CHAPTER 6

Multimedia Communications Across Networks

Chapter Overview

This chapter concentrates on multimedia communications across networks. After an introductory discussion concerning packet audio-video in the network environment, we invoke the concept of video transport across generic networks. We then describe Multimedia transport across ATM networks. This is followed by multimedia across IP networks, including video transmission, traffic specifications for MPEG video transmission on the Internet and bandwidth allocation mechanism. We outline the issues concerning multimedia Digital Subscriber Lines (DSLs). The concepts of Internet access networks are presented and illustrated. We finally discuss special issues relating to multimedia across wireless networks, such as wireless broadband communication for multimedia audiovisual solutions, mobile and broadcasting networks as well as digital TV infrastructure for interactive multimedia services.

6.1 Packet Audio/Video in the Network Environment

Packet-switched networks were invented for carrying computer data because the burst-type nature of such information makes it uneconomical to use continuously connected circuits. Audio and video signals, in contrast, have for many years been carried across fixed-bit-rate circuit-

switched connections. However, developments in ATM networks have generated discussions between network and coding specialists concerning the potential advantages of variable bit-rate transmissions across such networks. In recent years, considerable interest has been shown in the general statistical multiplexing of digitally encoded audio and video signals, and particular attention has been given to packet-based systems. A large number of papers and books has appeared in the literature, addressing topics such as delays involved, the associated queuing problems, the effects of packet loss and the regeneration of lost packets.

The increase in communication of multimedia information over the past decades has resulted in many new multimedia processing and communication systems being put into service. The growing availability of optical fiber links and rapid progress in Very Large-Scale Integration (VLSI) circuits and systems have fostered a tremendous interest in developing sophisticated multimedia services with an acceptable cost. Today's fiber technology offers a transmission capacity that can easily handle high bit rates. This leads to the development of networks that integrate all types of information services [6.1]. By basing such a network on packet switching, the services (video, voice and data) can be dealt with in a common format. Packet switching is more flexible than circuit switching in that it can emulate the latter while vastly different bit rates can be multiplexed together. In addition, the network's statistical multiplexing of variable rate sources may yield a higher fixed-capacity allocation [6.2, 6.3, 6.4].

6.1.1 Packet Voice

In comparison to circuit-switched networks, packet switching offers several potential advantages in terms of performance. One advantage is efficient use of channel capacity, particularly for bursty traffic. Although not as bursty as interactive data, speech exhibits some burstiness in the form of talksparts [6.5]. Average talkspart duration depends on the sensitivity of the speech detector, but it is well known that individual speakers are active only about 35 to 45% in typical telephone conversations. By sending voice packets only during talksparts, packet switching offers a natural way to multiplex voice calls as well as voice with data. Another advantage is that call blocking can be a function of the required average bandwidth rather than the required peak bandwidth. In addition, packet switching is flexible. For example, packet voice is capable of supporting point-to-multipoint connections and priority traffic. Furthermore, because packets are processed in the network, network capabilities in traffic control, accounting and security are enhanced. However, packet voice is not without difficulties. Continuous speech of acceptable quality must be reconstructed from a voice packet that experiences variable delays through the network. The reconstruction process involves compensating for the variable delay component by imposing an additional delay. Hence, the packet should be delivered with low-average delay and delay variability.

Speech can tolerate a certain amount of distortion (for example, compression and clipping) but is sensitive to end-to-end delay. The exact amount of maximum tolerable delay is subject to debate. It is generally accepted to be in the range of 100 to 600 ms. For example, the

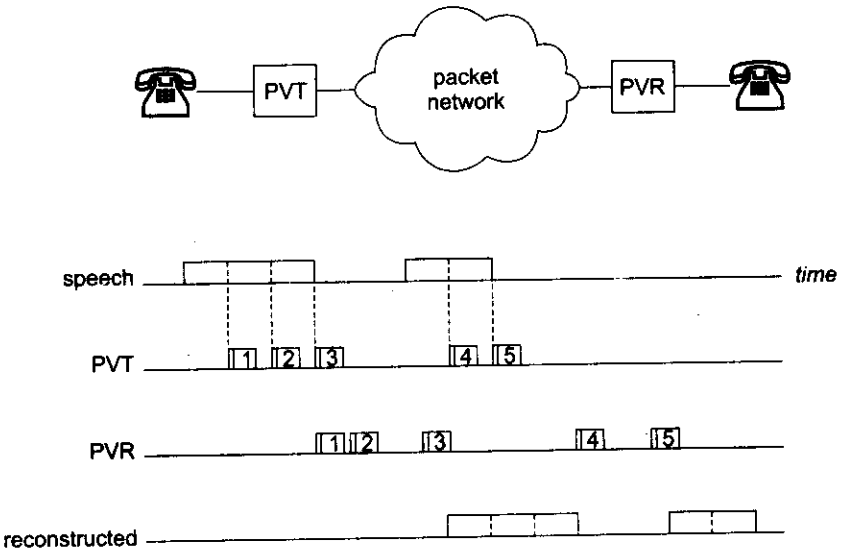

Figure 6.1 Packet voice [6.10]. ©1989 IEEE.

public telephone network has a maximum specification of 600 ms. In order to minimize packetization and storage delays, it has been proposed that voice packets should be relatively short, on the order of 200 to 700 bits, and generally should contain less than 10 to 50 ms of speech [6.6, 6.7, 6.8]. Network protocols should be simplified to shorten voice packet headers (for example, on the order of 4 to 8 bytes) although time stamps and sequence numbers are likely needed. Because a certain amount of distortion is tolerable, error detection, acknowledgements and retransmissions are unnecessary in networks with low error rates. Flow control can be exercised end-to-end by blocking calls. In addition, network switches can possibly discard packets under heavy traffic conditions. In this case, embedded coding has been proposed whereby speech quality degrades gracefully with the loss of information [6.9]. Packet voice is shown in Figure 6.1 [6.10]. It can be seen that the packets are generated at regular intervals during talksparts at the Packet Voice Transmitter (PVT). The reconstruction process at the Packet Voice Receiver (PVR) must compensate for the variable delay component by adding a controlled delay before playing out each packet. This is constrained by some value, D_{max}, which is the specified maximum percentage of packets that can be lost or miss playout. In addition to buffering voice packets, it might be desirable for the PVR to attempt to detect lost packets and to recover their information.

There are two basic approaches to the reconstruction process [6.11, 6.12, 6.13]. In the Null Timing Information (NTI) scheme, reconstruction does not use timing information (that is, time stamps) to determine packet delays through the network. The PVR adds a fixed delay D to the first packet of each talksparts as shown in Figure 6.2.

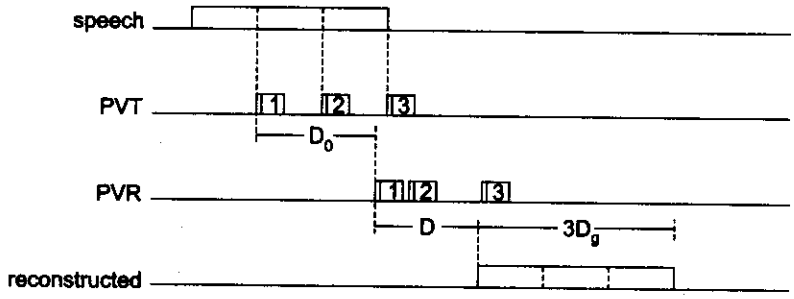

Figure 6.2 NTI reconstruction scheme [6.10]. ©1989 IEEE.

If D_0 is the transit delay of a first packet through the network and D_g is a packet-generation time (assumed to be constant), the total delay of the first packet from entry into the network to playout is

$$D_t = D_0 + D_g \tag{6.1}$$

Subsequent packets in the talkspart are played out at intervals of D_g after the first packet. Therefore, sequence numbers are required to indicate the relative positions of packets in the talkspart. If a packet is not present at the PVR at its playout time, it is considered lost. The choice of D involves a trade-off. Increasing D reduces the percentage of lost packets, but increases total end-to-end delays and the size of the queue at the PVR. D cannot be too large due to the constraint from D_{max} or too small due to P_{loss}. Because D_0 is random, the silence intervals between talksparts are not reconstructed accurately.

Example 6.1 Reconstruction of silences in an NTI scheme is shown in Figure 6.3. Let d and d' denote the values of D_0 for the talksparts preceding and following a silence interval(s). Suppose that d and d' are identically distributed with variance σ^2 and have some positive correlation r. Then, the error in the length of the reconstructed silence is

$$\varepsilon = d - d' \tag{6.2}$$

and has the variance

$$\text{var}(\varepsilon) = 2\sigma^2(1 - r) \tag{6.3}$$

which is directly proportional to the variance of packet delays. Evidently, the NTI scheme would be adequate only if a small delay variance could be guaranteed.

Because the scheme depends on the first packet of each talkspart, the loss of a first packet might cause confusion at the PVR.

If delay variability can be significant, a more elaborate reconstruction is necessary. In the Complete Timing Information (CTI) approach, the reconstruction process uses full timing information in the form of time stamps to determine each packet's delay accurately through the net-

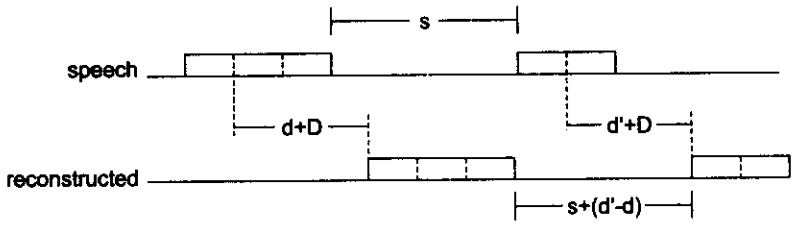

Figure 6.3 Reconstruction of silence scheme [6.10]. ©1989 IEEE.

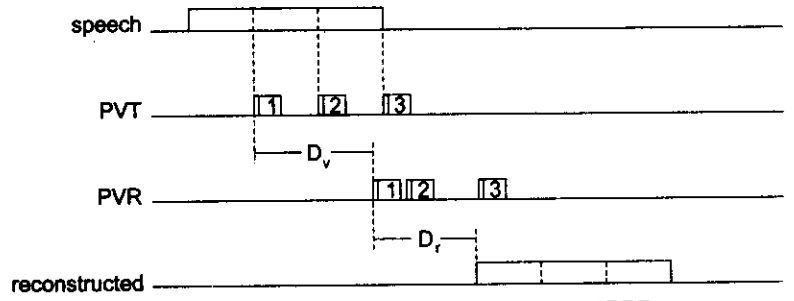

Figure 6.4 CTI reconstruction scheme [6.10]. ©1989 IEEE.

work, denoted D_v. As it can be seen from the Figure 6.4, the PVR adds a controlled delay D so that the total entry-to-playout delay D_t

$$D_t = D_v + D_r \tag{6.4}$$

is as uniform as possible for all packets. In addition to time stamps, sequence numbers are also desirable for detecting lost packets.

There are various choices for the format of the time-stamp fields. The most obvious choice is a global time stamp, but this requires precise synchronization of both PVT and PVR to a global clock. A second choice is to encode the relative time between consecutive packets. This means there is an unknown constant end-to-end-delay. A large time-stamp field is also required because the time between packets could be long. Finally, the time stamp can indicate the delay that a packet has accumulated in transit so far [6.11]. In this case, the time stamp might be more appropriately called a delay stamp. A packet is generated with a delay stamp initialized to zero. Each node increments the delay stamp by the amount of time that the packet has spent in that node, possibly including propagation delays along links as well.

6.1.2 Integrated Packet Networks

The economies and flexibility of integrated networks make them very attractive, and packet network architectures have the potential for realizing these advantages. However, the effective integration of speech and other signals, such as graphics, image and video into an Integrated Packet Network (IPN) can rearrange network design properties. Although processing speeds will con-

tinue to increase, it will also be necessary to minimize the nodal per-packet processing requirements imposed by the network design. Data signals must generally be received error free in order to be useful. The inherent structure of speech and image signals and the way in which they are perceived allows for some loss of information without significant quality improvement. This presents the possibility of purposely discarding limited information to achieve some other goal, such as the control of temporary congestion. One of the goals in IPNs is to construct a model that considers the entire IPN (transmitters, packet multiplexers and receivers) as a system to be optimized for higher speeds and capabilities [6.14]. In order to simplify the processing at network nodes, more complex processing at network edges can be allowed. The transmitter forms a packet switch, varying in its importance to the reconstruction of high-quality speech at the receiver. Packet multiplexers discard speech packets according to this delivery priority in order to control overload. The receiver then attempts to regenerate the information contained in any discard packets. Although this model is concerned specifically with speech, the approach can be extended to other structural signals, such as graphics, image and video signals.

A transmitter subsystem is shown in Figure 6.5. The transmitter first classifies speech segments according to models of the speech production process (voiced sounds, fricatives and plosives).

This model-based classification is used to remove redundancy during coding, to assign delivery properties and to regenerate discarded speech packets. After classification, the transmitter removes redundancy from the speech using a coding algorithm based on the determined model. For example, voiced sounds (vowels) could be coded with a block-oriented pitch prediction coder. After coding, the transmitter assigns a delivery priority to each packet based on the quality of regeneration possible at the receiver. In forming packets from speech segments, the delivery priority would be included in the network portion of the packet header. The classification and any coding parameters would be included in the end-to-end portion of the header. Packet multiplexers exist at each outgoing link of each network node as well as at each multiplexed network access point. A packet multiplexer subsystem with the arriving packet discarded is shown in Figure 6.6. Here, λ is the effective arrival rate, and μ represents the effective service

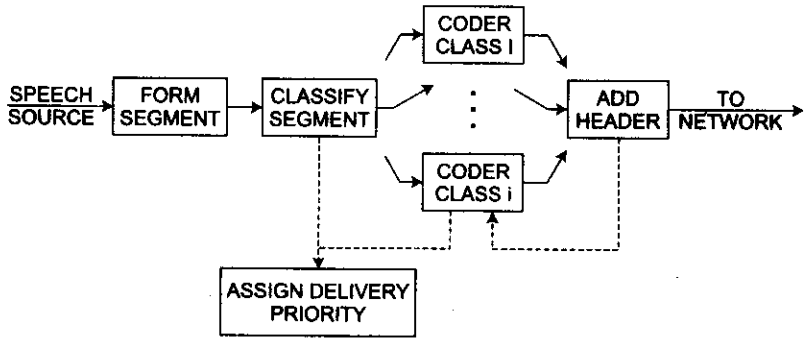

Figure 6.5 Transmitter subsystem [6.14]. ©1989 IEEE.

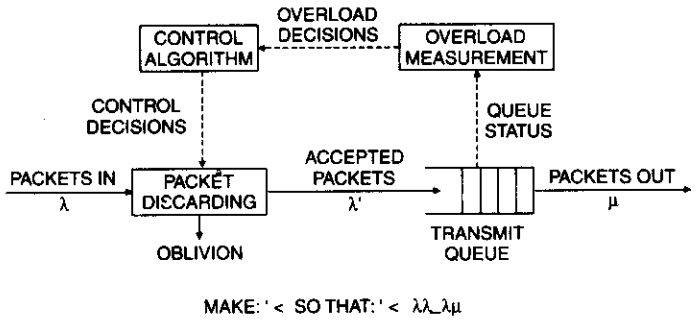

Figure 6.6 Packet multiplexer subsystem with arriving packet discarded [6.14]. ©1989 IEEE.

rate. Each packet multiplexer monitors local overload and discards packets, according to packet delivery priority (read from the network portion of the packet header) and is locally determined by the measure of overload level. It is assumed that arriving packets are discarded. It is also possible to discard already-queued packets. In addition, if error checking is performed by the nodes, any packet (data or speech) found to have an error is discarded.

