its storage pattern. Otherwise, an elaborate merging algorithm in which multiple media streams are interleaved should be applied to improve disk storage efficiency.

c) *Disk scheduling*: Traditional disk scheduling is concerned with maximizing throughput by minimizing seek time and latency, or with fairness in usage by avoiding starvation. Such criteria are inappropriate with multimedia real-time constraints. Multimedia disk scheduling needs real-time guarantees. Examples of approaches include Scan-EDF [173], GSS [174], quasi-static retrieval scheme [175], QuIVeR [176], and DIVeR [177].

Scan-EDF combines Scan with EDF, providing both the seek optimization of Scan and the real-time performance of EDF. Data are stored on concentric tracks on a disk, which rotates at a high speed. The disk read/write head is at the end of a disk arm, which moves back and forth just above the surface of the disk in a direction perpendicular to the tracks. Basically, the disk head moves to the target track, waits for the data to rotate around until they are underneath the disk head, and reads or writes the data. Scan is one of the traditional disk-scheduling algorithms. It serves requests in the same order as the data are encountered as the disk arm moves forward. The disk arm retraces back as soon as no pending requests are waiting in the forward direction. Data are transferred in both the forward and the retracing directions. In a variation called C-Scan, no data is transferred during the retracing step. In Scan-EDF, the requests are normally served in EDF order. When there are several requests with the same deadline, these requests are served with Scan scheduling, in which the specific one located nearest to the disk arm in the disk scan direction is served first. The optimization of this algorithm highly depends on the number of requests with the same deadline, and therefore its efficiency relies on how often seek optimization can be applied.

GSS is a variation of Scan. It divides the set of n requests being served into g groups. Groups are scheduled in a fixed order, and individual requests within a group are served with Scan scheduling. If $g = 1$, GSS degenerates

to Scan scheduling, and if $g = n$, GSS becomes round-robin scheduling. Optimization of this algorithm depends on deriving groups to balance the tradeoff between the round duration and the latency of successive retrievals [159].

The video retrieval scheme in [178] uses the scan algorithm for video retrieval of all segments in a service round. A novel shared buffer architecture at the server is proposed for storing segments retrieved from disk. Video retrieved from the disk is stored at designated locations in the buffer, while the transfer of video to the network takes place in a serial fashion from the start to the end of the buffer. The retrieval schemes in [175] and [176] divide a service round into a certain number of groups, and the scan algorithm is used for video retrieval within each group. The quasi-static video retrieval scheme [175] uses the EDF algorithm to assign segments to be retrieved to groups. The QuIVeR protocols [176] describe several simple and efficient ways of assigning segment retrievals to groups. By suitably constraining allowable user playback rates, the protocols achieve a small number of groups within a service round with a large number of segment retrievals within each group. Thus, in addition to achieving a high disk retrieval efficiency, these protocols have an extremely small buffer requirement. They dispense with the server buffer and require a set-top box buffer of merely two video segments per user irrespective of the user playback rate. The DIVeR protocol [177] builds on the QuIVeR protocol by reducing start-up and interactive latencies. All the advantages of the QuIVeR protocol—no server buffer, a set-top box buffer requirement of two segments per user, and a high disk retrieval efficiency—are retained by the DIVeR protocol.

3) *Disk Array*: Each device has a limit on storage capacity and throughput bandwidth. For example, a disk drive with 45 GB of storage and 68 Mb/s of data transfer rate is able to store only about 40 MPEG-1 encoded movies and serve less than 20 video sessions with NTSC quality. Thus, employing multiple disk drives in a storage system is a promising way to dramatically increase the disk throughput and storage space with currently available technology. This approach exploits the transfer parallelism in a disk array and increases the effective transfer rate in proportion to the number of disk drives. The redundant arrays of inexpensive disk (RAID) [179], [180] technique is one such attempt to increase I/O bandwidth.

How to distribute data blocks across multiple disks is an important issue for the disk array. There are three different techniques: disk striping [181], [182], [183], disk declustering [184], and hybrid combination.

a) *Disk striping*: Assume that there are n disks in a disk array. The data blocks in a file are striped in a round-robin manner over n disks. To play back the file, each disk in the array operates synchronously (i.e., each disk head is located at the same position and transfers data together). Therefore, the n disks function as a single disk with n times the transfer rate and storage capacity. For example, if an audio clip X is striped onto 10 disks, the frames $X_0, X_{10}, X_{20}, \dots$ are stored on disk 1, $X_1, X_{11}, X_{21}, \dots$ are stored on disk 2, and so forth. To play back audio X in

the first time period, disk 1 reads X_0 , disk 2 reads X_1, \dots , and disk 10 reads X_9 . All these reading activities happen at the same time. Therefore, the large logical data block transferred is composed of n smaller physical data blocks. This approach works well for large files since it exploits the parallelism of multiple disks.

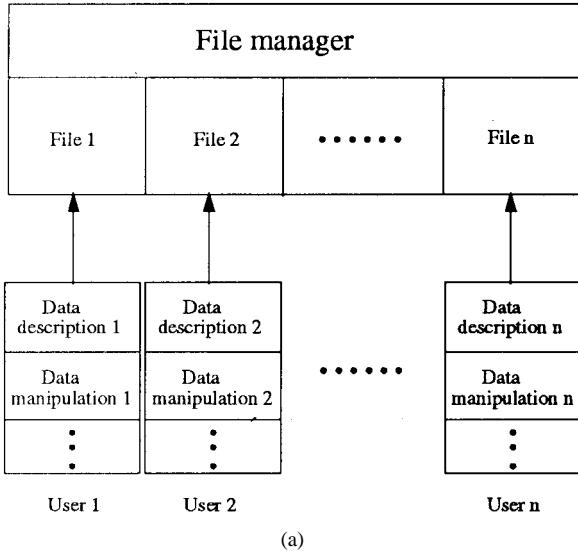
b) *Disk declustering*: Assume that we have the same number of disks and also distribute data blocks in a round-robin manner as in data striping. To play back the file, each disk in the array operates asynchronously and retrieves data blocks in the same round-robin manner. For instance, in the above example, disk 1 reads frame X_0 at the first time period, disk 2 reads X_1 at the second time period, and so forth until all blocks of X are read. This approach exploits the concurrence of multiple disks.

c) *Hybrid combination*: This method combines the two schemes described above, i.e., it scatters data blocks into multiple, independent, and synchronized modules by data declustering, and each module consists of several synchronized disks by using data striping. These independent modules can even be distributed in remote servers over the network.

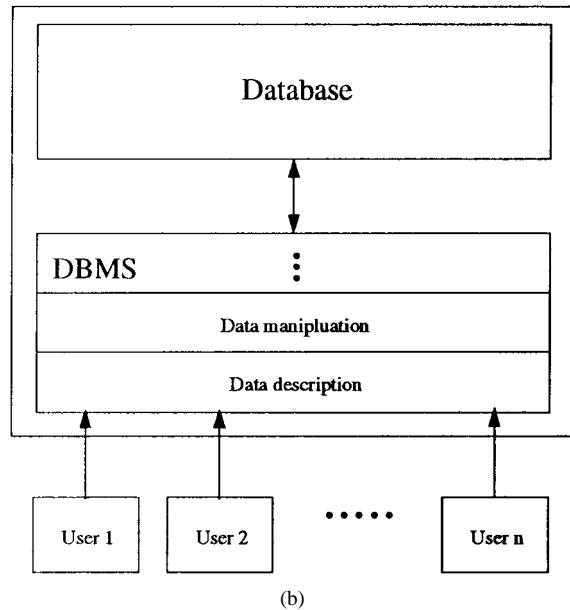
G. Multimedia Database System (MDBS)

An MDBS plays an important role in the management of the exponentially growing multimedia information generated by various applications, ranging from desktop multimedia interactive services to distributed, collaborative, hypermedia systems. As shown in Fig. 40, an MDBS differs from a multimedia file system in that the MDBS provides all users a common interface, high-level query support, and management services to share and to integrate the structured, semistructured, and unstructured data [185] while the file system leaves the responsibilities of formatting, organizing, and managing multimedia data to the users [186]. In addition to the support of conventional database functionalities (such as data independence, data persistency, integrity, reliability, consistency, and security), an MDBS in essence must be an intelligent, integrated object repository [187]. No restriction should be placed on the creation and modeling of how the data is actually used in practice, and users should be allowed to compose data freely in an arbitrary manner. An MDBS should understand how to extract internal components of objects and provide content-based retrievals. It should allow users to share objects or subobjects while also protecting data from unauthorized access. Last, it should be efficient at managing complex data and long-computing-time media functions. The important issues include multimedia data modeling; representation, manipulation, and presentation of data; metadata management; and transaction management.

1) *Multimedia Data Modeling*: The data model sits at the heart of the MDBS. It models the nested logical structure and semantics of multimedia documents, expresses the temporal and spatial compositions of the presentation task, coordinates the sharing of multimedia information among multiple users, and supports the fundamental functionalities for handling digital media, including content-based



(a)



(b)

Fig. 40. Multimedia file system versus multimedia database system. (a) File system. (b) Database system.

retrieval, searching, browsing, and other user interactions. The simple node and link model in hypermedia systems, as well as the flat relational data model, were proven not to be rich and flexible enough to satisfy the productivity and performance requirements of multimedia application development [188]. An object-oriented data model, on the other hand, enables rich semantic relationships among the objects and supports such core concepts as the encapsulation of data structural and behavioral properties into an object, object identity, multiple inheritance, arbitrary data types, and nested objects [189]. As a result, arbitrary media types, arbitrarily large data sizes, arbitrary object structures, and arbitrary storage devices (including magnetic disk, CD-ROM, audio CD, laser-disc, and videotape) can easily be included. Moreover, a clear distinction between modeling (i.e., internal structure) and presentation (i.e., external temporal-spatial composition) of multimedia objects can

be made [188]. Such features drive the desire to employ object-oriented models to meet the special requirements of multimedia applications. Another promising alternative is the object-relational data model, which in essence is a hybrid of the relational and the object-oriented data models. It combines the rich modeling power of the object-oriented model with the structured query language support of the relational model. One notable commercial product of the object-relational model is *Illustra*.

2) Representation, Manipulation, and Presentation of Data: An MDBS provides mechanisms to define and manipulate multimedia information. The DDL allows users to specify the content, structure, and temporal-spatial composition in the presentation of multimedia data objects [81], [190], including arbitrary data types, object structures, QoS parameters, and arbitrarily large data sizes. The concepts of basic and composite objects, the hyperlink relationships, and the specific characteristics and semantics of a given multimedia object are also supported by the DDL. The DML allows users to:

- 1) manipulate data objects with such functions as insertion, deletion, retrieval, modification, query, and browsing;
- 2) share and integrate multimedia information;
- 3) perform nonlinear editing;
- 4) support media-specific user interactions, including VCR-like functions for video or audio such as play, fast-forward, and pause, and specific functions for images, such as display, overlay, condense, and expand.

Unlike structured text data, indexing for digital media data is more difficult due to its unstructured nature [191]. The content-based or visual and/or aural supported query for multimedia information should be provided by the DML. There is much research on devising multimedia query languages [186], [192]–[195]. In addition to query specification, browsing may be used to search for multimedia information when one has incomplete information on what one is looking for, especially in hypermedia applications. Typically, one alternates between two steps: browsing the material to narrow the searching range and precisely searching for a particular target. During browsing, one just glances over the data. Therefore, the representative features of objects (such as in the form of content-based metadata as described below) can be extracted to facilitate browsing [196], [197].

3) Metadata Management: Metadata is the data about data [185]. It contains concise information about the characteristics, location, and QoS parameters of data. Metadata will play a more important role in the MDBS than in the traditional structured text-oriented system. Multimedia information, in addition to text, contains a large portion of visual (e.g., images and video) and aural (e.g., speech and music) data. It is usually very hard to perform an exact match via content-based searches, and even if it is possible, it is very time consuming.

Therefore, additional information, such as the semantics and the structure of the related data, derived or interpreted data, and content-irrelevant data (e.g., attributes of data, creation date, and author), are required. Such information constitutes metadata.

Metadata help the management of large volumes of data, information query and retrieval, material sharing, and directory services. According to the functionalities, metadata may be classified as descriptive, content based, or structural [198]–[202].

a) Descriptive metadata: Descriptive metadata specify the content-irrelevant information about data, such as data type (e.g., text, image, video, audio, or document), attribute (e.g., video with refresh rate of 30 fps, 24-b color, MPEG encoding), material title, file size, program type (e.g., entertainment, education, sports, or news), author name, creation or modification dates, storage location, and other related information. This type of metadata is usually generated manually.

b) Content-based metadata: Content-based metadata specify content-dependent features and are generated automatically or semiautomatically from the information content. This type of metadata is used to assist in content-based query and retrieval. For example, the eigenvalues and eigenvectors extracted from images via the user impression and image content [203] can be used as content-based metadata.

c) Structural metadata: Structural metadata specify the temporal and/or spatial compositions of data. The spatial structure describes the spatial relationship of multimedia information for display and includes the logical and layout structures of the multimedia document, the width and height of an image for display, etc. The temporal composition describes the timing relationship among the related objects. For a coarser-grained temporal specification, it may describe the temporal relationships among multiple objects within a document (such as the voice annotation in a slide show), while a finer-grained temporal specification describes a series of frames within a single multimedia data object. This type of metadata is strongly related to the correct presentation of multimedia information to the users.

4) Transaction management: Transaction management is concerned with such issues as concurrency control, failure recovery, reliability, version control, and replication control to maintain the atomicity, consistency, isolation, and durability of user transactions [204] and to ensure reliable storage of multimedia information and consistent concurrent accesses to these data and their subdata.²⁷ Multimedia data may be arbitrarily large and may have a nested structure for composite objects. The unit of data used in a transaction should be flexible enough to represent logical objects [189] as well as small basic data objects. In addition, multimedia data in a transaction may be time dependent and time independent. The concurrency control manager should satisfy the real-time constraints of the time-

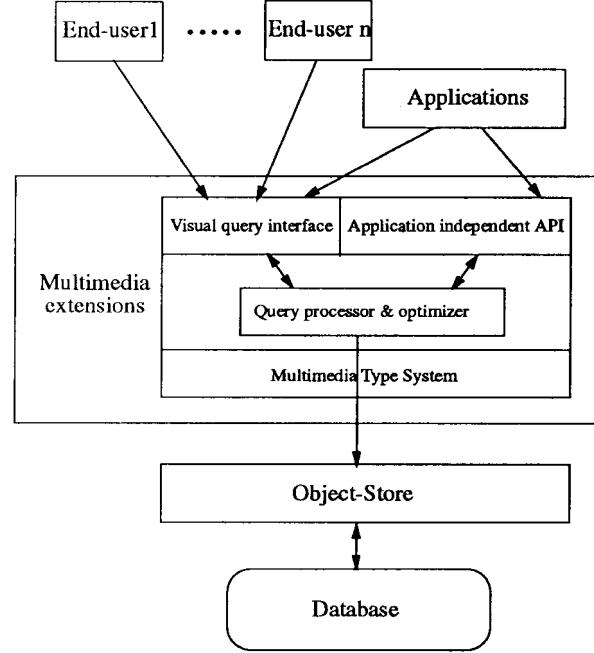


Fig. 41. The conceptual architecture of an MDBS using the extension approach [186].

dependent transactions while keeping the time-independent transactions.

Multimedia transactions should be able to model the characteristic of long duration and interactive database access of multimedia applications. Traditional concurrency control mechanisms, such as two-phase locking and time-stamp ordering, strictly enforce the serializability of transaction executions and are appropriate for short-duration transactions. With such exclusive, serialized access, intolerably long waiting times will be incurred and a large amount of work will be lost when transactions are aborted before being committed [189]. In multimedia transactions, temporary violation of the serializability in execution will not result in big errors, as in traditional business transactions, but may allow flexibility of interactions among cooperative users. This calls for relaxing serializability as the consistency criteria and devising new correctness criteria.