The receiver decodes the samples in speech packets delivered to it based on the classification and coding parameters contained in the end-to-end header. It also determines the appropriate time to play them out. A receiver subsystem is shown in Figure 6.7. The receiver synchronization problem requires only packet sequence numbers. Global synchronization is administratively difficult, and time stamps must be modified at each packet multiplexer, requiring additional per-packet processing [6.11]. Potential speech detector impairments, such as clipping, are eliminated whenever the network is not overloaded. Even deriving periods of considerable overload, the received quality may be better if at least a few background noise packets are delivered and then used to regenerate noise that is similar in character to the actual noise. If a packet is lost for any reason (for example, discarded by the network because of overload or errors, excessively delayed in the network, and so forth) the receiver must first detect the loss by inspecting sequence numbers of those packets that have been received. It must further make a determination of the class of each lost packet so that the appropriate regeneration model can be applied using the previous header and sample history. A correct class determination will be critical to regenerating the lost information accurately. It can be easily done as follows. In a string of packets with the same class, we can virtually ensure that the first packet will be received by assigning it a high delivery priority. Assuming perfect delivery of these first packets, the class of any lost packet will match the class of the last received packet. Thus, the receiver's class decision can be virtually error free.

In summary, the advantages gained by taking a total system approach to an integrated packet network are as follows:

- A powerful overload control mechanism is provided.
- The structure of speech is effectively exploited.

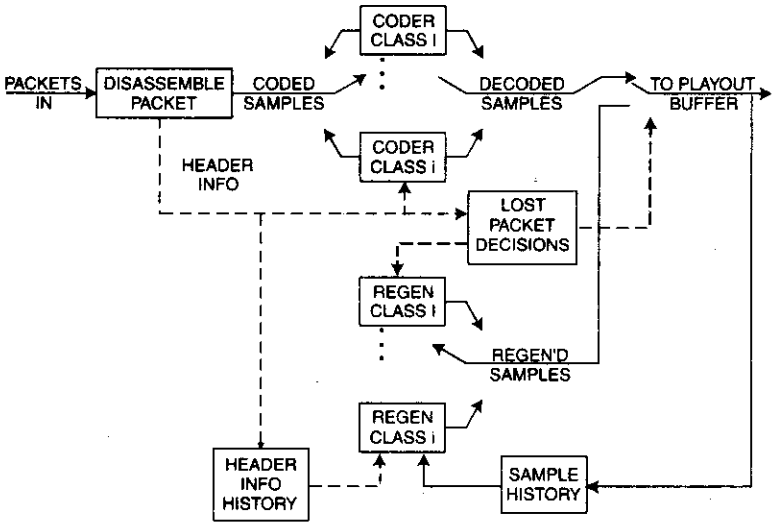

Figure 6.7 Receiver subsystem [6.14]. ©1989 IEEE.

- Extremely simple per-packet processing for overload control is allowed.
- Only one packet per speech segment is required.
- Receiver speech synchronization is simplified.
- Reduced per-packet error processing at packet multiplexers is possible.

6.1.3 Packet Video

Asynchronous transfer of video, which often is referred to as packet video, can be defined as the transfer of video signals across asynchronous Time Division Multiplex (ATDM) networks, such as IP and ATM. The video may be transferred for instantaneous viewing or for subsequent storage for replay at a later time. The former case has requirements on pacing so that the received video data can be displayed in a perceptually continuous sequence. The latter case can be seen as a large data transfer with no inherent time constraints. In addition to the requirement on pacing, the maximal transfer delay may also have bounds from camera to monitor if the video is a part of an interactive conversation or conference. These limits are set by human perception and determine when the delay starts at the information exchange. Parts of the signal may be lost or corrupted by errors during the transfer. This will reduce the quality of the reconstructed video, and, if the degradation is serious enough, it may cause the viewer to reject the service. Thus, the general topics of packet video are to code and to transfer video signals asynchronously under quality constraints.

The synchronous transfer mode combines the circuit-switched routing of telephony networks with the asynchronous multiplexing of packet switching. This is accomplished by establishing a connection (fixed route) through the network before accepting any traffic. The information is then sent in 53-octet long cells. The switches route cells according to address

information contained in each cell's five-octet header. Traffic on a particular link consists of randomly interleaved cells belonging to different calls. The network guarantees that all cells of a call follow the same route and, hence, get delivered in the same order as they were sent. The intention is that ATM networks should be able to guarantee the QoS in terms of cell loss and maximum delay, as well as maximum delay variations [6.15].

The IP differs in two major respects from ATM. There is no pre-established route and the packets are of variable length (up to 65,535 octets). IP does not give any guarantees on the delivery of the packets, and they may even arrive out of order if the routing decision is changed during the session. These issues will be addressed by the introduction of IPng in conjunction with RSVP. In IPng, often called IP (version 6), packets contain a 24-bit flow identifier in addition to the source and destination addresses and can be used in routers for operations like scheduling and buffer management to provide service guarantees. Delay and some loss is inevitable during transfers across both ATM and IP networks. The delay is chiefly caused by propagation and queuing. The queuing delay depends on the dynamic load variations on the links and must be equalized before video can be reconstructed. Bit errors can occur in the optics and electronics of the physical layer through thermal and impulsive noise. Loss of information is mainly caused by a multiplexing overload of such magnitude and duration that buffers in the nodes overflow. Video in digital form is a 3D signal. It is a time sequence of equidistantly spaced 2D pictures or frames. Frames can be samples of a real scene captured by a camera or a sensor. They may also be generated by computer graphics. The digitized frames of a video sequence can either be scanned sequentially row by row or be interlaced, where first the odd-numbered rows are scanned from top to bottom followed by the even-numbered rows. If the source produces a signal with RGB components, it is transformed into a YIQ format with one luminance component (Y) and two chrominance, or color, components (I and Q). The structure of a video stream is illustrated in Figure 6.8. The stream consists of frames that may be composed of fields if interlaced scanning is used. The fields are composed of lines of pixels where each pixel consists of color components (each has a fixed number of bits).

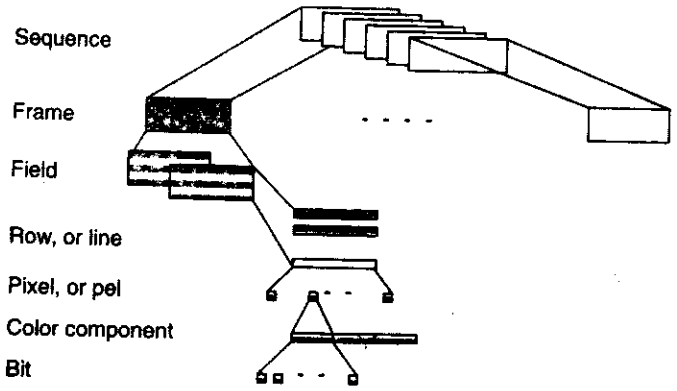

Figure 6.8 Structure of the video signal.

A video communication system is shown in Figure 6.9. The camera continuously captures a scene. The digitized video is passed to an encoder. A function that is often part of the encoder is the bit-rate control, which is used to regulate compression to adapt the bit rate to the channel in the network. Typically, a common reconstruction is that of the access capacity to the network. However, the constraint need not be a single upper limit on the rate, but could be a more general function. It can also be effected by flow-control messages from the network as well as from the receiver. Often the bit-rate control is the information segmentation and framing. A frame is a segment of data with added control information. Segments that are formed at the application level typically constitute the loss unit. Errors and loss in the network load to the loss of one or more application segments. Further segmentation occurs at the network level, where the data is segmented into multiplexing (IP packets or ATM cells), which is the loss unit for the network. The application layer segmentation and framing should simplify the handling of information loss that may occur during the transfer. The network framing is needed to detect and possibly correct bit and burst errors as well as packet or cell losses. The framing thus contains control information that may even include error-control coding. The receiver side performs functions that are reciprocal to the sending functions and may compensate for errors during the transfer. These functions include decoding, error handling, delay equalization, clock synchronization and digital-to-analog conversion.

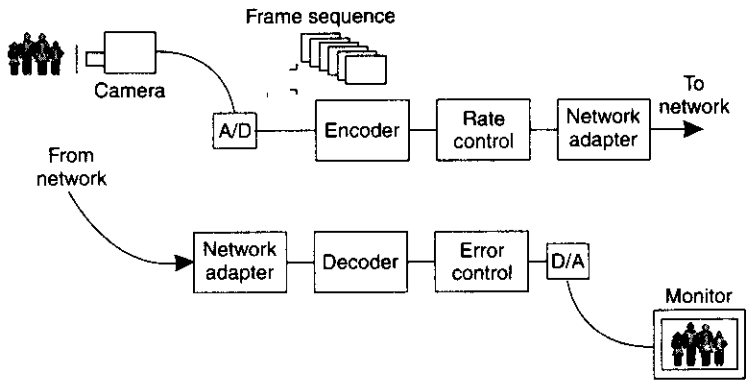

Figure 6.9 An example of a video transmission system.

6.2 Video Transport across Generic Networks

The emergence of digital storage and transmission of video has been driven by the availability of fast hardware at affordable prices, largely thanks to the economies obtained through standardization of compression algorithms. Digital video is being used in many applications ranging from videoconferencing (where two or more parties can carry out an interactive communication) to video on demand (where several users can access the video information stored at a central location). Each application has different requirements in terms of bit rate, end-to-end delay, delay jitter, and so forth. ATM technology is targeted to be used with BISDN and allows flexible and

efficient delivery of multimedia data, accommodating many different delays and bit-rate requirements [6.16]. Our goal is not to present an exhaustive survey of issues concerning video transport across generic networks. We choose to concentrate on the general ideas. Details can be found in several reports [6.17, 6.18, 6.19].

Video applications involve real-time display of the decoded sequence. Transmission across a CBR channel requires that there is a constant end-to-end delay between the time that the encoder processes a frame and the time at which that same frame is available to the decoder [6.17]. Because the channel rate is constant, it will be necessary to buffer the variable rate information generated by the video encoder. The size of the buffer memory will depend on the acceptable end-to-end delay. It is necessary to provide a rate-control mechanism that will ensure that the buffer does not overflow and that all the information arrives at the decoder. The basic idea is to lower the video quality for scenes of higher complexity to avoid overflow. The simplest approach to rate control relies on deterministic mapping of each buffer occupancy level to a fixed coder mode of operation [6.18]. Some models of the coder have been proposed to set up coding-rate predictions that are used to drive the buffer control [6.19]. In other cases, ideas from control theory are used to devise the buffer controller [6.20]. In general, the buffer control is designed for a particular encoding scheme, and the scheme-dependent heuristics tend to be introduced [6.21, 6.22]. A more detailed review of the problem in connection with a survey of the algorithms proposed for rate control can be found in Ortega, Ramchandran and Vetterli [6.23]. The goal of all these algorithms is to maximize the received quality given the available resources.

Whereas ISDN offers both circuit-switched and packet-switched channels, video transmission uses a circuit-switched channel. This is the case for most videoconferencing products. In this scenario, the transmission capacity available to the end-user is constant throughout the duration of the call. The main advantage of this approach is its reliability because the channel capacity is guaranteed. On the other hand, in computer networks, video is manipulated just as any other type of data. Video data is packetized and routed through the network sharing the transmission resources with other available services, such as remote login, file transfer, and so forth, which are also built on the top of the same transport protocols. Such systems are being implemented on LANs [6.24] and WANs [6.25, 6.26] with both point-to-point and multipoint connections. The systems are often referred to as best effort because they provide no guarantees on the end-to-end transmission delay and other parameters. In a best effort environment, the received video quality may change significantly over time.

ATM networks seek to provide the best of both worlds by allowing the flexibility and efficiency of computer networks while providing sufficient guarantees to permit reliable transmission of real-time services. Using ATM techniques allows flexible use of capacity, permitting dynamic routing and reuse of bandwidth. Because video compression algorithms produce a variable number of bits, the periods of low activity of one source can be reused by other sources. For example, for N sources (each requires a CBR channel at a rate R bits/s), it might be possible to transmit them together across a single channel with a rate less than RN. This reduction in capac-

ity is the so-called Statistical Multiplexing Gain (SMG). If all services are using their maximum capacity simultaneously, packets might be lost, so transmission can also be guaranteed most of the time. Contrary to the best-effort characteristic of most computer networks, ATM networks are designed so as to allow QoS parameters to be met, at least statistically. ATM networks aim to accommodate very heterogeneous services, thus allowing a customized set of QoS parameters to be selected by each application. The ATM design should be able to support both the best-effort network protocols and circuit-switched connections. However, although the QoS parameters are meaningful for non-real-time data, they are not the only factors to take into account for real-time video transmission. Video differs from other types of data in that acceptable transmission quality can be achieved even if some of the data is lost [6.16]. The effect of packet losses, which in other applications results in retransmission of data, can be reduced in the video case with appropriate encoding strategies combined with error-concealment techniques.

We now examine some of the network design issues, emphasizing those aspects that directly affect video transmission. Admission control is the task of deciding whether a new connection with a given set of requested QoS parameters can be allowed into the network. The connection should be admitted if it can be guaranteed to have the required QoS without degrading the QoS of other ongoing connections. Because video encoders produce variable rates, a key factor in the admission control problem is to find statistical models for the expected bit rates of video sources. A model characterizes the bit rate of a video connection at various time scales and attempts to capture the short- and long-term dependencies in the bit rate as well. Typical models are correlated with the number of bits for the previous frame [6.27] or Markov chains [6.28]. For a given model, the performance of the network model based on different routing and queuing strategies can be examined. The decision on whether to admit a call can be made based on the expected performance of the network. Admission control is much simpler for circuit-switched networks, because the transmission resources are constant throughout the duration of the call. The only issue is to find out whether currently unused resources are sufficient to carry the additional call. If they are sufficient, the call can be completed; otherwise, it will be rejected. In the ATM environment, the resources needed by each of the services change over time. Thus, the main problem is to estimate the likelihood that resources will be insufficient to guarantee QoS. Best-effort networks do not perform explicit admission control, but insufficient QoS during high-load conditions will drive users out or make them delay their connection [6.29]. Admission control is part of the negotiation process between user and network to set up a connection. The result of the negotiation is a contract that will specify the traffic parameters of the connection. Typical traffic parameters are peak cell rate and sustainable cell rate. These are operational measures of the offered bit rate and are implemented with counters.

The function called Usage Parameter Control (UPC), or policing mechanisms, has the goal of preventing sources from maliciously or unwillingly exceeding the traffic parameters negotiated at call setup. Typically, the network will look at policing methods that are directly linked to the negotiated traffic parameters. For instance, if a certain peak cell rate has been agreed upon, the policing mechanism may consist of a counter that tracks the peak rate and ver-

ifies that it does not exceed the negotiated value. One of the most popular policing mechanisms, due to its simplicity, is the so-called leaky bucket [6.30]. A leaky bucket is simply a counter incremented with each cell arrival and decremented at fixed intervals such that the decrement is equivalent to an average cell rate of R. The other parameter of the leaky bucket is the size of the bucket, that is, the maximum allowable value for the counter. The network can detect violations by monitoring whether the maximum value of the counter is reached. If the same cells are found to be violating the policing functions, the network can decide to delete them or to just mark them for possible deletion in case of congestion. The choice of policing function is important because it may determine the type of rate that the sources will transmit through the network.

Video-encoding algorithms for ATM transmission need to be robust to packet losses. This can be achieved in part by using a multiresolution encoding scheme along with different levels of priorities for the cells corresponding to each resolution. Additionally, error-concealment techniques can be used to mark to some extent the perceptual effects in the decoded sequence of the loss information.

Multiresolution encoding schemes separate the information into two or more layers or resolutions. The coarse resolution contains a rough approximation of the full resolution image or sequence. The enhancement or details resolution provides the additional information needed to reconstruct at the decoder the full resolution sequence at the targeted quality. The coarse resolution sequence is obtained by reducing the spatial or temporal resolution of the sequence or by having images of lower quality. A survey of multiresolution encoding techniques can be found in Vetterli and Uz [6.31]. To take advantage of the multiresolution encoding, the information is packetized into two classes of packets according to whether the priority bit provided by the ATM format is set or not [6.16]. The coarse resolution will be transmitted using high priority packets while the detail resolution will be sent with the low priority ones. Using the properties so that the packets with lower priority are discarded first in case of congestion is beneficial in terms of the end-to-end quality [6.32].