5) System Design Approaches: An MDBS must be capable of capturing the abstraction of the complex structure of multimedia documents in addition to accommodating the heterogeneity of multimedia data. Multimedia information is usually very large and requires the support of similarity-based retrieval for multimedia query and browsing. Without proper extensions, the traditional design approaches are inadequate for the multimedia environment. Typical design approaches include extending the existing database systems or developing a new system from scratch. The extension approach develops a multimedia extension on top of the existing data model or database system. The object-oriented database systems designed for news on demand at the University of Alberta [186] are one such example. The generic architecture of the multimedia database system used for news on demand is shown in Fig. 41. The multimedia-type system is the key component of the multimedia extension

²⁷ Composite multimedia data are composed of several components, and users are allowed to retrieve them in parts and in whole.

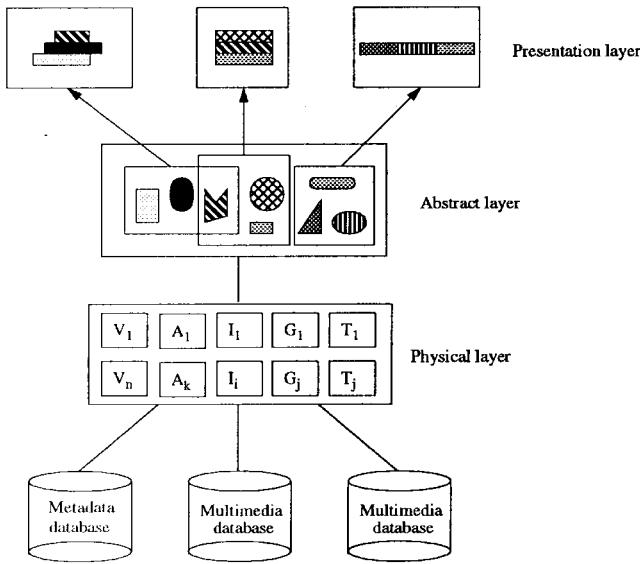


Fig. 42. The conceptual multimedia data model.

and deals with the common multimedia types of data. The query processor and optimizer support the content-based retrievals of images and video and optimize the queries. The visual query interface provides visual access capability with the integration of query and browsing to end users and application developers. The API provides application-independent program interfaces for a number of multimedia applications. The advantage of the extension approach is ease of development. The traditional database systems, however, are optimized for text-oriented and numeric transaction processing. The inherent restrictions and lack of multimedia capabilities of the underlying database management system may introduce inefficiencies and inflexibilities in the management and manipulation of multimedia information. With the limited set of extended functionalities, such multimedia extension is not satisfactory in supporting multimedia applications [205].

The development-from-scratch approach allows one to embed multimedia features into the core of the data model. The conceptual data model may consist of three layers [143], [188], [206]: the presentation (external) layer, the abstract (conceptual) layer, and the physical (internal) layer, as shown in Fig. 42. The physical layer provides location transparency and media transparency of multimedia data. Multimedia data is typically uninterpreted and represented in the BLOB format. They may be centralized in a single database system or distributed among multiple database systems across the network. The physical layer maintains the location description and hides the physical characteristics of media, such as media type and associated attributes, from the upper layer. For example, in audio data, the associated attributes include sampling rate, sample size, encoding method, number of channels, and QoS specification. The abstract layer provides a conceptual view of the real world and creates abstract data objects for the upper layer. The major function is to perform a series of derivations from the raw media, such as extract, expand, or shrink [206]. For

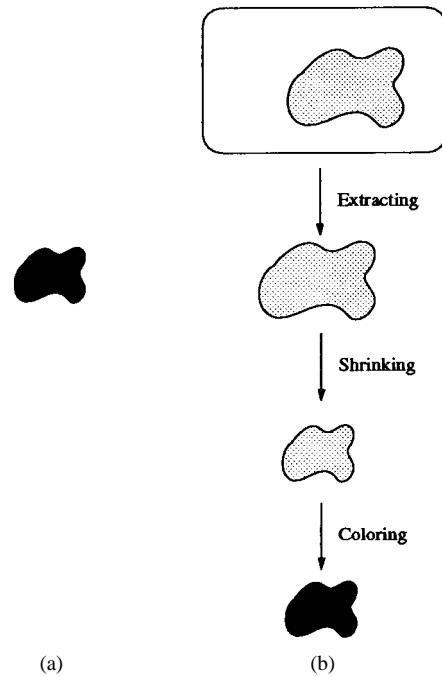


Fig. 43. A series of derivations. (a) Requested object. (b) Original source.

example, Fig. 43(a) shows the requested object from the composition layer. It needs to be transformed three times, i.e., extracting, shrinking, and coloring, from the original source, as shown in Fig. 43(b). Note that if no derivation is necessary, this layer performs an identity function. It allows one to reuse stored data, and the intermediate and resultant abstract objects are allowed to be shared with several external views.

The presentation layer uses the abstract objects from the abstract layer and is concerned with the temporal and spatial composition, document layout structure, and hyperlinks to render a synchronized and interactive presentation. Fig. 44 shows the functional components of a generic architecture of the multimedia database system [196]. The standard alphanumeric database contains non-multimedia, real-world application entities. The multimedia database contains information about multimedia objects and their properties and relationships. The metadatabase stores content-dependent metadata extracted by the feature-processing module and used for content-based retrieval. With the support of processing modules, the database system allows complex content-based queries and acts as an intelligent agent.

V. CONCLUSIONS

Distributed multimedia applications emerge from advances in a variety of technologies, spawning new applications that in turn push such technologies to their limits and demand fundamental changes to satisfy a diversity of new requirements. With the rapid development of the Internet, distributed multimedia applications are accelerating the realization of the information superhighway, creating gigantic potential new markets, and ushering in a new era

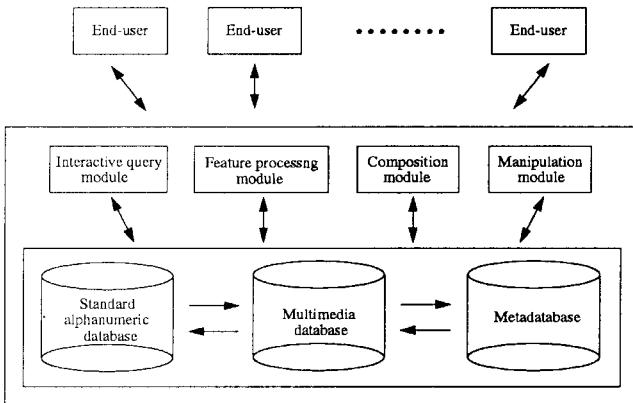


Fig. 44. The generic architecture of an MDBS.

in human communications. Up-to-the-minute information, computer-simulated immersive experience (virtual reality), on-demand services, telecommuting, etc. will be the way of life in the future.

This paper gives an overview of DMS's, identifies the applications, and surveys the enabling technologies. Many efforts have been made in various areas by both the academic and the industrial communities. Many research issues still need to be resolved, however, before distributed multimedia applications can be widely deployed. Such applications consume considerable system resources, require system-wide QoS guarantees to ensure their correct delivery and presentation, and must be provided with user-friendly interfaces. More user control and interactivity are desired. Faster processors and hardware, higher network bandwidths and data compression ratios, and improvements in a variety of related technologies are necessary. Standardization is also important to accommodate the heterogeneity of techniques and to provide portability of applications. The intellectual property rights of the creators of multimedia works must be protected to encourage the creation of such works. Deregulation will accelerate the convergence of the telecommunications, cable TV, entertainment, and consumer electronics industries. Social and human factors should be considered in the development and deployment of new applications. Last, for these applications to be commercially viable, costs will be an important consideration.

ACKNOWLEDGMENT

The authors wish to thank the anonymous reviewers for their helpful comments on earlier versions of this paper.

REFERENCES

- [1] G. Blakowski and R. Steinmetz, "A media synchronization survey: Reference model, specification, and case studies," *IEEE J. Select. Areas Commun.*, vol. 14, pp. 5-35, Jan. 1996.
- [2] B. Furht, "Multimedia systems: An overview," *IEEE Multimedia Mag.*, pp. 47-59, Spring 1994.
- [3] N. Williams and G. S. Blair, "Distributed multimedia applications: A review," *Comput. Commun.*, vol. 17, no. 2, pp. 199-132, 1994.
- [4] "Special Issue on Multimedia Communications," *IEEE J. Select. Areas Commun.*, vol. 8, Apr., 1990.
- [5] "Special Issue on Multimedia Information Systems," *IEEE Comput. Mag.*, vol. 24, Oct. 1991.
- [6] "Special Issue on Multimedia Communications," *IEEE Commun. Mag.*, vol. 30, May 1992.
- [7] "Special Issue on Distributed Multimedia Systems," *Comput. J.*, vol. 36, no. 1, 1993.
- [8] "Special Issue on Multimedia Information Systems," *IEEE Trans. Knowledge Data Eng.*, vol. 5, Aug. 1993.
- [9] "Special section on Multimedia Information Systems," *IEEE Trans. Knowledge Data Eng.*, vol. 5, pp. 545-718, Aug. 1993.
- [10] "Special Issue on Multimedia," *IEEE Comput. Mag.*, vol. 28, May 1995.
- [11] J. Burger, *The Desktop Multimedia Bible*. Reading, MA: Addison Wesley, 1993.
- [12] K. C. Pohlmann, *Principles of Digital Audio*. New York: McGraw-Hill, 1995.
- [13] J. F. Blinn, "The world of digital video," *IEEE Comput. Graph. Applicat.*, vol. 12, no. 5, pp. 106-112, 1992.
- [14] ———, "NTSC: Nice technology, super color," *IEEE Comput. Graph. Applicat.*, vol. 13, no. 2, pp. 17-23, 1993.
- [15] V. O. K. Li, W. J. Liao, X. X. Qiu, and E. W. M. Wong, "Performance model of interactive video-on-demand systems," *IEEE J. Select. Areas Commun.*, vol. 14, pp. 1099-1109, Aug. 1996.
- [16] Y. H. Chang, D. Coggins, D. Pitt, D. Skellern, M. Thapar, and C. Venktraman, "An open-systems approach to video on demand," *IEEE Commun. Mag.*, vol. 32, pp. 68-80, May 1994.
- [17] D. Deloddere, W. Verbiest, and H. Verhille, "Interactive video on demand," *IEEE Commun. Mag.*, vol. 32, pp. 82-88, May 1994.
- [18] T. D. C. Little and D. Venkatesh, "Prospects for interactive video-on-demand," *IEEE Multimedia Mag.*, pp. 14-24, Spring 1994.
- [19] W. D. Sincoskie, "System architecture for a large scale video on demand service," *Comput. Networks ISDN Syst.*, vol. 22, pp. 155-162, 1991.
- [20] J. Sutherland and L. Litteral, "Residential video services," *IEEE Commun. Mag.*, pp. 36-41, 1992.
- [21] "Special Issue on Access to Broadband Services," *IEEE Commun. Mag.*, vol. 33, Aug. 1995.
- [22] "Special Issue on Digital Interactive Broadband Video Dial Tone Networks," *IEEE Network Mag.*, vol. 9, Sept./Oct. 1995.
- [23] W. J. Liao and V. O. K. Li, "The Split-and-Merge (SAM) protocol for interactive video-on-demand systems," in *Proc. IEEE INFOCOM'97*, Kobe, Japan, 1997, pp. 1351-1358.
- [24] L. Crutcher and J. Grinham, "The networked video jukebox," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 105-120, Apr. 1994.
- [25] C. L. Lin and S. U. Guan, "The design and architecture of a video library system," *IEEE Commun. Mag.*, vol. 34, pp. 86-91, Jan. 1996.
- [26] S. Ramanathan and R. Gueslla, "Toward management systems for emerging hybrid fiber-coax access networks," *IEEE Network Mag.*, pp. 58-68, Sept./Oct. 1995.
- [27] T. S. Perry, "The trials and travails of interactive TV," *IEEE Spectrum*, vol. 33, pp. 22-28, Apr. 1996.
- [28] S. Dixit and P. Skelly, "MPEG-2 over ATM for video dial tone networks: Issues and strategies," *IEEE Network Mag.*, vol. 9, pp. 30-40, Sept./Oct. 1995.
- [29] D. Large, "Creating a network for interactivity," *IEEE Spectrum*, vol. 32, pp. 58-63, Apr. 1995.
- [30] W. Y. Chen and D. L. Waring, "Applicability of ADSL to support video dial tone in the copper loop," *IEEE Commun. Mag.*, vol. 32, pp. 102-109, May 1994.
- [31] P. J. Kyees, R. C. McConnell, and K. Sistanizadeh, "ADSL: A new twisted-pair access to the information highway," *IEEE Commun. Mag.*, vol. 33, pp. 52-59, Apr. 1995.
- [32] L. Campbell, "Digital Oracle in VOD tie-up," *South China Morning Post*, May 23, 1995.
- [33] "Special Issue on Fiber-Optic Subscriber Loops," *IEEE Commun. Mag.*, vol. 32, Feb. 1994.
- [34] V. Leon and E. Miller, "Subscriber terminal units for video dial tone systems," *IEEE Network Mag.*, pp. 48-57, Sept./Oct. 1995.
- [35] J. Morales, A. Patka, P. Choa, and J. Kui, "Video dial tone sessions," *IEEE Network Mag.*, vol. 9, Sept./Oct. 1995.
- [36] T. D. C. Little and D. Venkatesh, "Popularity-based assignment of movies to storage devices in a video-on-demand system," *ACM/Springer-Verlag Multimedia Syst.*, vol. 2, pp. 137-150, 1995.
- [37] G. H. Petit, D. Deloddere, and W. Verbiest, "Bandwidth resource optimization in video-on-demand network architectures,"

in *Proc. Int. Workshop Community Networking Integrated Multimedia Service to the Home*, New York, 1994, pp. 91–97.

[38] F. Schaffa and J. P. Nussbaumer, “On bandwidth and storage tradeoffs in multimedia distribution networks,” in *Proc. Int. Conf. Multimedia Computing Systems*, Washington, D.C., 1995, pp. 1020–1026.

[39] Y. Tanaka and O. Berlage, “The storage location problem in video-on-demand service,” pp. 1–10, 1994.

[40] C. A. Ellis, S. J. Gibbs, and G. L. Rein, “Groupware: Some issues and experiences,” *Commun. ACM*, vol. 34, pp. 38–58, Jan. 1991.

[41] J. Grudin, “Computer-supported cooperative work: History and focus,” *IEEE Comput. Mag.*, pp. 19–26, May 1994.

[42] H. Ishii and N. Miyake, “Toward an open shared workspace: Computer and video fusion approach of team workstation,” *Commun. ACM*, vol. 34, pp. 37–50, Dec. 1991.

[43] R. Kling, “Cooperation, coordination, and control in computer-supported work,” *Commun. ACM*, vol. 34, pp. 83–88, Dec. 1991.

[44] J. D. Palmer and N. A. Fields, “Computer-Supported Cooperative Work,” *IEEE Comput. Mag.*, pp. 15–17, May 1994.

[45] W. Reinhard, J. Schweitzer, G. Völksen, and M. Weber, “CSCW tools: Concepts and architectures,” *IEEE Comput. Mag.*, pp. 28–36, May 1994.

[46] M. R. Macedonia and D. P. Brutzman, “MBone provides audio and video across the Internet,” *IEEE Comput. Mag.*, pp. 30–36, Apr. 1994.

[47] R. Bentley, T. Rodden, P. Sawyer, and I. Sommerville, “Architectural support for cooperative multiuser interfaces,” *IEEE Comput. Mag.*, pp. 37–46, May 1994.

[48] RFC. (1996). MIME: Multipurpose Internet Mail Extensions. Available: <http://www.oac.uci.edu/indiv/ehood/MIME/MIME.html>.