A further advantage of using multiresolution coding schemes is that they enable efficient error concealment techniques [6.33, 6.34, 6.35, 6.36]. The idea is to use the available information, that is, packets that were not lost, to interpolate the missing information. When multiresolution coding is used, the information decoded from only the lower resolution layer may be sufficiently good. Other approaches that have been proposed involve interleaving the information so that a cell loss causes minor perceptual degradation in several image blocks (10 to 100) rather than severe degradation in just a few blocks (1 to 10).

ATM transmission provides the possibility of transporting a variable number of bits per frame and thus could seem to make the use of rate control unnecessary. This view is not realistic because each connection will be specified by a series of traffic parameters that will be monitored by the network. Transmission over th limits set by the traffic characteristics may result in lost packets, so rate control is still necessary [6.17]. We can make the distinction between rate control and rate shaping. Rate control entails changing the rate produced by the encoder, and rate

shaping only affects the times at which cells are re-sent to the network, but not the total amount of information transmitted.

6.2.1 Layered Video Coding

An often-cited approach for coping with receiver heterogeneity in real-time multimedia transmission is the use of layered media streams. In this model, the source distributes multiple levels of quality simultaneously across multiple network channels. In turn, each receiver individually tunes its reception rate by adjusting the number of layers that it receives. The net effect is that the signal is delivered to a heterogeneous set of receivers at different levels of quality using a heterogeneous set of rates. To fully realize this architecture, we must solve two subproblems: the layered compression problem and the layered transmission problem. In other words, we must develop a compression scheme that allows us to generate multiple levels of quality using multiple layers simultaneously with a network delivery mode that allows us to selectively deliver a subset of layers to individual receivers. We first define the layered compression problem.

Layered Compression

Given a sequence of video frames $\{F_1, F_2, ...\}$, for example, $F_k \in [0, 255]^{640 \times 480}$ for gray-scale NTSC video, we want to find an encoding E that maps a given frame into L discrete codes, (that is, into L layers): $E: F_k \rightarrow \{C_k^1, ..., C_k^L\}$ and further a decoding D that maps a subset of $M \leq L$ codes into a reconstructed frame, $\hat{F}_k^M$, $D: \{C_k^1, ..., C_k^M\} \rightarrow \hat{F}_k^M$ with the property that $d\left(F_k, \hat{F}_k^M\right) \geq d\left(F_k, \hat{F}_k^n\right)$ for $0 \leq m \leq n \leq L$ and a suitably chosen metric d (for example, mean squared error or a perceptual distortion measure). With this decomposition, an encoder can produce a set of codes that are striped across multiple network channels $\{N_1, ..., N_L\}$ by sending codes $\{C_1^k, C_2^k, ...\}$ across N_k. A receiver can then receive a subset of the flows $\{N_1, ..., N_M\}$ and reconstruct the sequence $\left\{\hat{F}_1^M, \hat{F}_2^M, ...\right\}$. One approach for delivering multiple levels of quality across multiple network connections is to encode the video signal with a set of independent encoders each producing a different output rate. Hence, we can choose a $D = (D_1, ..., D_L)$ where $D_m: C_k^m \rightarrow \hat{F}_k$. This approach, often called simulcast, has the advantage that we can use existing codecs and/or compression algorithms as system components. Figure 6.10 illustrates the simplicity of a simulcast coder. It produces a multirate set of signals that are independent of one another. Each layer provides improved quality, but does not depend on subordinate layers. Here, we show an image at multiple resolutions, but the refinement can occur across dimensions of line frame rate or SNR. A video signal is duplicated across the inputs to a bulk of independent encoders. These encoders compress the signal to a different rate and different quality. Finally, the decoder receives the signal independent of the other layers. In simulcast coding, each layer of video representing a resolution or quality is coded independently. Thus, a single layer (nonscalable)

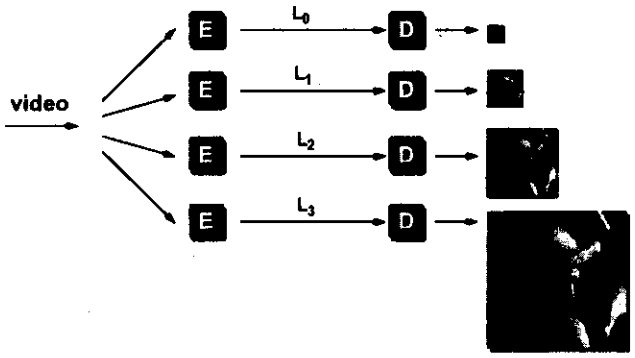

Figure 6.10 Simulcast coder.

decoder can decode any layer. In simulcast coding, total available bandwidth is simply portioned depending on the quality desired for each independent layer that needs to be coded. It is assumed that independent decoders would be used to decode each layer [6.37].

In contrast, a layered coder exploits correlation across subflows to achieve better overall compression. The input signal is compressed into a number of discrete layers, arranged in a hierarchy that provides progressive refinement. For example, if only the first layer is received, the decoder will produce the lowest quality version of the signal. On the other hand, if the decoder receives two layers, it will combine the second-layer information with the first layer to produce improved quality. Overall, the quality progressively improves with the number of layers that are received and decoded.

Figure 6.11 gives a rough sketch of the trade-off between the simulcast and layered approaches from the perspective of rate-distortion theory. Each curve traces the distortion incurred for imperfectly coding an information source at the given rate. Distortion rate functions for an ideal coder $D_I(R)$, a real coder $D_R(R)$, a layered coder $D_L(R)$ and a simulcast coder $D_S(R)$ are presented. The distortion measures the quality degradation between the reconstructed and original signals. The ideal curve $D_I(R)$ represents the theoretical lower bound on distortion

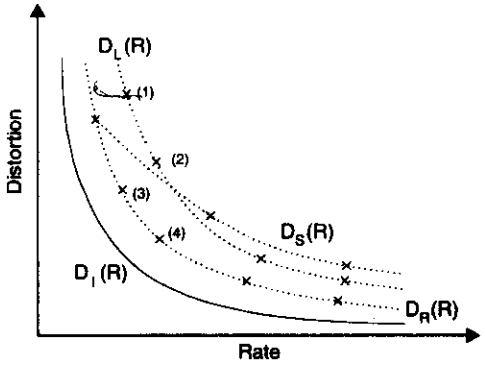

Figure 6.11
Rate distortion characteristics.

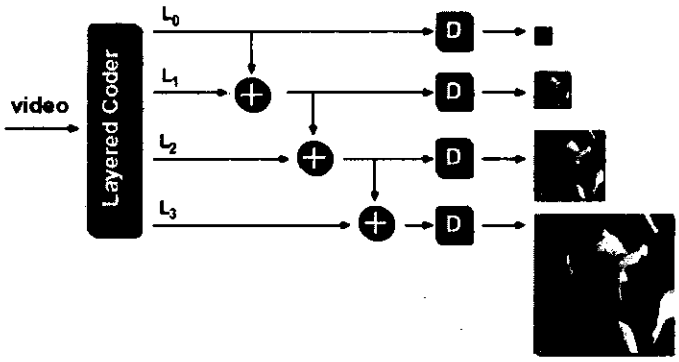

Figure 6.12 Conceptual structure of a layered video.

achievable as a function of rate. A real coder $D_R(R)$ can perform close to the ideal curve, but never better. The advantage of layer representation is that both the encoder and decoder can travel along the distortion rate curve. That is, to move from point (1) to (2) on $D_L(R)$, the encoder carries out incremental computation and produces new output that can be appended to the previous output. Conversely, to move from point (3) to (4) on $D_R(R)$, the encoder must start from scratch and compute a completely new output string. Finally, a simulcast coder $D_S(R)$ incurs the most overhead because each operating point redundantly contains all of the operating points of lesser rate.

The structure of a layered video coder is given in Figure 6.12. The input video is compressed by a layered coder that produces a set of logically distinct output strings. The decoder module D is capable of decoding any cumulative set of bitstrings. Each additional string produces an improvement in reconstruction quality.

Layered Transmission

By combining the approach of layered compression with a layered transmission system, we can solve the multicast heterogeneity problem. In this architecture, the simulcast source produces a layered stream where each layer is transmitted on a different network channel. The network forwards only the number of layers that each physical link can support. Each user receives the best quality signal that the network can deliver. The network must be able to drop layers selectively at each bottleneck link. The concept of layered video coding was first introduced in the context of ATM networks [6.38, 6.39]. The video information is divided into several layers, with lower layers containing low-resolution information and higher layers containing the fine information. Such a model enables integration of video telephony and broadcast video services. In the former case, where a bandwidth is at a premium, lower layers can provide the desired quality. In broadcast applications, a variable number of higher layers can be integrated with the lower ones to provide the quality and the bit rate that is compatible with the receiver.

6.2.2 Error-Resilient Video Coding Techniques

Error-resilience techniques for real-time video transport across unreliable networks include protocol and network environments and their characteristics, encoder error-resilience tools; decoder

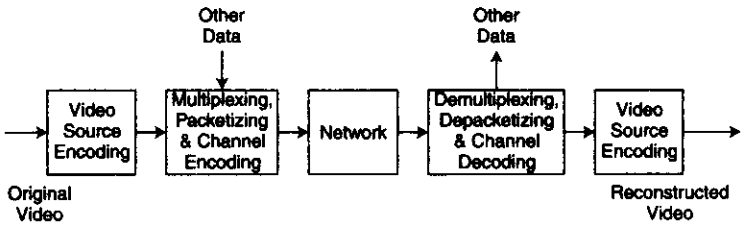

Figure 6.13 Video communication system.

error-concealment techniques and techniques that require cooperation among encoder, decoder and the network. A typical video transmission system involves five steps as shown in Figure 6.13. The video is first compressed by a video encoder to reduce the data rate. The compressed bit stream is then segmented into fixed or variable length packets and multiplexed with other data types such as audio. The packets might be sent directly across the network if the network guarantees bit-error-free transmission. Otherwise, they usually undergo a channel-encoding stage, typically using Forward Error Correction (FEC), to protect them from transmission errors. At the receiver end, the received packets are FEC decoded and unpacked. The resulting bit stream is then input to the video decoder to reconstruct the original video. In practice, many applications embed packetization and channel encoding in the source coder as an adaptation layer to the network.

To make the compressed bit stream resilient to transmission errors, one must add redundancy into the stream so that it is possible to detect and correct errors. Such redundancy can be added in either the source or channel coder. The classical Shannon information theory states that one can separately design the source and channel coders to achieve error-free delivery of a compressed bit stream as long as the source is represented by a rate below the channel capacity. Therefore, the source coder should compress a source as much as possible for a specified distortion. The channel coder can then add redundancy through FEC to the compressed stream to enable the correlation of transmission errors. All the error-resilient encoding methods make the source coder less efficient than it can be so that erroneous or missing bits in a compressed stream will not have a disastrous effect on the reconstructed video quality. This is usually accomplished by carefully designing both the predictive coding loop and variable length coder to limit the extent of error propagation.

Mechanisms devised for combating transmission errors can be categorized into three groups:

- Those introduced at the source and channel encoder to make the bit stream more resilient to potential errors.
- Those invoked at the decoder upon detection of errors to conceal the effect of errors.
- Those that require interactions between the source encoder and decoder so that the encoder can adapt its operations based on the loss conditions detected at the decoder.

We will refer to all of them as error-resilience techniques.

Error-Resilient Encoding

In this approach, the encoder operates so that transmission errors on the coded bit stream will not adversely affect the decoder operation and lead to unacceptable distortions in the reconstructed video quality. Compared to coders that are optimized for coding efficiency, error-resilient coders typically are less efficient in that they use more bits to obtain the same video quality in the absence of any transmission errors. The extra bits are called redundancy bits, and they are introduced to enhance the video quality when the bit stream is corrupted by transmission errors. The design goal in error-resilient coders is to achieve a maximum gain in error resilience with the smallest amount of redundancy.

There are many ways to introduce redundancy in the bit stream. Some of the techniques help to prevent error propagation, and others enable the decoder to perform better error concealment upon detection of errors. Yet another group of techniques is aimed at guaranteeing a basic level of quality and providing a graceful degradation upon the occurrence of transmission errors [6.40, 6.41].

One main cause for the sensitivity of a compressed video stream to transmission errors is that a video coder uses VLC to represent various symbols. Any bit errors or lost bits in the middle of a code word can not only make this code word undecodable, but they can also make the following code words undecodable, even if they are received correctly.

One simple and effective approach for enhancing encoder error resilience is by inserting resynchronization markers periodically. Usually, some header information is attached immediately after the resynchronization information. Obviously, insertion of resynchronization markers will reduce the coding efficiency [6.41]. In practical video-coding systems, relatively long synchronization code words are used.

With RVLC, the decoder can not only decode bits after a resynchronization code word, but also decode the bits before the next resynchronization code word, from the backward direction. Thus, with RVLC, fewer correctly received bits will be discarded, and the affected area by a transmission error will be reduced. RVLC can help the decoder to detect errors that are not detectable when non-RVLC is used, or it can provide more information on the position of the errors and increase the amount of data unnecessarily discarded. RVLC has been adopted in both MPEG-4 and H.263 in conjunction with the insertion of synchronization markers.

Because of the syntax constraint present in compressed video bit streams, it is possible to recover data from a corrupted bit stream by making the corrected stream conform to the right syntax. Such techniques are very much dependent on the particular coding scheme. The use of synchronization codes, RVLC and other sophisticated entropy coding means, such as error-resilient entropy coding, can all make such recovery more feasible and more effective.

Another major cause for the sensitivity of a compressed video to transmission errors is the use of temporal prediction. After an error occurs so that a reconstructed frame at the decoder differs from that assumed at the encoder, the reference frames used in the decoder from there on will differ from those used at the encoder. Consequently, all subsequent reconstructed frames will be in error. The use of spatial prediction for the DC coefficients and motion vectors will also cause

error propagation, although it is confined within the same frame. In most video-coding standards, such spatial prediction, and therefore error propagation, is further limited to a subregion in a frame.

One way to stop temporal error propagation is by periodically inserting intracoded pictures. For real-time applications, the use of intraframes is typically not possible due to delay constraints. However, the use of a sufficiently high number of intracoded pictures has turned out to be an efficient and highly scalable tool for error resilience. When applying intracoded pictures for error-resilience purposes, both the number of such intracoded pictures and their spatial placement have to be determined. The number of necessary intraframes is obviously dependent on the quality of the connection. The currently best known way for determining both the correct number and placement of intraframes for error-resilience purposes is the use of a loss-aware rate distortion optimization scheme [6.42].

Another approach to limit the extent of error propagation is to split the data domain into several segments and to perform temporal/spatial prediction only within the same segment. In this way, the error in one segment will not affect another segment. One such approach is to include even-indexed frames in one segment and odd-indexed frames into another segment. Even frames are only predicted from even frames. This approach is called video redundancy coding [6.43]. It can also be considered as an approach for accomplishing multiple description coding. Another approach is to divide a frame into regions, and a region can only be predicted from the same region of the previous frame. This is known as Independent Segment Decoding (ISD) in H.263.

By itself, layered coding is a way to enable users with different bandwidth capacities or decoding powers to access the same video at different quality levels. To serve as an error-resilience tool, layered coding must be paired with UEP in the transport system so that the base layer is protected more strongly, for example, by assigning a more reliable subchannel, using stronger FEC codes or allowing more retransmissions [6.44]. There are many ways to divide a video signal into two or more layers in the standard block-based hybrid video coder. For example, a video can be temporally down-sampled, and the base layer can include the bit stream for the low-frame-rate video, whereas the enhancement layers can include the error between the original video and the up-sampled one from the low frame-rate coded video. The same approach can be applied to the spatial resolution so that the base layer contains a small frame-size video. The base layer can also encode the DCT coefficients of each block with a coarser quantizer, leaving the fine details to be specified in the enhancement layers. Finally, the base layer may include the header and motion information, leaving the remaining information for the enhancement layer. In MPEG and H.263 terminologies, the first three options are known as temporal, spatial and SNR scalabilities, respectively, and the last one is data partitioning.