[49] R. Fish, R. Kraut, M. Leland, and M. Cohen, “Quilt: A collaborative tool for cooperative writing,” in *Proc. COIS’88*, Palo Alto, CA, 1988, pp. 23–25.

[50] I. Greif, R. Seliger, and W. Weihl, “Atomic data abstractions in a distributed collaborative editing system,” in *Proc. ACM CSCW’86*, Seattle, WA, 1986, pp. 160–172.

[51] G. E. Kaiser, S. M. Kaplan, and J. Micallef, “Multiuser, distributed language-based environments,” *IEEE Software Mag.*, pp. 58–67, Nov. 1987.

[52] Intel. (1997). Video Phone. Available: <http://intel.com/proshare/videophone/about/index.html>.

[53] Cornell. (1997). CU-SeeMe. Available: http://cu-seeme.cornell.edu/pc.CU_SeeMeCurrent.html.

[54] K. L. Kraemer and J. L. King, “Computer-based systems for cooperative work and group decision making,” *ACM Computing Surveys*, vol. 20, pp. 115–146, June 1988.

[55] L. M. Applegate, B. R. Konsynski, and J. F. Nunamaker, “A group decision support system for idea generation and issue analysis in organization planning,” in *Proc. ACM CSCW’86*, Seattle, WA, 1986, pp. 16–34.

[56] J. C. Lauwers and K. A. Lantz, “Collaborative awareness in support of collaboration transparency: Requirements for the next generation of shared window systems,” in *Proc. CHI’90*, Seattle, WA, 1990, pp. 303–311.

[57] K. Watabe, S. Sakata, K. Maeno, H. Fukuka, and T. Ohmori, “Distributed desktop conferencing system with multiuser multimedia interface,” *IEEE J. Select. Areas Commun.*, vol. 9, pp. 531–539, May 1991.

[58] T. P. Liang, H. Lai, N. S. Chen, H. Wei, and M. C. Chen, “When client/server isn’t enough: Coordinating multiple distributed tasks,” *IEEE Comput. Mag.*, pp. 73–79, May 1994.

[59] M. Stefk, D. G. Bobrow, G. Foster, S. Lanning, and D. Tatar, “WYSIWIS revised: Early experiences with multiuser interfaces,” *ACM Trans. Office Inform. Syst.*, vol. 5, pp. 147–167, Apr. 1987.

[60] S. Sarin and I. Greif, “Computer-based real-time conferencing systems,” *IEEE Comput. Mag.*, vol. 18, pp. 33–45, Oct. 1985.

[61] E. J. Conklin, “Hypertext: An introduction and survey,” *IEEE Comput. Mag.*, vol. 2, pp. 17–41, Sept. 1987.

[62] J. Nielsen, *Multimedia and Hypertext: The Internet and Beyond*. AP Professional, 1995.

[63] “Special Issue on Hypertext,” *Commun. ACM*, vol. 32, July 1988.

[64] “Special Issue on Hypermedia,” *Commun. ACM*, vol. 37, Feb. 1994.

[65] J. B. Smith and S. F. Weiss, “Hypertext,” *Commun. ACM*, vol. 31, pp. 816–819, July 1988.

[66] J. Nielsen, “The art of navigating,” *Commun. ACM*, vol. 33, pp. 296–310, Mar. 1990.

[67] R. M. Akscyn, D. L. McCracken, and E. A. Yoder, “KMS: A distributed hypermedia system for managing knowledge in organizations,” *Commun. ACM*, vol. 31, pp. 820–835, July 1988.

[68] J. Nielsen and U. Lyngbak, “Two field studies of hypermedia usability,” in *Proc. Hypertext II Conf.* Pittsburgh, PA, 1989, pp. 29–30.

[69] M. Bernstein, “The bookmark and compass: Orientation tools for hypertext users,” *ACM SIGOIS Bulletin*, vol. 9, pp. 34–45, Oct. 1988.

[70] E. Rivlin, R. Botafogo, and B. Schneiderman, “Navigating in hyperspace: Designing a structure-based toolbox,” *Commun. ACM*, vol. 37, pp. 87–96, Feb. 1994.

[71] B. Campbell and J. M. Goodman, “HAM: A general purpose hypertext abstract machine,” *Commun. ACM*, vol. 32, pp. 856–861, July 1988.

[72] F. Garzotto, P. Paolini, and D. Schwabe, “HDM—A model for the design of hypertext applications,” in *Proc. ACM Hypertext’91* San Antonio, TX, 1991, pp. 313–328.

[73] F. Halasz and M. Schwartz, “The Dexter hypertext reference model,” in *NIST Hypertext Standardization Workshop*, Gaithersburg, MD, pp. 95–133, 1990.

[74] ———, “The Dexter hypertext reference model,” *Commun. ACM*, vol. 37, pp. 30–39, Feb. 1994.

[75] S. R. Newcomb, N. A. Kipp, and V. T. Newcomb, “The HyTime: Hypermedia/time-based document structuring language,” *Commun. ACM*, vol. 34, pp. 67–83, Nov. 1991.

[76] T. Meyer-Boudnik and W. Effelsberg, “MHEG explained,” *IEEE Multimedia Mag.*, pp. 26–38, Spring 1995.

[77] CCITT, “CCITT Press Release on MHEG,” Dec. 1992.

[78] *Information technology—Coded representation of multimedia and hypermedia information (MHEG), Part 1: Base notation (ASN.1)*, Int. Standard ISO/IEC CD 13552-1:1993, Oct. 1994.

[79] K. Nahrstedt and J. M. Smith, “The QOS broker,” *IEEE Multimedia Mag.*, pp. 53–67, Spring 1995.

[80] A. Campbell, G. Coulson, and D. Hutchinson, “A quality of service architecture,” *Comput. Commun. Rev.*, vol. 24, pp. 6–27, Apr. 1994.

[81] A. Vogel, B. Kerhervé, G. V. Bockmann, and J. Gecsei, “Distributed multimedia and QOS: A survey,” *IEEE Multimedia Mag.*, pp. 10–19, Summer 1995.

[82] K. Nahrstedt and R. Steinmetz, “Resource management in networked multimedia systems,” *IEEE Comput. Mag.*, vol. 28, pp. 52–63, May 1995.

[83] W. J. Liao and V. O. K. Li, “Packet loss model in video-on-demand systems,” *IEEE/ACM Trans. Networking*, submitted for publication.

[84] C. Nicolaou, “An architecture for real-time multimedia communication systems,” *IEEE J. Select. Areas Commun.*, vol. 8, pp. 391–400, Apr. 1990.

[85] C. Topolcic, “Experimental Internet stream protocol: Version 2 (ST-II),” *Internet RFC 1190*, Oct. 1990.

[86] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, “RSVP: A new resource ReSerVation Protocol,” *IEEE Network Mag.*, pp. 8–18, Sept. 1993.

[87] A. Banerjea and B. Mah, “The real-time channel administration protocol,” in *Proc. 2nd Int. Workshop Network and Operating System for Digital Audio and Video*, Heidelberg, Germany, 1991, pp. 160–170.

[88] J. Turner, “New directions in communications (or which way to the information age?),” *IEEE Commun. Mag.*, vol. 24, pp. 8–15, Oct. 1986.

[89] I. Khan and V. O. K. Li, “Traffic control in ATM networks,” *Comput. Networks ISDN Syst.*, vol. 27, pp. 85–100, Oct. 1994.

[90] N. Jayant, “Signal compression: Technology targets and research directions,” *IEEE J. Select. Areas Commun.*, vol. 10, no. 5, pp. 796–818, 1992.

[91] R. Steinmetz, “Data compression in multimedia computing—principles and techniques,” *ACM/Springer-Verlag Multimedia Syst.*, vol. 1, no. 4, pp. 166–172, 1994.

[92] G. K. Wallace, “The JPEG still picture compression standard,” *Commun. ACM*, vol. 34, no. 4, pp. 30–34, 1991.

[93] *Information Technology—Digital Compression and Coding of Continuous-Tone Still Images*, Int. Standard ISO/IEC IS 10918, 1993.

[94] M. Liou, "Overview of the $p \times 64$ kbit/s video coding standard," *Commun. ACM*, vol. 34, no. 4, pp. 59–63, 1991.

[95] D. L. Gall, "MPEG: A video compression standard for multimedia applications," *Commun. ACM*, vol. 34, no. 4, pp. 46–58, 1991.

[96] *Information Technology—Coding of Moving Pictures and Associated Audio for Digital Storage Media up to About 1.5 Mbit/s*, International Standard ISO/IEC IS 11 172, 1993.

[97] T. Chiang and D. Anastassiou, "Hierarchical coding of digital television," *IEEE Commun. Mag.*, pp. 38–45, May 1994.

[98] D. Y. Pan, "A tutorial MPEG/audio compression," *IEEE Multimedia Mag.*, pp. 60–74, Spring 1995.

[99] Dolby Labs, Inc. (1996). Multichannel perceptual coding. Available: <http://www.dolby.com/tech/multipc.html>.

[100] S. Armstrong, A. Freier, and K. Marzullo, "Multicast Transport Protocol," *Internet RFC 1301*, 1992.

[101] S. Casner, "Are you on the MBone?" *IEEE Multimedia Mag.*, pp. 76–79, Summer 1994.

[102] ———, (1994) FAQ on the Multicast Backbone (MBONE). Available: <http://www.research.att.com/mbone-faq.html>.

[103] D. Zappala. (1996). RSVP Protocol Overview. Available: <http://www.isi.edu/div7/rsvp/overview.html>,

[104] S. Berson. (1996). Classical RSVP and IP over ATM. Available: <http://www.isi.edu/div7/rsvp/overview.html>.

[105] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," *Internet RFC 1889*, Jan. 1996.

[106] X. Forum. (1996). XTP: The Xpress Transport Protocol. Available: <http://www.ca.sandia.gov/xtp/xtp.html>.

[107] J. D. Salehi, Z. L. Zhang, J. F. Kurose, and D. F. Towsley, "Supporting stored video: Reducing rate variability and end-to-end resource requirements through optimal smoothing," in *ACM SIGMETRICS'96*, Philadelphia, PA, pp. 222–231.

[108] N. Kamiyama and V. O. Li, "Renegotiated CBR transmission in interactive video-on-demand systems," in *Proc. IEEE Conf. Multimedia Computing Systems (ICMCS)*, Ottawa, Canada, 1997, pp. 12–19.

[109] C. J. Turner and L. L. Peterson, "Image transfer: An end-to-end design," in *ACM SIGCOMM'92*, Baltimore, MD, pp. 258–268, 1992.

[110] G. Coulson, G. S. Blair, and P. Robin, "Micro-kernel support for continuous media in distributed systems," *Comput. Networks ISDN Syst.*, vol. 26, pp. 1323–1341, 1994.

[111] H. Tokuda, T. Nakajima, and P. Rao, "Real-time mach: Toward a predictable real-time system," in *Proc. Usenix Mach Workshop*, 1990.

[112] R. Steinmetz, "Analyzing the multimedia operating system," *IEEE Multimedia Mag.*, pp. 68–84, Spring 1995.

[113] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard real-time environment," *J. ACM*, vol. 20, pp. 46–61, Jan. 1973.

[114] H. Tokuda, "Operating system support for continuous media applications," in *Multimedia Systems*. Reading, MA: Addison-Wesley, 1994, pp. 201–220.

[115] R. Rooholamini and V. Cherkassky, "ATM-based multimedia servers," *IEEE Multimedia Mag.*, pp. 39–52, Spring 1995.

[116] A. L. N. Reddy and J. C. Wyllie, "I/O issues in a multimedia system," *IEEE Comput. Mag.*, vol. 27, pp. 69–74, Mar. 1994.

[117] M. J. Perez-Luque and T. D. C. Little, "A temporal reference framework for multimedia synchronization," *IEEE J. Select. Areas Commun.*, vol. 14, pp. 36–51, Jan. 1996.

[118] T. D. C. Little, "A framework for synchronization delivery of time-dependent multimedia data," *ACM/Springer-Verlag Multimedia Syst.*, vol. 1, no. 2, pp. 87–94, 1993.

[119] R. Steinmetz, "Synchronization properties in multimedia systems," *IEEE J. Select. Areas Commun.*, vol. 8, pp. 401–412, Apr. 1990.

[120] "Special issue on synchronization issues in multimedia communications," *IEEE J. Select. Areas Commun.*, vol. 14, Jan. 1996.

[121] D. L. Mills, "Internet time synchronization: The network time protocol," *IEEE Trans. Commun.*, vol. 39, pp. 1482–1493, Oct. 1991.

[122] S. Ramanathan and P. Rangan, "Adaptive feedback techniques for synchronized multimedia retrieval over integrated networks," *IEEE/ACM Trans. Networking*, vol. 1, pp. 246–260, Apr. 1993.

[123] T. D. C. Little and A. Ghafoor, "Network considerations for distributed multimedia object composition and communication," *IEEE Network Mag.*, pp. 32–49, Nov. 1990.

[124] H. Santoso, L. Dairaine, S. Fdida, and E. Horlait, "Preserving temporal signature: A way to convey time constrained flows," in *IEEE GLOBECOM'93*, Houston, TX, pp. 872–876.

[125] R. G. Herrtwich, "Summary of the second international workshop on network and operating system support for digital audio and video," *ACM Comput. Commun. Rev.*, pp. 6–8, Oct. 1992.

[126] D. Shepherd and M. Salmony, "Extending OSI to support synchronization required by multimedia applications," *Comput. Commun.*, vol. 13, pp. 399–406, Sept. 1990.

[127] D. Ferrari, "A scheme for real-time channel establishment in wide-area networks," *IEEE J. Select. Areas Commun.*, vol. 8, pp. 368–379, Apr. 1990.

[128] J. P. Courtiat, L. F. R. de Costa Carmo, and R. C. de Oliveira, "A general-purpose multimedia synchronization mechanism based on causal relations," *IEEE J. Select. Areas Commun.*, vol. 14, pp. 185–195, Jan. 1996.

[129] Y. Ishibashi and S. Tasaka, "A synchronization mechanism for continuous media in multimedia communications," in *IEEE INFOCOM*, Boston, MA, pp. 1010–1019, Apr. 1995.

[130] W. J. Liao and V. O. K. Li, "Synchronization of distributed multimedia systems with user interactions," *ACM/Springer-Verlag Multimedia Syst.*, to be published.

[131] T. D. C. Little and A. Ghafoor, "Synchronization and storage models for multimedia objects," *IEEE J. Select. Areas Commun.*, vol. 8, pp. 413–427, Apr. 1990.

[132] ———, "Multimedia synchronization protocols for broadband integrated services," *IEEE J. Select. Areas Commun.*, vol. 9, pp. 1368–1382, Dec. 1991.

[133] G. J. Lu, H. K. Pung, T. S. Chua, and S. F. Chan, "Temporal synchronization support for distributed multimedia information systems," *Comput. Commun.*, vol. 17, pp. 852–862, Dec. 1994.

[134] K. Ravindran and V. Bansal, "Delay compensation protocols for synchronization of multimedia data streams," *IEEE Trans. Knowledge Data Eng.*, vol. 5, pp. 574–589, Aug. 1993.

[135] M. Woo, N. Qazi, and A. Ghafoor, "A synchronization framework for communication of pre-orchestrated multimedia information," *IEEE Network Mag.*, vol. 8, pp. 52–61, Jan./Feb. 1994.

[136] I. F. Akyildiz and W. Yen, "Multimedia group protocols for integrated services networks," *IEEE J. Select. Areas Commun.*, vol. 14, pp. 162–173, Jan. 1996.

[137] L. Li, A. Karmouch, and N. Georganas, "Multimedia teleorchestra with independent source: Part I—Temporal modeling of collaborative multimedia scenarios," *ACM/Springer-Verlag Multimedia Syst.*, vol. 1, no. 4, pp. 143–153, 1994.

[138] T. D. C. Little and A. Ghafoor, "Spatio-temporal composition of distributed multimedia objects for value-added networks," *IEEE Comput. Mag.*, pp. 42–50, Oct. 1991.