As with layered coding, Multiple Description Coding (MDC) also codes a service into several substreams, known as descriptions, but the decomposition is such that the resulting descriptions are correlated and have similar importance. For each description to provide a certain degree of quality, all the descriptions must share some fundamental information about the

source and must be correlated. On the other hand, this correlation is also the rate of redundancy in MDC. An advantage of MDC over layered coding is that it does not require special provisions in the network to provide a reliable channel. To accomplish their respective goals, layered coding uses a hierarchical, decorrelating decomposition, whereas MDC uses a nonhierarchical, correlating decomposition [6.45].

The objective of error-resilient encoding is to enhance robustness of compressed video to packet loss. The standardized error-resilient encoding schemes include resynchronization marking, data partitioning, and data recovery. For video transmission across the Internet, the boundary of a packet already provides a synchronization point in the variable-length coded bit stream at the receiver side. With MDC, we have robustness to loss of enhanced quality. If a receiver gets only one description (other descriptions being lost), it can still reconstruct video with acceptable quality. If a receiver gets multiple descriptions, it can combine them to produce a better reconstruction than that produced from any one of them. To make each description provide acceptable usual quality, each description must carry sufficient information about the original video. This will reduce the compression efficiency compared to conventional Single Description Coding (SDC). In addition, although more combined descriptions provide a better visual quality, a certain degree of correlation between the multiple descriptions has to be embedded in each description, resulting in further reduction of the compressed efficiency.

Decoder Error Concealment

Decoder error concealment refers to the recovery or estimation of lost information due to transmission errors. Given the block-based hybrid coding paradigm, three types of information may need to be estimated in a damaged MB: the texture information, including the pixel and DCT coefficients values for either an original image block or a predictive error block; the motion estimation, consisting of Motion Vectors (MV) for an MB in either P- or B-mode and, finally, the coding mode of MB. A simple and yet very effective approach to recover a damaged MB in the decoder is by copying the corresponding MB in the previously decoded frame, based on the MV for this MB. The recovery performance by this approach is critically dependent on the availability of the MV. To reduce the impact of the error in the estimated MVs, temporal prediction may be combined with spatial interpolation. Another simple approach is to interpolate pixels in a damaged block from pixels in adjacent correctly received blocks as all blocks or MBs in the same row are put into the same packet. The only available neighboring blocks are those in the current row and the row above. Because most pixels in these blocks are too far away from the missing samples, usually only the boundary pixels in neighboring blocks are used for interpolation [6.46]. Instead of interpolating individual pixels, a simple approach is to estimate the DC coefficient (that is, the mean value) of a damaged block and replace the damaged block by a constant equal to the estimated DC value. The DC value can be estimated by averaging the DC values of surrounding blocks [6.47]. One way to facilitate such spatial interpolation is by an interleaved packetization mechanism so that the loss of one packet will damage only every other block.

A problem with the spatial interpolation approach is how to determine an appropriate interpolation filter. Another shortcoming is that it ignores received DCT coefficients, if any. These problems are resolved in [6.34] by requiring the recovered pixels in a damaged block to be smoothly connected with its neighboring pixels (Zhu, Wang, and Shaw), both spatially in the same frame and temporally in the previous/following frames.

Another way of accomplishing spatial interpolation is by using spatial interpolation using the Projection Onto Convex Set (POCS) method [6.48, 6.49]. The general idea behind POCS-based estimation methods is to formulate each constraint about the unknowns as a convex set. The optimal solution is the intersection of all the convex sets, which can be obtained by recursively projecting a previous solution onto individual convex sets. When applying POCS for recovering an image block, the spatial smoothness criterion is formulated in the frequency domain by requiring the DFT of the recovered block to have energy only in several low frequency coefficients. If the damaged block is believed to contain an edge in a particular direction, one can require the DFT coefficients to be distributed along a narrow strip orthogonal to edge direction, that is, low pass along the edge direction and all pass in the orthogonal direction. The requirement on the range of each DFT coefficient magnitude can also be converted into a convex set. Because the solution can only be obtained through an iterative procedure, this approach may not be suitable for real-time applications.

Error-Resilient Entropy Code

Video coders encode the video data using VLCs. Thus, in an error-prone environment, any error would propagate throughout the bit stream unless we provide a means of resynchronization. The traditional way of providing resynchronization is to insert special synchronization code words into the bit stream. These code words should have a length that exceeds the maximum VLC code length and also be robust to errors. Thus, a synchronization code should be recognized even in the presence of errors. The Error-Resilient Entropy Code (EREC) is an alternative way of providing synchronization. It works by rearranging variable-length blocks into fixed-length slots of data prior to transmission. The EREC is applicable to variable-length codes. For example, these blocks can be macroblocks in H.263. Thus, the output of the coding scheme is variable-length blocks of data. Each variable-length block must be a prefix code. This means that, in the presence of errors, the block can be decoded without reference to previous or future blocks. The decoder should also be able to know when it has finished decoding a block. The EREC frame structure consists of N slots of length s_i bits. This, the total length of the frame, is $T = \sum_{i=1}^{N} s_i \ [bits]$. It is assumed that the values of T, N and s_i are known to both the encoder and the decoder. Thus, the N slots of data can be transmitted sequentially without the risk of loss of synchronization. EREC reorganizes the bits of each block into the EREC slots. The decoding can be performed by relying on the ability to determine the end of each variable-length block. Figure 6.14 shows an example of the operation of the EREC algorithm. There are six blocks of lengths 11, 9, 4, 3, 9, 6 and six equal length slots with s_i=7 bits. In the first stage of the algorithm, each block of data is allocated to a corresponding

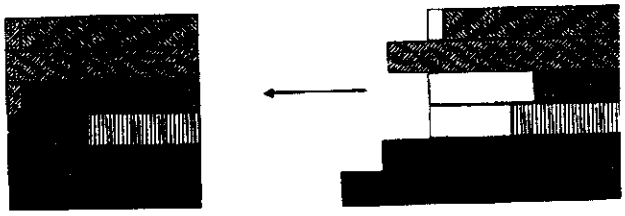

Figure 6.14 An example of the EREC algorithm [6.48]. ©1996 IEEE.

EREC slot [6.48]. Starting from the beginning of each variable-length block, as many bits as possible are placed into the corresponding slot. In the following stages of the algorithm, each block with data yet to be coded searches for slots with space remaining. If there is space available in the slot searched, all or as many bits as possible are placed into that slot. If there is enough space in the slots, the reallocation of the bits will be completed within N stages of the algorithm. The final result of the EREC algorithm is shown in Figure 6.14. In the absence of errors, the decoder starts decoding each slot. If it finds the block end before the slot end, it knows that the rest of the bits in that slot belong to other blocks. If the slot ends before the end of the block is found, the decoder has to look for the rest of the bits in another slot. In case one slot is corrupted, the location of the beginning of the rest of the slots is still known, and the decoding of them can be attempted.

6.2.3 Scalable Rate Control

The main challenge in designing a multimedia application across a communication network is how to deliver multimedia streams to users with minimal replay jitters. In general, a network-based multimedia system can be conceptually viewed as a layer-structure system, which consists of application layer on the top, compression layer, transport layer and transmission layer, as shown in Figure 6.15 [6.49]. To diminish the impact on the video quality due to the delay jitter and available network resources (for example, bandwidth and buffers), traffic shaping and SRC are qualified candidates at two different system levels. Traffic shaping is a transport-layer approach, and SRC is a compression-layer approach.

The basic concept behind the traffic-shaping approach is that, before the encoded video bit stream is injected into the network for transmission, the traffic pattern is already shaped with the desired characteristics, such as maximal delay bounds and peak instantaneous rate [6.50]. Therefore, all the system components along the network path from the sender to the receiver can be configured to meet the QoS as desired by allocating the appropriate resources a priori. On the other hand, the SRC approach is a compression-layer technique where the source video sequence is compressed according to the application's requirement and available network resource.

In the development of an SRC scheme, we need to consider a common feature of employing an Internet-frame coding between two consecutive video frames in several widely used video-compression schemes such as MPEG-1, MPEG-2 and H.263. Although the interframe coding scheme exploits the similarity usually found in encoding two consecutive video frames and achieves significant coding efficiency, the output with a variable-length video bit stream is

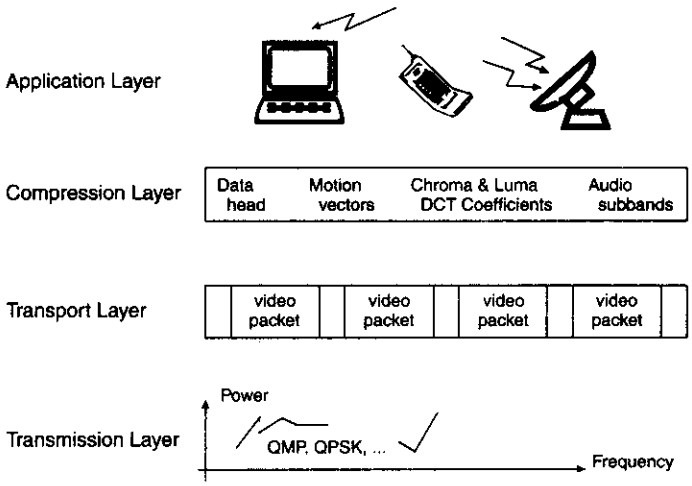

Figure 6.15 Layer structure of a network-based multimedia system [6.49]. ©2000 IEEE.

not well suited for a fixed-rate communication channel. To better use network resources and to transmit coded video bit stream as accurately as possible, the network parameters and encoding parameters should be jointly considered, and their relationship should be modeled accurately. Technically speaking, rate control is a decision-making process where the desired encoding rate for a source video can be met accurately by properly setting a sequence of Quantization Parameters (QP). To cope with various requirements of different coding environments and applications, a rate-control scheme needs to provide sufficient flexibility and scalability. For example, multimedia applications are categorized into two groups, which are VBR application and CBR application. For VBR applications, rate control attempts to achieve the optimum quality for a given target rate. In CBR and real-time applications, a rate-control scheme must satisfy low-latency and buffer constraints. In addition, the rate-control scheme has to be applicable to a variety of sequences and bit rates. Thus a rate-control scheme must be scalable for various bit rates, various spatial resolutions, various temporal resolutions, various coders (DCT and wavelet) and various granularities of VO.

The purpose of rate control is consequently to enforce the specification of the bit stream. The general system is shown in Figure 6.16. The bit stream from the coder is fed into a buffer at a rate $R'(t)$, and it is served at some rate $\mu(t)$ so that the output rate $R(t)$ meets the specified behavior. The bit stream is smoothed by the buffer whenever the service rate is below the input rate. The size of the buffer is determined by delay and implementation constraints. In the encoder, the compression rate is increased when buffer overflow is at risk. The issue is to reduce the variability of the rate function $R(t)$ while minimizing the effects of the consistency of the perceptual quality [6.51, 6.52]. The joint problem of traffic characterization and rate control is to find a suitable description of the bit stream that is sufficiently useful to the network and that can be enforced without overly throttling the compression rate. Provided that a model has been cho-

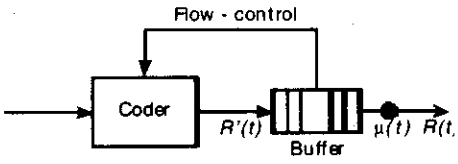

Figure 6.16
Bit-rate control.

sen, the user should not estimate the parameters for it and find a way to regulate the service rate $\mu(t)$ so that R(t) strictly obeys the specification. The network would then have the possibility to verify that the traffic is in accordance with its specifications.

Rate Control Techniques

In developing a rate-control technique, there are two widely used approaches:

- Analytical model-based approach
- Operational rate distortion R(D)-based approach

In the model-based approach, various distribution characteristics of signal source along with associated guarantees are considered. Based on the selected model, a closed-form solution is derived using optimization theory. Such a theoretical optimization solution cannot be implemented easily because there is only a finite discrete set of quanitzers and the source signal model varies spatially. Alternatively, an operational R(D)-based approach is used in a practical coding environment. For example, to minimize the overall coding distortion subject to a total bit budget constraint, lots of techniques based on dynamic programming or Lagrangian multiplier for optimization solutions have been developed [6.53, 6.54, 6.55, 6.56]. These methods share the similar concepts of data preanalysis. By analyzing the R(D) characteristics of future frames, the bit-allocation strategy is determined afterward. The Lagrangian multiplier is a well-known technique for optimal bit allocation in image and video coding, but with an assumption that the source consists of statistically independent components. Thus, an interframe-based coding may not find the Lagrangian multiplier approach applicable because of the temporal dependency.

Frame dependencies are taken into account in bit-rate control [6.54]. However, potentially high complexity with increasing operating R(D) points make this method unsuitable for the applications requiring interactivity or low encoding delay. In [6.57], Ding investigated a joint encoder and channel rate-control scheme for VBR video across ATM networks and claimed that the rate control scheme has to balance both issues of consistent video quality in the encoder side and bit-stream smoothness for statistical multiplexing gain in the network side. A parametric R(D) model for MPEG encoders, especially for the picture-level rate control, has been proposed [6.58]. Based on the bit rate "m quant" model, the desired "m quant" is calculated and used for encoding every MB by combining with appropriate quantization matrix entry in a picture. A normalized R(D) model-based approach has been also developed for H.263-compatible video codecs. By providing good approximation of all 32 rate-distortions relations, the authors claim that the proposed model offers an efficient and less-memory-requirements approach to approxi-

mate the rate and distortion characteristics for all QPs [6.59]. Rate-control techniques for MPEG-4 object-level and MB level video coding were proposed in [6.60, 6.61]. However, most of the aforementioned techniques only focus on a single coding environment, either frame level, object level or macro level. None of these techniques demonstrates its applicability to MPEG-4 video coding, including the previous three coding granularities simultaneously.

In Lee, Chiang and Zhang [6.49], based on a revised quadratic R(D) model, SRC proposes a single framework that is designed to meet both VBR without delay constraints and CBR with low latency and buffer constraints. With this scalable framework based on a new R(D) model and several new concepts, not only more accurate bit rate control with buffer regulation is achieved, but scalability is also preserved for all test video sequences in various applications [6.62]. By considering video contents and coding complexity in the quadratic R(D) model, the rate-control scheme with joint buffer control can dynamically and appropriately allocate the bits among VOs to meet the overall bit-rate requirement with uniform video quality.

Because of the precision of the R(D) model and ease of implementation, the rate-control scheme with the following new concepts and techniques has been adopted as part of the rate-control scheme in MPEG-4 standard:

- A more accurate second order R(D) model for target bit-rate estimation
- A sliding-window method for smoothing the impact of scene change
- An adaptive selection criterion of data points for a better model updating process
- An adaptive threshold shape control for better use of bit budget
- A dynamic bit-rate allocation among VOs with different coding complexities

This rate-control scheme provides a scalable solution, meaning that the rate-control technique offers a general framework for multiple layers of control for objects, frames and MBs in various coding contexts.

Theoretical Foundation of the SRC

In the R(D) model, the distortion is measured in terms of quantization parameter [6.49]. The block diagram is presented in Figure 6.17.