[139] M. Holliday and M. Vernon, "A generalized timed petri net model for performance analysis," in *Proc. Int. Conf. Timed Petri Nets*, Torino, Italy, 1985, pp. 181–190.

[140] J. L. Peterson, "Petri nets," *ACM Computing Surveys*, vol. 9, pp. 223–252, Sept. 1977.

[141] R. R. Razouk and C. Phelps, "Performance analysis using timed petri nets," in *Proc. Int. Conf. Parallel Processing*, 1984, pp. 126–129.

[142] P. Hoepner, "Synchronizing the presentation of multimedia objects," *Comput. Commun.*, vol. 15, pp. 557–564, Nov. 1992.

[143] W. J. Liao and V. O. K. Li, "Multimedia data modeling in distributed multimedia systems," in *Proc. 1996 Pacific Workshop Distributed Multimedia Systems*, Hong Kong, 1996, pp. 10–17.

[144] B. Prabhakaran and S. V. Raghavan, "Synchronization models for multimedia presentation with user participation," *Multimedia Syst.*, vol. 2, no. 2, pp. 53–62, 1994.

[145] M. C. Buchanan and P. T. Zellweger, "Automatically generating schedules for multimedia documents," *Multimedia Syst.*, vol. 1, no. 2, pp. 55–67, 1993.

[146] G. S. Blair, G. Coulson, M. Papathomas, P. Robin, J. B. Stefani, F. Horn, and L. Hazard, "A programming model and system infrastructure for real-time synchronization in distributed multimedia systems," *J. Select. Areas Commun.*, vol. 14, no. 1, pp. 249–263, 1996.

[147] R. B. Dannenberg, T. Neuendorffer, J. Newcomer, D. Rubine, and D. Anderson, "Tactus: Toolkit-level support for synchronized interactive multimedia," *Multimedia Syst.*, vol. 2, no. 1, pp. 77–86, 1993.

[148] R. Hamakawa and J. Rekimoto, "Object composition and playback models for handling multimedia data," *Multimedia Syst.*, vol. 2, no. 1, pp. 26–35, 1994.

[149] F. Horn and J. B. Stefani, "On programming and supporting multimedia object synchronization," *Comput. J.*, vol. 36, no. 1, pp. 4–18, 1993.

[150] M. Vazirgiannis and C. Mourlas, "An object-oriented model for interactive multimedia presentations," *Comput. J.*, vol. 36, no. 1, pp. 78–86, 1993.

[151] C. M. Huang and C. M. Lo, "An EFSM-based multimedia synchronization model and the authoring system," *IEEE J. Select. Areas Commun.*, vol. 14, pp. 138–152, Jan. 1996.

[152] C. L. Jamblin, "Instants and intervals," in *Proc. 1st Conf. Int. Society Study of Time*, 1972, pp. 324–331.

[153] J. F. Allen, "Maintaining knowledge about temporal intervals," *Commun. ACM*, vol. 26, pp. 832–843, Nov. 1983.

[154] T. D. C. Little and A. Ghafoor, "Interval-based conceptual models for time-dependent multimedia data," *IEEE Trans. Knowledge Data Eng.*, vol. 5, pp. 551–563, Aug. 1993.

[155] T. Wahl and K. Rothermel, "Representing time in multimedia systems," in *Proc. Int. Conf. Multimedia Computing Systems*, Boston, MA, 1994, pp. 538–543.

[156] V. O. K. Li and W. J. Liao, "A synchronization model for distributed multimedia systems," Communication Sciences Institute, University of Southern California, Los Angeles, Rep. CSI-94-12-02, submitted for publication.

[157] J. Escobar, C. Partridge, and D. Deutsch, "Flow synchronization protocol," *IEEE/ACM Trans. Networking*, vol. 2, pp. 111–121, 1994.

[158] P. N. Zarros, M. J. Lee, and T. N. Saadawi, "A synchronization algorithm for distributed multimedia environments," *ACM/Springer-Verlag Multimedia Syst.*, vol. 3, pp. 1–11, Feb. 1996.

[159] D. J. Gemmell, H. M. Vin, D. D. Kandlur, P. V. Rangan, and L. A. Rowe, "Multimedia storage servers: A tutorial," *IEEE Comput. Mag.*, vol. 28, pp. 40–49, May 1995.

[160] H. M. Vin, A. Goyal, and P. Goyal, "Algorithms for designing multimedia servers," *Comput. Commun.*, vol. 18, pp. 192–203, Mar. 1995.

[161] H. M. Vin and P. V. Rangan, "Designing a multiuser HDTV storage server," *IEEE J. Select. Areas Commun.*, vol. 11, pp. 153–164, Jan. 1993.

[162] A. Dan and D. Sitaram, "An online video placement policy based on bandwidth to space ratio (BSR)," in *Proc. ACM SIGMOD'95*, San Jose, CA, 1995, pp. 376–385.

[163] S. Berson, L. Golubchik, and R. R. Muntz, "Fault tolerant design of multimedia servers," in *Proc. ACM SIGMOD'95*, San Jose, CA, 1995, pp. 364–375.

[164] D. Bitton and J. Gray, "Disk shadowing," in *Proc. VLDB'86*, Kyoto, Japan, 1986, pp. 331–338.

[165] D. A. Patterson, G. Gibson, and R. H. Katz, "A case for redundant arrays of inexpensive disks (RAID)," in *Proc. ACM SIGMOD'88*, Chicago, IL, 1988, pp. 109–116.

[166] D. J. Gemmell and S. Christodoulakis, "Principles of delay sensitive multimedia data storage and retrieval," *ACM Trans. Inform. Syst.*, vol. 10, pp. 51–90, Jan. 1992.

[167] R. Steinmetz, "Multimedia file systems survey: Approaches for continuous media disk scheduling," *Comput. Commun.*, vol. 18, pp. 133–144, Mar. 1995.

[168] S. Sengodan and V. O. K. Li, "A generalized grouping and retrieval schemes for stored MPEG video," in *Proc. IEEE ICC*, Montreal, Canada, 1997, pp. 1674–1678.

[169] —, "Grouping and retrieval schemes for stored MPEG video," *IEEE/ACM Trans. Networking*, submitted for publication.

[170] D. Anderson, Y. Osawa, and R. Govindan, "A file system for continuous media," *ACM Trans. Comput. Syst.*, vol. 10, pp. 311–337, Nov. 1992.

[171] P. V. Rangan and H. M. Vin, "Efficient storage techniques for digital continuous multimedia," *IEEE Trans. Knowledge Data Eng.*, vol. 5, pp. 564–573, Aug. 1993.

[172] P. V. Rangan, H. M. Vin, and S. Ramanathan, "Designing an on-demand multimedia service," *IEEE Commun. Mag.*, vol. 30, pp. 56–65, July 1992.

[173] A. L. N. Reddy and J. Wyllie, "Disk scheduling in a multimedia I/O system," in *Proc. ACM Multimedia'93*, Anaheim, CA, pp. 255–233.

[174] M. T. Chen, D. D. Kandlur, and P. S. Yu, "Optimization of the grouped sweeping scheduling (GSS) with heterogeneous multimedia streams," in *Proc. ACM Multimedia'93*, Anaheim, CA, 1993, pp. 57–64.

[175] S. Sengodan and V. O. K. Li, "A quasistatic retrieval scheme for interactive disk-based VOD servers," *Comput. Commun.*, to be published.

[176] —, "QuIVeR: A class of interactive video retrieval protocols," in *Proc. IEEE Conf. Multimedia Computing Systems (ICMCS)*, Ottawa, Canada, 1997, pp. 186–193.

[177] —, "DIVEr: A dynamic interactive retrieval protocol for VOD servers," *ACM/Springer-Verlag Multimedia Syst.*, submitted for publication.

[178] —, "A shared buffer architecture for video servers," in *Proc. IEEE INFOCOM'97*, Kobe, Japan, 1997, pp. 1343–1350.

[179] P. Chen, E. Lee, G. Gibson, R. Katz, and D. Patterson, "RAID: High-performance, reliable secondary storage," *ACM Comput. Surveys*, vol. 26, pp. 145–186, June 1994.

[180] D. A. Paterson, G. Gibson, and R. A. Katz, "Introduction to redundant arrays of inexpensive disks (RAID)," in *Proc. IEEE Compcon'89*, San Francisco, CA, pp. 352–363.

[181] M. Y. Kim, "Synchronized data interleaving," *IEEE Trans. Comput.*, vol. 35, no. 11, pp. 978–988, 1986.

[182] K. Salem and H. Garcia-Molina, "Disk striping," in *Proc. IEEE Data Engineering Conf.*, Los Angeles, CA, 1986, pp. 336–342.

[183] F. A. Tobagi, J. Pang, R. Baird, and M. Gang, "Streaming RAID—A disk array management system for video files," in *Proc. ACM Multimedia'93*, Anaheim, CA, pp. 393–400.

[184] S. Ghandeharizade and L. Ramos, "A continuous retrieval of multimedia data using parallelism," *IEEE Trans. Knowledge Data Eng.*, vol. 4, no. 5, pp. 658–681, 1993.

[185] W. Klas and A. Sheth, "Metadata for digital media: Introduction to the special issue," *ACM SIGMOD Rec.*, vol. 23, no. 4, pp. 19–20, 1994.

[186] M. T. Özsu, D. Szafron, G. Elmedani, and C. Vittal, "An object-oriented multimedia database system for a news-on-demand application," *ACM/Springer-Verlag Multimedia Syst.*, vol. 3, no. 5/6, pp. 217–227, 1995.

[187] M. Colton. (1995). "Multimedia asset management white paper." Available: <http://www.illustra.com>.

[188] W. Klas, E. J. Neuhold, and M. Schrefl, "Using an object-oriented approach to model multimedia data," *Comput. Commun.*, vol. 13, no. 4, pp. 204–216, 1990.

[189] W. Kim, *Modern Database Systems: The Object Model, Interoperability, and Beyond*. Reading, MA: Addison-Wesley, 1995.

[190] R. Weiss, A. Duda, and D. K. Gifford, "Composition and search with a video algebra," *IEEE Multimedia Mag.*, pp. 12–25, Spring 1995.

[191] S. W. Smoliar and H. J. Zhang, "Content-based video indexing and retrieval," *IEEE Multimedia Mag.*, vol. 1, pp. 62–72, Spring 1994.

[192] S. Christodoulakis, M. Theodoridou, F. Ho, and M. Papa, "Multimedia document presentation, information extraction and document formation in MINOS: A model and a system," *ACM Trans. Office Inform. Syst.*, vol. 4, no. 4, pp. 345–383, 1986.

[193] W. W. Chu, I. T. Jeong, R. K. Taira, and C. M. Breant, "Temporal evolutionary object-oriented data model and its query language for medical image management," in *Proc. Very Large Database System'92*, Vancouver, British Columbia, Canada, pp. 53–64.

[194] H. Ishikawa, F. Suzuki, F. Kozakura, A. Makinouchi, M. Miyagishima, I. Y. M. Aoshima, and Y. Yamane, "The model, language, and implementation of an object-oriented multimedia knowledge base management system," *ACM Trans. Database Syst.*, vol. 18, no. 1, pp. 1–50, 1993.

[195] T. Joseph and A. Cardenas, "PICQUERY: A high level query language for pictorial database management," *IEEE Trans. Software Eng.*, vol. 14, no. 5, pp. 630–638, 1988.

[196] W. I. Grosky, "Multimedia information systems," *IEEE Multimedia Mag.*, pp. 12–24, Spring 1994.

[197] Y. Tonomura, A. Akutsu, Y. Taniguchi, and G. Suzuki, "Structured video computing," *IEEE Multimedia Mag.*, pp. 34–43, Fall 1994.

[198] J. T. Anderson and M. Stonebraker, "SEQUOIA 2000 metadata schema for satellite images," *ACM SIGMOD Rec.*, vol. 23, no. 4, pp. 42–48, 1994.

[199] K. Böhme and T. C. Rakow, "Metadata for multidata docu-

ments," *ACM SIGMOD Rec.*, vol. 23, no. 4, pp. 21–26, 1994.

[200] U. Glavitsch, P. Schäuble, and M. Wechsler, "Metadata for integrating speech documents in a text retrieval system," *ACM SIGMOD Rec.*, vol. 23, no. 4, pp. 57–63, 1994.

[201] W. I. Grosky, F. Fotouhi, and I. K. Sethi, "Using metadata for the intelligent browsing of structured multimedia objects," *ACM SIGMOD Rec.*, vol. 23, no. 4, pp. 49–56, 1994.

[202] R. Jain and A. Hampapur, "Metadata in video databases," *ACM SIGMOD Rec.*, vol. 23, no. 4, pp. 27–33, 1994.

[203] Y. Kiyoki, T. Kitagawa, and T. Hayama, "A metadatabase system for semantic image search by a mathematical model of meaning," *ACM SIGMOD Rec.*, vol. 23, no. 4, pp. 34–41, 1994.

[204] M. T. Ozsu and P. Valduriez, *Principles of Distributed Database Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1991.

[205] C. Y. Chen, D. S. Meliksetian, M. C. S. Chang, and L. J. Liu, "Design of a multimedia object-oriented DBMS," *ACM/Springer-Verlag Multimedia Syst.*, vol. 3, no. 5/6, pp. 217–272, 1995.

[206] S. Gibbs, C. Breiteneder, and D. Tsichritzis, "Data modeling of time-based media," in *Proc. ACM SIGMOD'94*, Minneapolis, MN, 1994, pp. 91–102.



Victor O. K. Li (Fellow, IEEE) was born in Hong Kong in 1954. He received the S.B., S.M., and Sc.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, in 1977, 1979, and 1981, respectively.

In February 1981, he joined the University of Southern California (USC), Los Angeles. Since September 1997, he has been Chair Professor of Information Engineering, Department of Electrical and Electronic Engineering, University of Hong Kong. He is on leave from USC, where he is a Professor of electrical engineering and Director of the USC Communication Science Institute. His research interests include high-speed communication networks, personal communication networks, distributed multimedia systems, distributed databases, queueing theory, graph theory, and applied probability. He has lectured and consulted extensively around the world.

Dr. Li was Chairman of the Computer Communications Technical Committee of the IEEE Communications Society in 1987–1989 and the Los Angeles Chapter of the IEEE Information Theory Group in 1983–1985. He was Chairman of the Steering Committee of the International Conference on Computer Communications and Networks (IC^3N) in 1992–1997. He was General Chair of the 1st Annual IC^3N , June 1992, Technical Program Chair of the Institute of Electrical Engineers Personal Communication Services Symposium, June 1995, and Chair of the 4th IEEE Workshop on Computer Communications, October 1989. He is a member of the Association for Computing Machinery, Inc. (ACM). He was an Editor of IEEE NETWORK: THE MAGAZINE OF COMPUTER COMMUNICATIONS and of *Telecommunication Systems*, Guest Editor of IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS and of *Computer Networks and ISDN Systems*, and is now serving as an Editor of ACM *Wireless Networks*. He has served on the International Advisory Board of IEEE TENCON'90, IEEE TENCON'94, IEEE SICON'91, IEEE SICON'93, IEEE SICON/ICIE'95, the International Conference on Microwaves and Communications '92, and the International Symposium on Communications '91.



Wanjiun Liao (Student Member, IEEE) was born in Taiwan in 1968. She received the B.S. and M.S. degrees from National Chiao Tung University, Taiwan, in 1990 and 1992, respectively, and the Ph.D. degree in electrical engineering from the University of Southern California (USC), Los Angeles

Since August 1997, she has been an Assistant Professor, Department of Electrical Engineering, National Taiwan University. Her research interests include distributed multimedia systems, multimedia communications, networking, and distributed database systems.

Dr. Liao is a member of Phi Tau Phi. She received the Acer Long-Term Thesis Award, the Chinese Institute of Electrical Engineers Graduate Student Thesis Award, and the USC Outstanding Research Paper Award.