The rate control consists of four stages: initialization stage, pre-encoding stage, encoding stage and post-encoding stage. Assuming that the source statistics are Laplacian distributed [6.63]:

$$P(x) = \frac{\alpha}{2} e^{-\alpha|x|}, \quad \text{where } -\infty < x < \infty \tag{6.5}$$

The distortion measure is defined as

$$D\left(x, \hat{x}\right) = \left|x - \hat{x}\right| \tag{6.6}$$

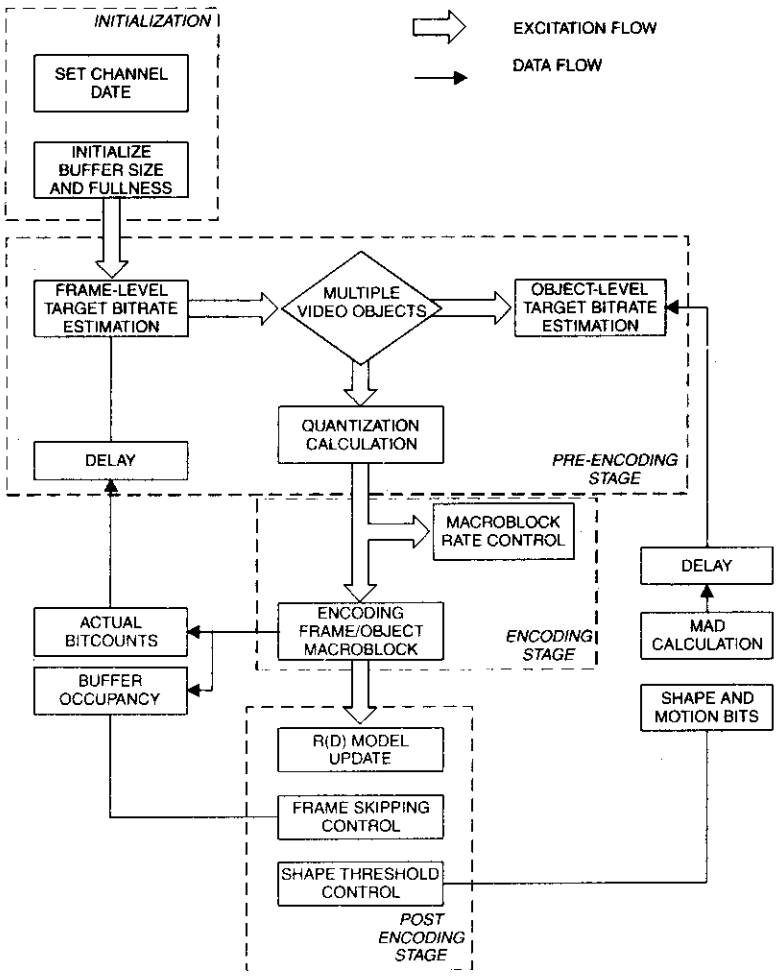

Figure 6.17 Block diagram of the SRC [6.49]. ©2000 IEEE.

There is a closed-form solution for the R(D) functions derived in Viterbi and Omura [6.64]:

$$R(D) = \ln\left(\frac{1}{\alpha D}\right) \tag{6.7}$$

where

$$D_{\min} = 0, \quad D_{\max} = \frac{1}{\alpha}, \quad 0 < D < \frac{1}{\alpha} \tag{6.8}$$

The R(D) function is expanded into a Taylor series

$$R(D) = \left(\frac{1}{\alpha D} - 1\right) - \frac{1}{2}\left(\frac{1}{\alpha D} - 1\right)^2 + R_3(D) = -\frac{3}{2} + \frac{2}{\alpha} D^{-1} - \frac{1}{2\alpha^2} D^{-2} + R_3(D) \tag{6.9}$$

Based on this observation, a new model to evaluate the target bit rate before performing the actual encoding process is presented in Lee, Chiang and Zhang [6.49]. The new model is formulated as follows

$$R_i = \alpha_1 Q_i^{-1} + \alpha_2 Q_i^{-2} \tag{6.10}$$

where R_i is the total number of bits used for encoding the current frame i, Q_i is quantization level used for the current frame i, and α_1 and α_2 represent first and second order coefficients. Although this model provides the theoretical foundation for the rate-control scheme, the major drawback is its lack of considering two factors. At first, the R(D) model is not scalable for video contents. Second, the R(D) model does not exclude the bit counts used for coding the overhead including video/frame syntax, motion vectors and shape information.

To enhance the R(D) model with more accuracy, a simple prediction is used to predict those bits using the last coded frame as a reference. These bits used for nontexture information are considered as constant numbers irrespective of this distortion and are excluded from the target bit-rate estimation. To accurately estimate the target bit rate with scalability, the original quadratic R(D) formula is modified by introducing two new parameters: Mean Absolute Difference (MAD) and nontexture overhead (H).

$$\frac{R_i - H_i}{M_i} = \alpha_1 Q_i^{-1} + \alpha_2 Q_i^{-2} \tag{6.11}$$

where R_i, Q_i and α_1, α_2 are previously defined, H denotes the bits used for header, motion vectors and shape information and M_i represents MAD, computed using motion-compensated residual for the luminance component (that is, Y component).

To solve the target bit rate, it is assumed the video is encoded first as an I-frame and subsequently as P-frames. The scheme has been extended to variable GOP structure and B-frames. Let T_i be the bit budget used for the first I-frame, N_p the number of P-frames, H_p the bit budget used for nontexture information and T_p the bit budget used for all P-frames. Then, the total bit budget is

$$R = T_i + N_p T_p$$

$$\frac{T_p - H_p}{M_p} = \alpha_1 Q_p^{-1} + \alpha_2 Q_p^{-2} \tag{6.12}$$

Then, the T_p and Q_p can be obtained based on the technique described in Chang and Zhang [6.63].

Let

$$X_{n\times 2} = \left[1, 1/Q_p(i)\right], \quad \text{and} \quad Y_{n\times 1} = \left[Q_p(i)\, T_p(i)\right], \tag{6.13}$$

where i = 1,2,...,n, and n is the number of selected data samples. Then

$$\begin{bmatrix}\alpha_1 \\ \alpha_2\end{bmatrix} = \left(X^T X\right)^{-1} X^T Y \tag{6.14}$$

Based on these two model parameters α_1 and α_2, the quantization level Q_p and target bit rate T_p can be computed before encoding the next frame.

In the initialization stage, the major tasks that the encoder has to complete with respect to the rate control include the following:

- Initializing the buffer size based on the latency requirement
- Subtracting the bit counts of the first frame from the total bit counts
- Initializing the buffer fullness in the middle level

Without loss of generality, we assume that the video sequence is encoded first as an I-frame and subsequently as P-frames. In this stage, the encoder encodes the first I-frame using an initial Q_p value specified as an input parameter.

In the pre-encoded stage, the tasks of the rate control scheme include target bit estimation, further adjustment of the target bit based on the buffer status for each VO and QP calculation. The target bit count is estimated in the following phases: frame-level bit rate; object level, if desired and MB level bit-rate estimation, if desired.

In the encoding stage, the major tasks that the encoder has to complete include the following: encoding the video frame (object), recording all actual bit rates and activating the MB layer rate control if desired. In the encoding stage, if either the frame or object-level rate control is activated, the encoder compresses each video frame or VO using Q_p as computed in the pre-encoding stage. However, some low-delay applications may require strict buffer regulations, less accumulated delay and better spatial perceptual quality. An MB level rate control is necessary. However, an MB level rate control is costly at low rates because there is additional overhead if the QP is changed frequently within a frame.

In the postencoding stages, the encoder needs to complete the following tasks: updating the corresponding quadratic R(D) model for the entire frame or an individual VO, performing the shape-threshold control to balance the bit usage between the shape information and texture information and performing the frame-skipping control to prevent the potential buffer overflow and/or underflow.

6.2.4 Streaming Video across the Internet

Real-time transport of live video or stored video is the predominant part of real-time multimedia. On the other hand, video streaming refers to real-time transmission of stored video. There are two modes for transmission of stored video across the Internet: the download mode and the streaming mode. In the download mode, a user downloads the entire video file and then plays back the video file. However, full file transfer in the download mode usually suffers long and perhaps unacceptable transfer time. In the streaming mode, the video content need not be downloaded in full, but is being played out while parts of the content are being received and decoded. Due to its real-time nature, video streaming has bandwidth, delay and loss requirements. Designing mechanisms and protocols for streaming video pose

many challenges. Streaming video has six key areas: video compression, application-layer QoS control, continous media distribution services, streaming servers, media synchronization mechanisms and protocols for streaming media. Each of the six areas is a basic building block with which an architecture for streaming video can be built. The relations among these basic building blocks are illustrated in Figure 6.18 [6.64, 6.65]. Raw video and audio data are precompressed by video compression and audio compression algorithms and saved in storage devices. Upon the client's request, a streaming server retrieves compressed audio/video data from storage devices. Then, the application layer QoS control module adapts the audio-video bit streams according to the network states and QoS requirements. After the adaptation, the transport protocols packetize the compressed bit streams and send the audio-video packets to the Internet. Packets may be dropped as they experience excessive delays inside the Internet due to congestion. To improve the quality of audio-video transmission, continuous media distribution services are developed for the Internet. For packets that are successfully delivered to the receiver, they first pass through the transport layers and then are processed by the application layer before being decoded at the audio-video decoder. To achieve synchronization between video and audio presentations, media synchronization mechanisms are required. As it can be seen, these areas are closely related, and they are coherent constituents of the video streaming architecture.

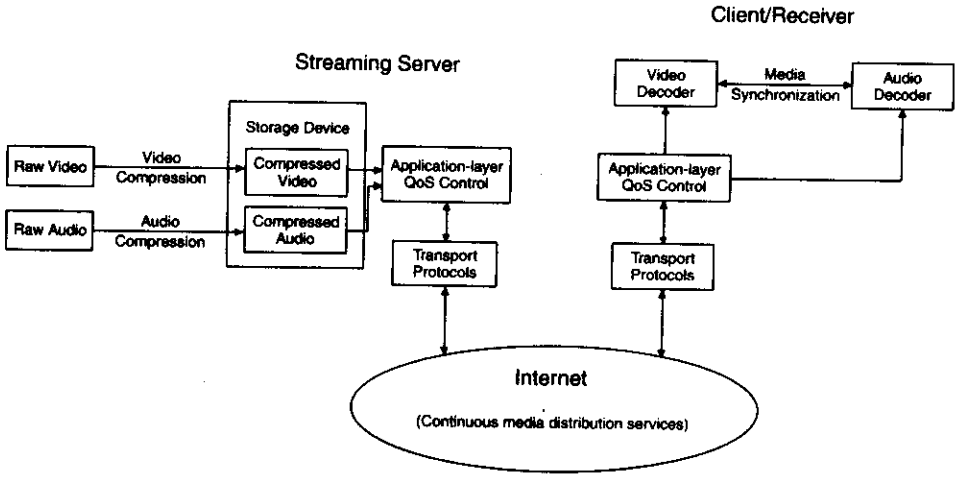

Figure 6.18 Video-streaming architecture [6.65]. ©2001 IEEE.

Video Compression

Video streaming is an important component of many Internet multimedia applications, such as distance learning, digital libraries, home shopping and video on demand. The best-effort nature of the current Internet poses many challenges to the design of streaming-video systems. Our objective is to give the reader a perspective on the range of options available and the associated

trade-off among performance, functionality and complexity of existing approaches. To provide insights on design of streaming-video systems, we begin with video compression. Raw video must be compressed before transmission to achieve efficiency. Video-compression schemes can be classified into two categories: scalable and nonscalable video coding. Scalable video is capable of gracefully coping with the bandwidth fluctuations on the Internet [6.66]. Because raw video consumes a large amount of bandwidth, compression is usually employed to achieve a transmission efficiency. The primary objectives of ongoing research on scalable video coding are to achieve high compression efficiency, high flexibility (bandwidth scalability) and/or low complexity. Due to the conflicting nature of efficiency, flexibility and complexity, each scalable video-coding scheme seeks a trade-off among these three factors. Designers of video-streaming service need to choose an appropriate scalable video-coding scheme that meets the target efficiency and flexibility at an affordable cost/complexity. For simplicity, we only show the encoder and decoder in intramode and only use DCT. Intramode coding refers to coding a video unit (for example, an MB) without reference to previously coded data. For wavelet-based scalable video coding, we recommend references [6.67, 6.68, 6.69]. A nonscalable video encoder and decoder are presented in Figure 6.19.

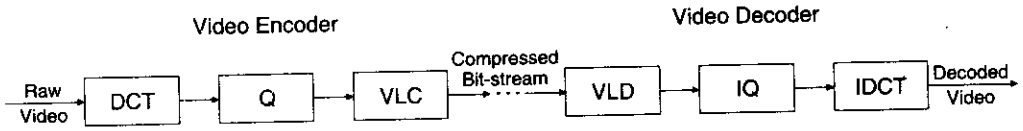

Figure 6.19 Nonscalable video encoder and video decoder.

In contrast, a scalable video encoder (Figure 6.20) compresses a raw video sequence into multiple substreams. One of the compressed substreams is the base substream, which can be independently decoded and can provide coarse visual quality. Other compressed substreams are enhanced substreams, which can only be decoded together with the base substream and which can provide better visual quality. The complete bit stream (that is, combination of all the substreams) provides the highest quality.

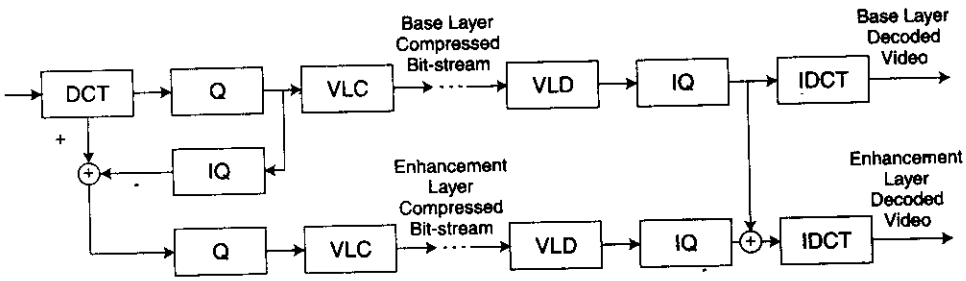

Figure 6.20 SNR scalable video encoder and video decoder.

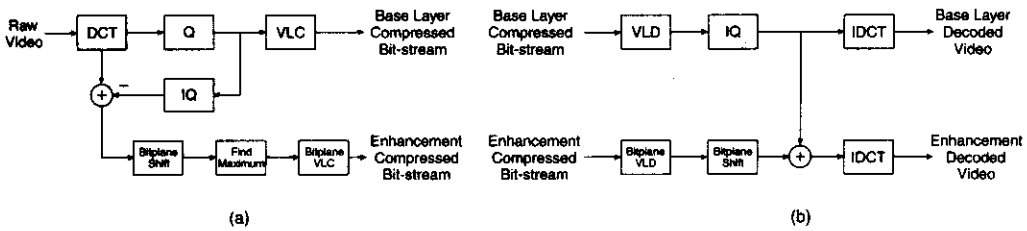

Figure 6.21 FGS: a) encoder and b) decoder [6.65]. ©2001 IEEE.

To provide more flexibility in meeting different demands of streaming (for example, different access link bandwidths and different latency requirements), a scalable coding mechanism called Fine Granularity Scalability (FGS) was proposed to MPEG-4 [6.70, 6.71, 6.72]. As shown in Figure 6.21 an FGS encoder compresses a raw video sequence into two substreams: a base layer bit stream and an enhancement layer bit stream. The FGS encoder uses bit-plane coding to represent the enhancement stream. Bit-plane coding uses embedded representations [6.73]. With bit-plane coding, an FGS encoder is capable of achieving combination rate control for the enhancement stream. This is because the enhancement bit stream can be truncated anywhere to achieve the target bit rate. For example, a DCT coefficient can be represented by 7 bits (that is, its value ranges from 0 to 127). There are 64 DCT coefficients in an 8x8 block. Each DCT coefficient has a Most Significant Bit (MSB), and all the MSBs from the 64 DCT coefficients form bit plane 0. Similarly, all the second-most significant bits form bit plane 1. Bit planes of enhancement DCT coefficients are shown in Figure 6.22.

A version of FGS is Progressive FGS (PFGS). It shares the good features of FGS, such as fine granularity bit-rate scalability and error resilience. Unlike FGS, which only has two layers, PFGS can have more than two layers. The essential difference between FGS and PFGS is that FGS only uses the base layer as a reference for motion prediction while PFGS uses multiple layers as references to reduce the prediction error, resulting in higher coding efficiency [6.74].

Requirements Imposed by Streaming Applications

These requirements are bandwidth, delay, loss, VCR (video-cassette-recorder) functions and decoding complexity.

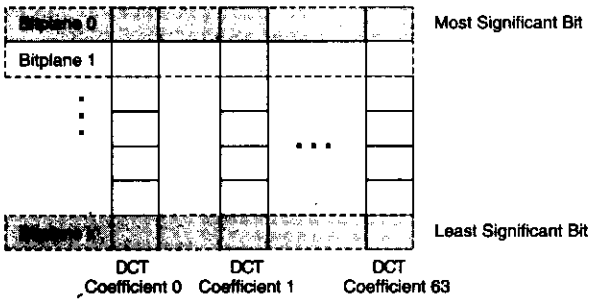

Figure 6.22 Bit planes of enhancement DCT coefficients [6.65]. ©2001 IEEE.

To achieve acceptable perceptual quality, a streaming application typically has a minimum bandwidth requirement. However, the current Internet does not provide bandwidth reservation to support this requirement. In addition, it is desirable for video-streaming applications to employ congestion control to avoid congestion, which happens when the network is heavily loaded. For video streaming, congestion control takes the form of rate control, that is, adapting the sending rate to the available bandwidth in the network. Compared with nonscalable video, scalable video is more adaptable to the varying available bandwidth in the network.

Streaming video requires bounded end-to-end delay so that packets can arrive at the receiver in time to be decoded and displayed. If a packet does not arrive on time, the playout process will pause, which is annoying to human eyes. A video packet that arrives beyond its delay bound (playout time) is useless and can be regarded as lost. Because the Internet introduces time-varying delay, a buffer at the receiver is usually introduced before decoding [6.75].

Packet loss is inevitable on the Internet. It can damage pictures, which is displeasing to human eyes. Thus, it is desirable that a video stream be robust to packet loss. Multiple description coding is such a compression technique to deal with packet loss.

Some streaming applications require VCR-like functions, such as stop, pause/resume, fast forward, fast backward and random access. A dual-bit-stream least-cost scheme to provide VCR-like functionality efficiently for MPEG video streaming is proposed in Lin et al. [6.76].

Some devices, such as cellular phones and PDAs, require low power consumption. Therefore, streaming-video applications running on these devices must be simple. In particular, low decoding complexity is desirable.

So far, we have discussed various compression mechanisms and requirements imposed by streaming applications on the video encoder and decoder. Next, we present the applications-layer QoS control mechanisms, which adapt the video bit streams according to the network status and QoS requirements.

Application Layer QoS Control

The objective of application layer QoS control is to avoid congestion and to maximize video quality in the presence of packet loss. The application layer QoS control techniques include congestion control and error control. These techniques are employed by the end systems and do not require any QoS support from the network.

Burst loss and excessive delay have devastating effects on video presentation quality, and they are usually used by network congestion control. Thus, congestion control mechanisms are necessary to help reduce packet loss and delay. For streaming video, congestion control takes the form of rate control [6.77]. There are three kinds of rate control: source-based, receiver-based and hybrid rate control. The source-based rate control is suitable for unicast. The receiver-based and hybrid rate control are suitable for multicast because both can achieve good trade-off between bandwidth efficiency and service flexibility for multicast video.

Under the source-based rate control, the sender is responsible for adapting the video transmission rate. Feedback is employed by source-based control mechanisms. Based on the feedback information about the network, the sender could regulate the rate of the video stream. The

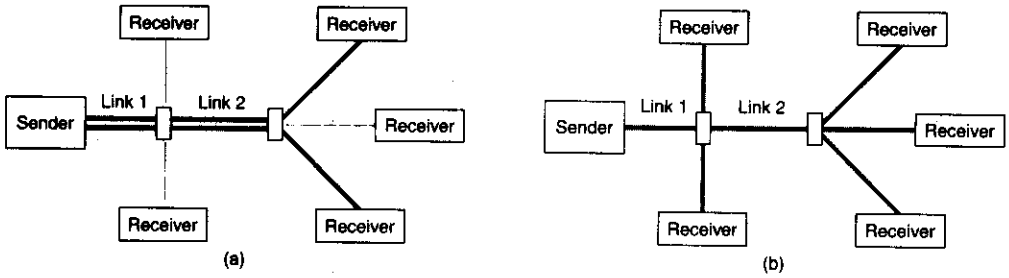

Figure 6.23 (a) Unicast video distribution using multiple point-to-point connections and (b) multicast video distribution using point-to-multipoint transmission [6.65]. ©2001 IEEE.

source-based rate control can be applied to both unicast [6.78] and multicast [6.79]. Unicast video distribution using multiple point-to-point connections as well as multicast video distribution using point-to-multipoint transmission are shown in Figure 6.23. For unicast video, source-based rate control mechanisms follow two approaches: a probe-based and a model-based approach [6.77]. The probe-based approach is based on probing experiments. Specifically, the services probe for the available network bandwidth by adjusting the sending rate in a way that could maintain the packet loss ratio p below a certain threshold p_{th} [6.78]. There are two ways to adjust the sending rate:

- Additive increase and multiplicative decrease [6.78]
- Multiplicative increase and multiplicative decrease [6.80]

The model-based approach is based on a throughput model of a TCP connection. Specifically, the throughput of a TCP connection can be characterized by the following formula:

$$\lambda = \frac{1.22\ MTU}{RTT\ \sqrt{p}} \tag{6.15}$$

where λ is a throughput of a TCP connection, MTU is the packet size used by the connection, RTT is the round-trip time for the connection and p is the packet-loss ratio experienced by the connection. Under the model-based rate control, this expression is used to determine the sending rate of the video stream. Thus, the video connection could avoid congestion in a way similar to that of TCP and it can compete fairly with TCP flows. For this reason, the model-based rate control is called TCP-friendly rate control [6.81]. For multicast under the service-based rate control, the sender uses a single channel to transport video to the receivers. Such multicast is called single-channel multicast. For single-channel multicast, only the probe-based rate control can be employed [6.79]. Single-channel multicast is efficient because all the receivers share one channel. However, single-channel multicast is unable to provide flexible services to meet the different demands from receivers with various access link bandwidths. In contrast, if multicast video was to be delivered through individual unicast streams, the bandwidth efficiency is low but the services could be differentiated because each receiver can negotiate the parameters of the services with the

source. Under the receiver-based rate control, the receivers regulate the receiving rate of video streams by adding/dropping channels while the sender does not participate in rate control [6.77]. Receiver-based rate control is used in multicasting scalable video where there are several layers in the scalable video and each layer corresponds to one channel in the multicast tree.

Similar to the source-based rate control, the existing receiver-based rate-control mechanisms follow two approaches: a probe-based and a model-based approach. The basic probe-based rate control consists of two parts:

- When no congestion is detected, a receiver probes for the available bandwidth by joining a layer/channel, resulting in an increase of its receiving rate. If no congestion is detected after the joining, the joint experiment is successful. Otherwise, the receiver drops the newly added layer.
- When congestion is detected, a receiver drops a layer (that is, leaves a channel), resulting in a reduction of its receiving rate [6.66].

Unlike the probe-based approach that implicitly estimates the available network bandwidth through probing experiments, the model-based approach uses explicit estimation for the available network bandwidth.

Under the hybrid rate control, the receivers regulate the receiving rate of video streams by adding/dropping channels while the sender also adjusts the transmission rate of each channel based on feedback from the receivers [6.82].

An error-control mechanism includes FEC, retransmission, error-resilient encoding and error concealment. There are three kinds of FEC: channel coding, source coding-based FEC and joint source/channel coding. The advantage of all FEC schemes over retransmission-based schemes is reduction in video transmission latency. Source coding-based FEC can achieve lower delay than channel coding, and joint source/channel coding could achieve optimal performance in a rate-distortion sense. The disadvantages of all FEC schemes are increase in transmission rate and inflexibility to varying loss characteristics. Unlike FEC, which add redundancy regardless of correct receipt or loss, a retransmission-based scheme only resends the packets that are lost. Thus, a retransmission-based scheme is adaptive to varying loss characteristics, resulting in efficient use of network resources. The limitation of delay-constrained retransmission-based schemes is that their effectiveness diminishes when the RTT is too large. Currently, an important direction is to combine FEC with retransmission [6.65] . In addition, FEC can be used in layered video multicast so that each client can individually trade off latency for quality based on specific requirements. MDC is a recently proposed mechanism for error-resilient coding. The advantage of MDC is its robustness to loss. The cost of MDC is reduction in compression efficiency. The current research effort gears toward finding a good trade-off between the compression efficiency and the reconstruction quality from one description. Error concealment is performed by the receiver when packet loss occurs and can be used in conjunction with other techniques, for example, congestion control and other error-control mechanisms.

Continuous Media Distribution Services

In order to provide quality multimedia presentations, adequate support from the network is critical. This is because network support can reduce transport delay and packet loss ratio. Streaming video and audio are classified as continuous media because they consist of a sequence of media quanta (such as audio samples or video frames), which convey meaningful information only when presented in time. Built on top of the Internet (IP), continuous media distribution devices are designed with the aim of providing QoS and of achieving efficiency for streaming video/audio across the best-effort Internet. Continuous media distribution services include the following:

- Network filtering
- Application-level multicast
- Content replication.

As a congestion-control technique, network filtering is aimed at maximizing video quality during network congestion. Figure 6.24 illustrates an example of placing filters in the network. The nodes labeled R denote routers that have no knowledge of the format of the media streams and that may randomly discard packets. The filter nodes receive the client's requests and adapt the stream sent by the server accordingly. This solution allows the service provider to place filters on the nodes that connect to network bottlenecks. Furthermore, multiple filters can be placed along the path from a server to a client.

To illustrate the operation of filters, a system model is depicted in Figure 6.25. The model consists of the server, the client, at least one filter and two virtual channels between them. One channel is for control, and the other is for data. The same channels exist between any pair of filters. The control channel is bidirectional, which can be realized by TCP connections. The model allows the client to communicate with only one host (the lost filter), which will either forward the requests or act upon them. The operations of a filter on the data plane include receiving video stream from a server or previous filter and sending video to the client or the next filter at the target rate. The operations of a filter on the control plane include receiving requests from the client or the next filter, acting upon requests and forwarding the requests to its previous filter. Typi-

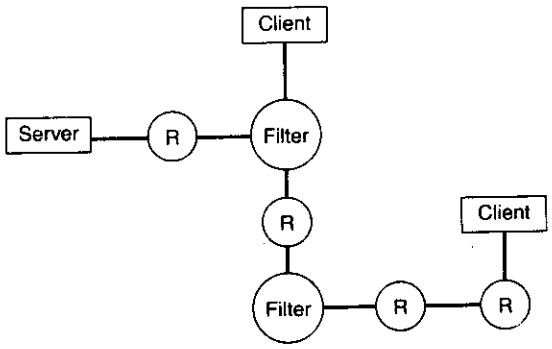

Figure 6.24 Filters placed inside the network [6.65]. ©2001 IEEE.

Figure 6.25 A system model of network filtering [6.65]. ©2001 IEEE.

cally, frame-dropping filters are used as network filters. The receiver can change the bandwidth of the media stream by sending requests to the filter to increase or decrease the frame-dropping rate. The advantages of using frame-dropping filters inside the network include improved video quality and bandwidth efficiency. This is because the filtering can help to save network resources by discarding those frames that are late.

As an extension to the IP layer, IP multicast is capable of providing efficient multipoint packet delivery. The efficiency is achieved by having only one copy of the original IP packet sent by the multicast source and transmitted along any physical path in the IP multicast tree. However, there are many barriers in deploying IP multicast. These problems include scalability, network management, deployment and support for higher layer functionality, for example, error flow and congestion control. The application-level multicast is aimed at building a multicast service on top of the Internet. It enables independent Content Service Providers (CSPs), ISPs or enterprises to build their own Internet multicast networks and to interconnect them into larger, world-wide media multicast networks. The advantage of the application level multicast is that it breaks the barriers such as scalability, network management, and support for congestion control, which have prevented ISPs from establishing "IP multicast" peering arrangements.

An important technique for improving scalability of the media delivery system is content media replication. The content replication takes two forms: caching and mirroring [6.83], which are deployed by publishers, CSPs and ISPs. Both caching and mirroring seek to place content closer to the clients and both share the following advantages:

- Reduced bandwidth consumption on network links
- Reduced load on streaming servers
- Reduced latency for clients
- Increased availability

Mirroring is to place copies of the original multimedia files on other machines scattered around the Internet, that is, the original multimedia files are stored on the main server while copies of the original multimedia files are placed on the duplicate servers. In this way, clients can retrieve multimedia data from the nearest duplicate server, which gives the clients the best performance. On the other hand, caching makes local copies of contents that the client retrieves. Clients in a single organization retrieve all contents from a single local machine, called a cache. The cache retrieves a video file from the original server, storing a copy locally and then passing it on to the client who requests it. If a client asks for a video file that the

cache has already stored, the cache will return the local copy rather than going all the way to the original server where the video file resides.

Streaming Servers

Streaming servers are essential in providing streaming services. To offer quality streaming services, streaming servers are required to process multimedia data under timing constraints in order to prevent artifacts, for example, jerkiness in video motion and pops in audio during playback on the clients. In addition, streaming servers also need to support VCR-like control operations, such as stop, pause/resume, fast forward and fast backward. Furthermore, streaming servers have to retrieve media components in a synchronous fashion. A streaming server consists of three subsystems: communicator, operating system and storage system. A communicator involves the application layer and transport protocols implemented on the server. Through a communicator, the clients can communicate with a server and can retrieve multimedia contents in a continuous and synchronous manner. Different from traditional operating systems, an operating system for streaming services needs to satisfy real-time requirements for streaming applications. A storage system for streaming services has to support continuous media storage and retrieval. In what follows, we discuss synchronization mechanisms for streaming media.

Media Synchronization

A major feature that distinguishes multimedia applications from other traditional data applications is the integration of various media streams that must be presented in a synchronized fashion. For example, in distance learning, the presentation of slides should be synchronized with the commenting audio stream as shown in Figure 6.26. Otherwise, the current slide being displayed on the screen may not correspond to the lecturer's explanation heard by the listeners. With media synchronization, the application at the receiver side can present the media in the same way as they were originally captured. Media synchronization refers to maintaining the temporal relationships within one data stream and between various media streams. There are three levels of synchronization: intrastream, interstream and interobject synchronization. The three levels of synchronization correspond to three semantic layers of multimedia data [6.84].

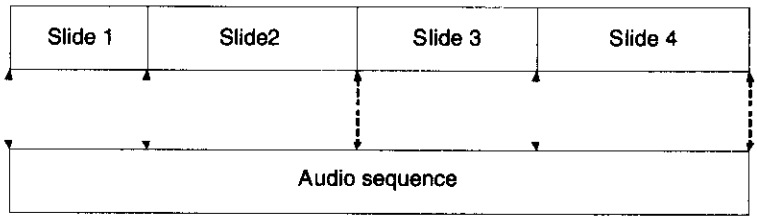

Figure 6.26 Synchronization between the slides and the commenting audio stream [6.65]. ©2001 IEEE.

The lowest layer of continuous media or time-dependent data (such as video or audio) is the media layer. The unit of media layer is a logical data unit such as an audio-video frame, which adheres to strict temporal constraints to ensure acceptable user perception at playback. Synchronization at this layer is referred to as intrastream synchronization, which maintains the continuity of logical data units. Without intrastream synchronization, the presentation of the stream may be interrupted by pauses or gaps.

The second layer of time-dependent data is the stream layer. The unit of the stream layer is a whole stream. Synchronization at this layer is referred to as interstream synchronization, which maintains temporal relationships among different continuous media. Without interstream synchronization, the skew between the streams may become intolerable. For example, users could be annoyed if they notice that the movements of the lips of a speaker do not correspond to the presented audio.

The highest layer of a multimedia document is the object layer, which integrates streams and time-dependent data, such as text and still images. Synchronization of this layer is referred to as interobject synchronization. The objective of interobject synchronization is to start and stop the presentation of the time-independent data within a tolerable time interval, if some previously defined points of the presentation of a time-dependent media object are reached. Without interobject synchronization, the audience of a slide show could be annoyed if the audio is commenting on one slide while another slide is being presented. For more information on media synchronization, we recommend [6.84] and [6.85].

Protocols for Streaming Video

Several protocols have been standardized for communication between clients and streaming servers. According to their functionalities, the protocols directly related to Internet streaming video can be classified into the following three categories: network layer protocol, transport protocol and session control protocol. Network layer protocol provides basic network service support such as network addressing. The IP serves as the network layer protocol for Internet video streaming.

Transport protocol provides end-to-end network transport functions for streaming applications. Transport protocols include UDP, TCP, RTP, and RTCP. UDP and TCP are lower-layer transport protocols, and RTP and RTCP are upper-layer transport protocols that are implemented on top of UDP and TCP.

Session control protocol defines the messages and procedures to control the delivery of the multimedia data during an established session. The RTSP [6.86] and the SIP [6.87] are such session control protocols.

To illustrate the relationship among the three types of protocols, protocol stacks for media streaming are presented in Figure 6.27. For the data plane at the sending side, the compressed audio-video data is retrieved and packetized at the RTP layer [5.21]. The RTP packetized streams provide timing and synchronization information, as well as sequence numbers [6.88]. The RTP packetized streams are then passed to the UDP/TCP layer and the IP layer. The resulting IP packets are transported across the Internet. At the receiver side, the media streams are pro-

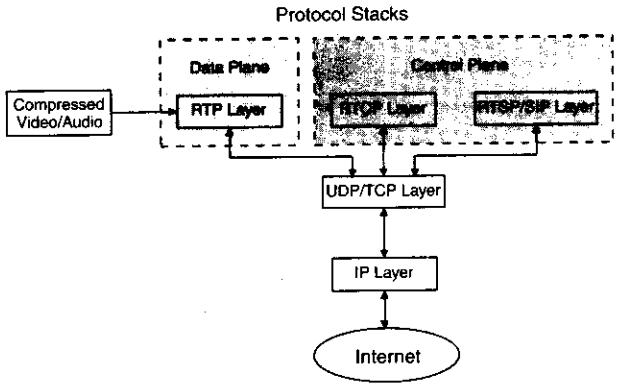

Figure 6.27 Protocol stacks for media streaming [6.65]. ©2001 IEEE.

cessed in the reversed manner before their presentations. For the control plane, RTCP packets and RTSP packets are multiplexed at the UDP/TCP layer and are moved to the IP layer for transmission across the Internet.

Basic building blocks for a streaming video architecture tie together a broad range of technologies from signal processing, networking and server design. A full understanding of the whole architecture is essential for developing the particular signal-processing techniques suitable for streaming video. Furthermore, an in-depth knowledge of both signal-processing and networking technologies helps to make effective design and use of application-layer QoS control, continuous media distribution services and protocols. Finally, a clear understanding of the overall architecture is instrumental in the design of efficient, scalable and/or fault-tolerant streaming servers.

6.3 Multimedia Transport Across ATM Networks

Multimedia itself denotes the integrated manipulation of at least some information represented as continuous media data, as well as some information encoded as discrete media data (text and graphics). Multimedia communication deals with the transfer, protocols, services and mechanisms of discrete media data and continuous media data (audio or video) on and across digital networks. Such communication requires that all involved components be capable of handling a well-defined QoS. The most important QoS parameters are required capacities of the involved resources and compliance to end-to-end delay and jitter as timing restrictions and restriction of the loss characteristics. A protocol designed to reserve capacity for continuous media data, transmitted in conjunction with the discrete media data over, for example, an ATM/LAN, is certainly a multimedia communication issue [6.89]. The success of ATM for multimedia communications depends on the successful standardization of its signaling mechanisms, its ability to attract the development of the native ATM applications and the integration of the ATM with other communications systems. The integration of ATM into the Internet world is under investigation. If there will be ATM applications such as video on demand, there is also the need for a side-by-side integration of ATM and Internet protocols. The success of wireless ATM (WATM) relies on the suc-

cess of ATM/BISDN in wired networks. When ATM networks become a standard in the wired area, the success of WATM will be realized.

6.3.1 Multiplexing in ATM Networks

In order to transfer the information to the destination, the network performs the generic functions of multiplexing and routing. The routing functions, in order to provide connectivity, are not dependent on the information type in the transfers. On the other hand, multiplexing is highly dependent on the requirements by the information type and application context because multiplexing determines much of the transfer quality on the network. The optimization criteria for the transfer are to minimize the queuing and to maximize the utilization. A joint optimization is possible if the multiplexed streams are shaped to minimize the temporal variability.

Asynchronous time division multiplexing enables statistical multiplexing, but does not mandate it. Statistical multiplexing has been successfully used for data communication for three decades and more recently also in radio networks by means of spread spectrum techniques. The network provides fair access to the transmission capacity and routing. The end equipment is responsible for the quality of the transmission by means of retransmission and forward error correction. The choice of multiplexing mode for asynchronous transfers depends on several issues, as illustrated in Figure 6.28 [6.90]. Here, the link has capacity C_{link}, and the source has peak rate $\hat{R}$, mean rate $\overline{R}$ and maximum burst length $\hat{b}$. The required quality is denoted by Q.

As for general service classes, we can define three classes:

- Deterministic multiplexing with fixed quality guarantees
- Statistical multiplexing with probabilistic quality guarantees
- Statistical multiplexing without quality guarantees

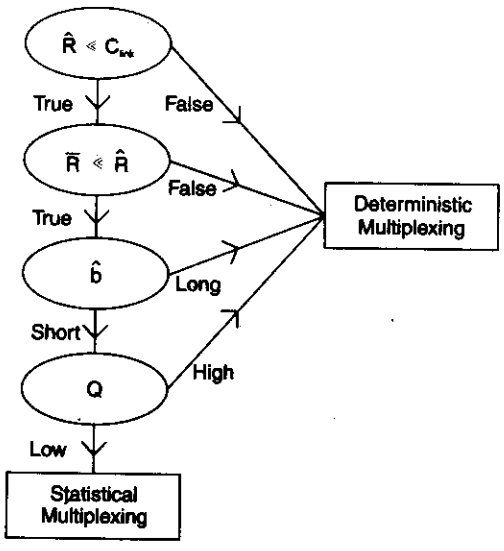

Figure 6.28 Choice of multiplexing mode [6.90]. ©2000 Prentice Hall.

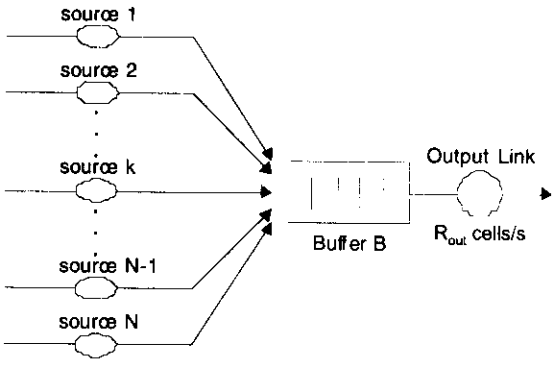

Figure 6.29 Multiplexer model [6.90]. ©2000 Prentice Hall.

The multiplexer can be modeled as a finite capacity queuing system with buffer size B (in ATM cells) and one server with fixed output rate R_{out}. The input of the multiplexer consists of various video sources. The model of the multiplexer is illustrated in Figure 6.29. There are different interleaving schemes for placing data from various sources into the common buffer at two time scales: frame time and cell time [6.91, 6.92, 6.93, 6.94].

In a frame interleaving scheme, the information of each video source is multiplexed in the unit of a frame. It is assumed that each video source has a buffer that can store one frame of information, and all sources are synchronized in the frame boundary. At each frame time, the multiplexer scans each source and puts the information into the common buffer.

In a cell interleaving scheme, the multiplexing process is performed in the unit of a cell. It is assumed that all sources are transmitted at their peak rates from the sources to the multiplexer and that the cells in each frame are uniformly spaced. Each video source is synchronized in frame. An example of a frame-based and cell based interleaving scheme is given in Figure 6.30.

The queuing model of the two-layer coding multiplexer is shown in Figure 6.31. The multiplexing queue is managed by a push-out strategy that allows the buffer to be fully shared by both traffic layers. Cells at the secondary layer are lost if the number of cells from both the primary and secondary layers in the buffer are greater than the buffer size. The cells at the primary layer are lost when the total number of cells from the primary layer is greater than the buffer size [6.95]. We can assume that the cell spacing is uniformly distributed across a frame interval by means of a smoothing scheme for a video source [6.96]. The multiplexer places the incoming cells in a common buffer with capacity B and then transmits them across a 155.5 Mb/s channel. When a large number of video sources are multiplexed, a Poisson arrival process can be assumed.

6.3.2 Video Delay in ATM Networks

Information is delayed in ATM networks, and because we consider asynchronous transfers, these delays will not be constant, not even for deterministic multiplexing. This delay has to be considered end-to-end because delay limits are posed by the application. The video signal is

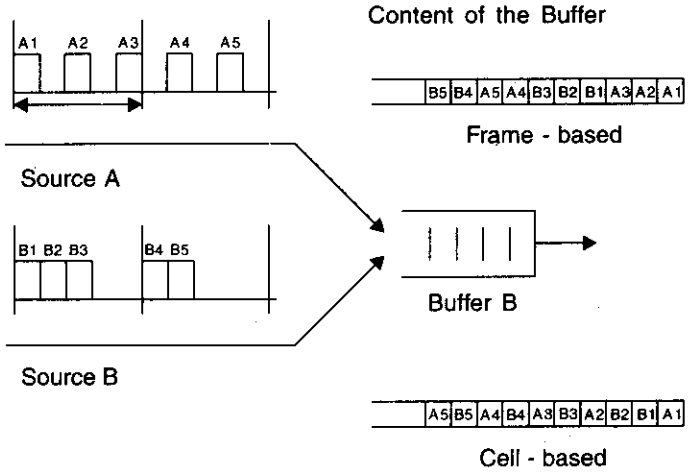

Figure 6.30 Example of multiplexing: frame-based and cell-based interleaving scheme [6.90]. ©2000 Prentice Hall.

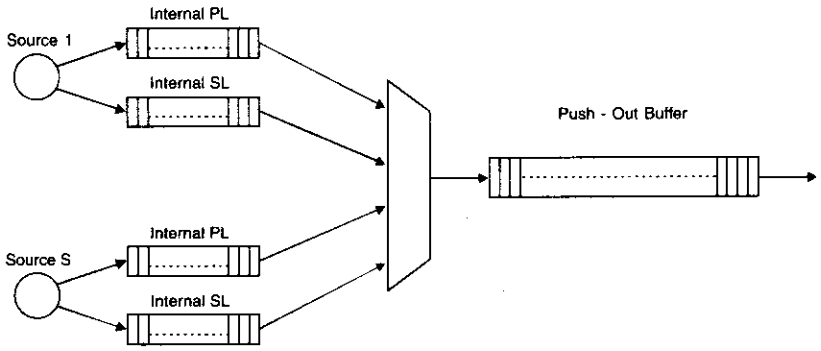

Figure 6.31 ATM video multiplexer for two-layer video streams [6.90]. ©2000 Prentice Hall.

delayed as protocol functions are executed and when the signal is transmitted across the network. The following instances may cause the end-to-end delays:

- Acquisition and display of the video
- Encoding, rate-control and decoding
- Segmentation and reassembly
- Protocol processing
- Wave propagation and transmission
- Queuing

The acquisition is the time that it takes to capture a field or a frame depending on scanning, to digitize it and to perform color and scanning (interlaced to progressive) conversions. The reciprocal functions are performed before display. If there is no scanning conversion, the delay can be of the order of a single pixel instance.

The functions closer to the network are the segmentation of service data units or streams into protocol data units and their reassembly. The time to fill a packet or a cell might be excessive at low rates. For example, it takes 125 μs per octet at 64 Kb/s. If the rate is temporarily or constantly low, it may be necessary to enforce a time limit or to send partially filled cells or packets of restricted length. Reassembly delay depends on message length and transfer rate. For instance, there is no delay for unstructured stream-oriented data. There may be restrictions on the MTU to achieve acceptable delay. The MTU is then dependent on the transfer rate, or the minimum acceptable rate is determined by the MTU size.

Protocol processing is a major cause of delay. It includes framing of information, calculation of check sums and address lookup in hosts and switches and routers. In general, protocols should be implemented to reduce maximum delay and to maximize throughput.

Wave propagation is limited by the speed of light. It takes roughly 100 ms to reach half way around the globe (5 μs per kilometer in fiber). The transmission time is the length of the packet or cell on the transmission line. The wave propagation determines when the first bit of a packet reaches the end of a transmission line, and the transmission time specifies how much later the last bit arrives. The transmission delay is reduced by increasing the line capacity and by reducing the number of links per route.

Because the multiplexing is asynchronous, there will be queuing in the network. Queuing delays in the network vary dynamically from cell to cell and packet to packet for a given route. The delay depends on the instantaneous load in each multiplexer, number of multiplexing hops on the route, amount of buffer space per node and whether deterministic or statistical multiplexing is used. The scheduling discipline affects the distribution of the delays.

Cell Delay Variation (CDV) or jitter can have a significant impact on the quality of a video stream. To keep the encoder and decoder in synchronization with each other, the encoder places PCRs periodically in the TS. These are used to adjust the system clock at the decoder as necessary. If there is jitter in the ATM cells, the PCRs will also experience jitter. Jitter in the PCRs will propagate to the system clock, which is used to synchronize the other timing functions of the decoder. This will result in picture quality degradation.

There are two control issues regarding delay:

- The variations must be equalized to maintain the isochronal sample rate.
- The absolute value must be limited for interactive applications.

Equalization at the network interface of the receiver is not sufficient unless all subsegment protocol processing and data transfers within the end system are fully synchronous. This means that equalization will basically always be needed at the application layer. It is, in fact, the most

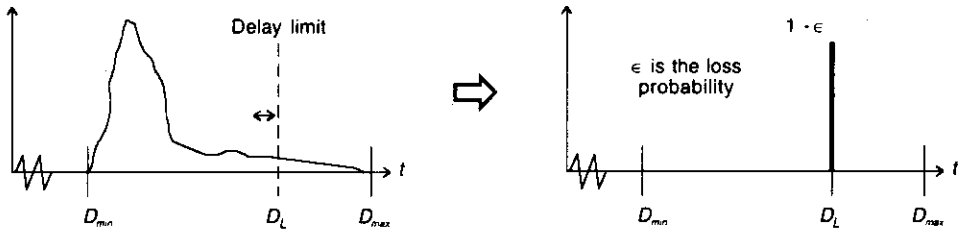

Figure 6.32 Equalization of delay variations.

appropriate location because the bit stream can be synchronized to the display system (the digital-to-analog converter). It should be noted that each stage of equalization introduces more delay.

In Figure 6.32, the delay is equalized by buffering data up to the acceptable limit D_L. Segments that are delayed by more than the limit are treated as if they were lost. Equalization of delay variations is done by buffering data delivered by the network to a predetermined limit before delivery. Late data is discarded (there is no loss if $D_L \geq D_{max}$). The general problem with this approach concerns the choice of D_L to find a proper trade-off between delay and loss and to determine that each pixel has been delayed by D_L when displayed.

A common simplification is to equalize queuing delay at the reassembly point. It is at the adaptation layer in case of ATM and at the transport layer in case of IP. Jitter introduced in the end system is then removed before or after the decoding to obtain signal synchronization.

The delay equalization requires the end system to have a clock that is synchronized in frequency to the sending clock. Usually the clocks at the sender and the receiver will have the same nominal frequency. The jitter is of much larger magnitude than the clock difference. Synchronization could be obtained by locking both clocks to a common reference clock, as carried by the global positioning system and by synchronous digital networks, or by using the network time protocol [6.97]. If a clock reference is not available, then the sender clock has to be estimated from the arriving packet stream. Such a technique uses a phase-locked loop presented in Figure 6.33. The input signal to the loop can be either time stamps carried in the cells or packets or the buffer full level [6.98]. After the clocks are sufficiently synchronized and the data stream is sent completely isochronously, the delay is equalized by simply reading the application frames from the buffer with the same time intervals as sent. Variable rate video complicates the equalization because it is difficult to know how much of the time between arrivals is due to the generating process and how much is due to queuing in the network. Therefore, time stamps in every cell or packet are needed to mark their generating instances.

Signal synchronization is finally obtained after decoding. The frame buffer absorbs much of the delay variations in the decoding, and the residual could be eliminated by repeating or skipping frames to make adjustments to fit the display's clock. If the display allows an external clock, finer adjustments can be made by stretching and shortening the vertical and horizontal tracing times of the cathode ray tube's display. When several video streams emanating from dif-

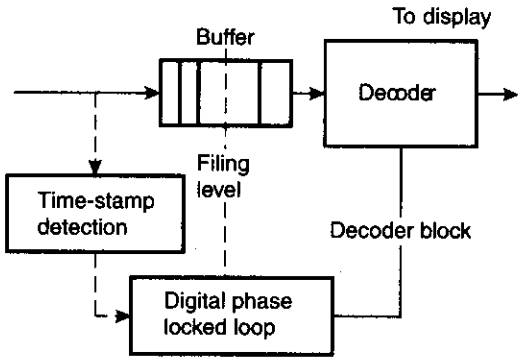

Figure 6.33 Phase-locked loop for the sender-clock frequency estimation [6.90]. ©2000 Prentice Hall.

ferent sources are displayed together, only one of the received signals can be used to synchronize the display system. The other signals must be stretched or contracted to fit that timebase.

6.3.3 Errors and Losses in ATM

The encoding process introduces controlled amounts of distortion in order to compress the signal. The video signal will also be exposed to bit errors induced in the electronics and in the optics. The probability of bit error is low, below 10^{-8}, but not negligible. More troublesome is the information loss in the ATM network when full stretches of the signal are deleted. The causes of loss are transmission burst errors, loss of cells and packets due to multiplexing overload, misrouting due to inaccurate addresses or entries in address tables and delay greater than the acceptable threshold. Undetected loss in a signal can place encoders and decoders out of phase. Burst errors caused by loss of synchronization and by equipment failures have durations of 20 to 40 ms. Their probability of occurrence has been estimated to be below 10^{-7} [6.94]. Loss, especially due to multiplexing overloads, appears to be the most common signal corruption caused by the ATM network.

Error recovery is based on limited error propagation and correction or concealment of the missing portion of the signal. Error propagation is restricted by proper framing of the bit stream so that errors and loss can be detected [6.99].

Generally speaking, in the ATM network, a cell can be lost due to two reasons:

- Channel errors
- Limitations of network capacity and statistical multiplexing

A communication channel is subject to different impairments. If an uncorrectable error occurs in the address field of an ATM cell, the cell will not be delivered to the right destination. This cell is considered to be lost. This is a rare cause of loss in ATM networks.

An ATM network takes advantage of statistical multiplexing, but also takes the risk of simultaneous traffic peaks of multiple users. Although a buffer can be used to absorb the instan-

taneous traffic peak to some extent, there is still a possibility of buffer overflow in case of congestion. In the case of network congestion or buffer overflow, the network congestion control protocol will drop cells. The malfunction or inefficient network management will also cause the cell loss. For example, loss of synchronization and lack of recovery measures in the physical layer would result in a stream of cell losses in the resynchronization/acquisition phase.

In an ATM network, cell discarding can occur on the transmitting side if the number of cells generated are in excess of allocated capacity, or it can occur on the receiving side if a cell has not been received within the delay time of the buffer memory. Cells can be discarded in the ATM network by the congestion control procedure.

If the incoming traffic exceeds allocated capacity and causes the buffer overflow, the sender could be informed by the network traffic control protocol to reduce the traffic flow or to switch to a lower grade service mode by subsampling and interlacing.

If the network becomes congested and the input buffer overflows, it will drop some cells to reduce the traffic and to assume the normal communication phase.

If the error occurs in the cell header, especially in the address field, the cell may be misdelivered or go astray in the network. In the receiver, if a cell is not received within the maximum time out window, the cell is considered to be lost. The loss of a cell leads to the loss of 384 consecutive bits, which may cause a serious degradation in picture quality for VBR compressed video signals. If the cell loss is caused by network congestion, a few consecutive cells, which contain thousands of bits of information, may be lost. Furthermore, the cell loss may affect the subsequent frames if an interframe coding scheme is employed. Therefore, cell loss is a major problem encountered in VBR coding in the ATM environment. A cell loss may cause the loss of code synchronization. Because a variable number of data is packed into a cell, there is no way of knowing how much information is lost when a cell loss occurs unless some side information is available. Cell loss can occur unpredictably in ATM networks. It is assumed to be random with the probability of cell loss depending only on whether a previous cell of the same priority was lost.

Asynchronously multiplexed networks, such as those based on ATM, have cells and packets as multiplexing units that are shorter than a full cell (session). The multiplexing unit in traditional Time Division Multiplexing (TDM) networks is a call (a session). Network framing means that appropriate control information is added to each multiplexing unit. An example of application framing is the MPEG slice layer that packs bits together for 16 consecutive lines. The purpose of the network framing is to detect and to possibly correct lost and corrupted multiplexing units. Errors and loss handling are shown in Figure 6.34 [6.15]. Errors may be detected by a CRC of sufficient length [6.100]. Loss is detected by means of sequence members, which turn it into erasures (known location and unknown values).

It is important that the sequence number is based on the number of transferred data octets. Knowing that a cell or a packet has been lost does not tell how much data it contained.

Errors and loss can be identified by a CRC on the application frame after reassembly. It is important that frame length is known a priori because the length of a faulty frame cannot be ascertained. The failed CRC could be caused by a bit error, which would not be affected by its length or by a lost packet or cell.

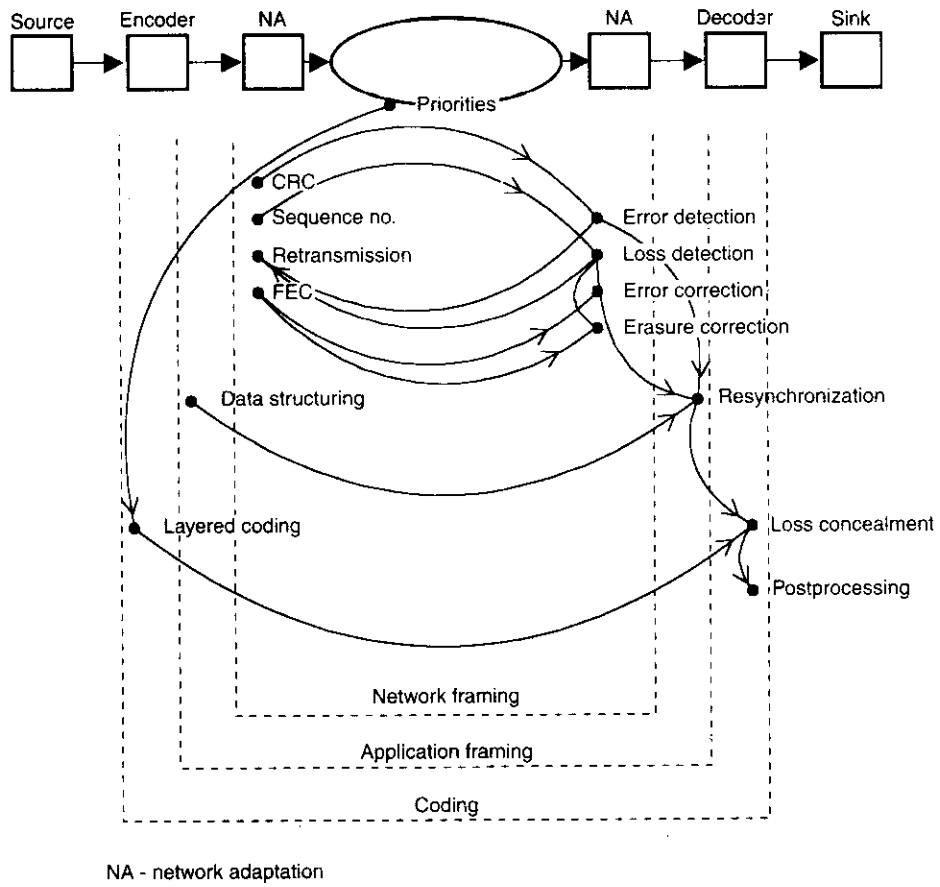

Figure 6.34 Errors and loss handling [6.90]. ©2000 Prentice Hall.

A lost or corrupted network frame would, in case of regular data communication, be retransmitted. There are complications with the use of retransmissions for video. First, the delay requirements might not allow it because it adds at least another round-trip delay that is likely to violate end-to-end delay requirements for conversational services. Second, the jitter introduced is much higher than that induced by queuing. Delay equalization is thus further complicated. Even if this would be acceptable, the continuously arriving datastream must be buffered until the missing frame eventually is received.

There are several reasons to be cautious of FEC of cell and packet loss. First, it adds a fairly complex function to the system, which will be reflected in its cost. Second, the interleaving adds delay. Third, loss caused by multiplexing overload is likely to be correlated because the overload is caused by traffic bursts and more loss may occur than what the code can correct. If an interleaving matrix cannot be corrected, then the full matrix is useless, and the loss situation is in fact made worse.

The interleaving matrix should, of course, be made to cope with burst losses but, again, it increases the delay. Fourth, the coding adds overhead.

6.3.4 MPEG Video Error Concealment

To transmit video traffic effectively across ATM networks, we need to study the issues involved in packetizing encoded video sequences. In particular, it is important to study the effect of ATM cell loss and to develop postprocessing techniques that can be used for error concealment. Error-concealment approaches by Wang [6.101] have assumed that both encoding and decoding occur simultaneously with the decoder communicating to the encoder the location of damaged picture blocks. Many of these techniques are not realistic for real-time applications because they require retransmission of ATM cells. Prioritization approaches to ATM cell-loss concealment have been proposed in several records [6.102 through 6.110]. Figure 6.35 shows a block diagram of the packing/error-concealment scheme using ATM. The cell depacketization operation also provides information as to which macroblocks are missing [6.111].

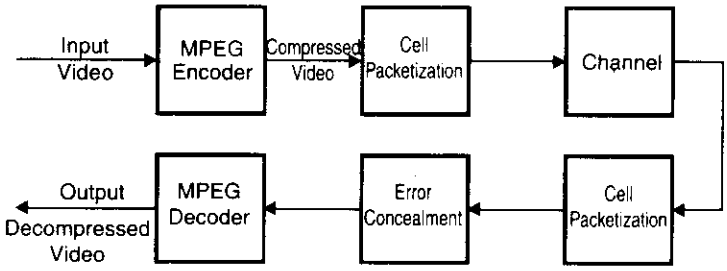

Figure 6.35 Block diagram of the packing/error concealment scheme [6.90]. ©2000 Prentice Hall.

This information is passed to the error concealment algorithm, which attempts to conceal the missing blocks. The goal of video error concealment is to estimate missing MBs in the MPEG data that were caused by dropped ATM cells. The use of spatial, temporal and picture quality concepts are exploited.

Two error-recovery approaches for MPEG encoded video across ATM networks are described in Salama et al. [6.111]. The first approach aims at reconstructing each lost pixel by spatial interpolation from the nearest undamaged pixels. The second approach recovers lost MBs by minimizing intersample variations within each block and across its boundaries.

6.3.5 Loss Concealment

A loss is detected either by means of the network or application-framing information. The corrupted application frame can be considered useless. The application framing should contain sufficient information to allow the next correctly received segment to be decoded. This means that the location within the picture of the information in the segment must be known and that there

cannot be any coding dependencies between the information in the segments. The latter condition implies that there cannot be any prediction dependencies across segment boundaries and that variable-length code words are not split by segment boundaries [6.112].

The decoded picture will contain an empty area that corresponds to the lost information. This area can be concealed by using surrounding pixels in time and space [6.102, 6.113]. For instance, the corresponding area in the previous frame can be used [6.112]. It might be best to repeat the full frame if the corruption is severe. When the coded motion vectors are correctly received, they can be used to find the most appropriate replacement in the previous frame. The prediction error is the only remaining error in the area.

The loss concealment can be improved by thoughtful packing of the information, such as separate transfers of motion vectors and prediction errors. A more general framework is often referred to as layered or hierarchical coding.

6.3.6 Video Across WATM Networks

Due to the success of ATM on wired networks, WATM has become the direct result of the ATM anywhere movement. WATM can be viewed as a solution for next-generation personal communication networks or a wireless extension of the BISDN networks. There has been a great deal of interest recently in the area of wireless networking. Issues, such as bit error rates and cell loss rates, are even more important when transmitting video across a wireless network. A very high performance wireless LAN which operates in the 60 GHz millimeter wave band can experience cell loss rates of 10^{-4} to 10^{-2} [6.114]. To provide adequate picture quality to the user, some form of error correction or concealment must be employed. One option is to use the MPEG-2 error-resilience techniques and to modify the MPEG-2 standard slightly when it is used across WATM networks. This technique is known as MB resynchronization [6.114]. In MB resynchronization, the first MB in every ATM cell is coded absolutely rather than differentially. This allows for resynchronization of the video stream much more often than would be possible if resynchronization could only take place at the slice level. It would be relatively simple to incorporate this method with the existing MPEG-2 coding standard by adding an interworking adapter at the boundary between the fixed and wireless networks [6.115]. A second proposal for improving error resilience in wireless networks is to use FEC methods. In addition, improved performance can be achieved by using a two-layer scalable MPEG-2 coding scheme rather than one layer [6.116].

Mobile ATM defines the design functions of control/signaling. In WATM networks, a mobile end-user establishes a VC to communicate with another end-user, either a mobile or ATM end-user. When the mobile end-user moves from one Access Point (AP) to another AP, proper handover is required. To minimize the interruption to cell transport, an efficient switching of the active VCs from the old data path to new data path is needed. Also, the switching should be fast enough to make the new VCs available to the mobile users. During the handover, an old path is released, and a new path is then re-established. In this case, no cell is lost, and cell sequence is preserved. Cell buffering consists of uplink buffering and downlink buffering. If a VC is broken when the mobile user is sending cells to ATM APs (AAPs), uplinking buffering is required. The mobile user will

buffer all the outgoing cells. When the connection is up, it sends out all the buffered cells so that no cells are lost unless the buffer overflows. Downlink buffering is performed by APs to preserve the downlink cells for sudden link interruptions, congestion or retransmissions. It may also occur when handover is executed. When the handover occurs, the current QoS may not be supported by the new data path. In this case, a negotiation is required to set up new QoS, because the mobile user may be in the access range of several APs. Therefore, it will select the one that can provide the best QoS.

When a connection is established between a mobile ATM endpoint and another ATM endpoint, the mobile ATM end point needs to be located. There are two basic location management schemes: the mobile scheme and the location register scheme. In the mobile scheme, when a mobile ATM moves, the reachability update information only propagates to the nodes in a limited region. The switches within the region have the correct reachable information for the mobiles. When a call is originated by switching in this region, it can use the location information to establish the connection directly. If a call is originated by a switch outside this region, a connection is established between this switch and the mobile's home agent, which then forwards the cells to the mobile. This scheme decreases the number of signaling messages during a local handover. In the location register scheme, an explicit search is required prior to the establishment of connections. A hierarchy of location registers, which is limited to a certain level, is used.

6.3.7 Heterogeneous Networking

Heterogeneity in networks comes from many sources. Link capacity may vary by several orders of magnitude: from 64 to 128 Kb/s for ISDN lines, several hundred Kb/s for wireless LANs, 10 to 100 Mb/s for LANs such as Ethernet and more for ATM networks. Protocols across these various networks may differ up to the link layer, but also at the network layer. For example, hosts directly connected to ATM networks may run a native ATM stack, but those connected to the Internet may run the TCP/IP stack. End stations can differ in the processing power, available to consume multimedia information. Some may have hardware video-decoding capability, and others may perform the decoding in software. The speed of the processor or the bus architecture may place a bottleneck on the multimedia consumption rate. We will deal with the network heterogeneity in the context of distributing multimedia information through multicast transmission. Multicasting significantly improves the efficiency of network resource use in situations involving one-to-many or many-to-many communications.

The layered video deals with implementing applications that use layered video coding and multicast transmission to handle heterogeneity caused by differences in network link capacity, processing power or display resolution. In such scenarios, receivers express interest in getting higher resolution data by subscribing to the appropriate multicast transport stream (a multicast address in IP or a multicast virtual circuit in ATM) [6.117]. For example, Figure 6.36 shows a near video-on-demand system, transmitting layered video in three layers. All receivers subscribe to the base layer with a bit rate chosen to match the transmission characteristics of the low-bit-rate wireless network. The first enhancement layer has a bit rate suited for receivers on the

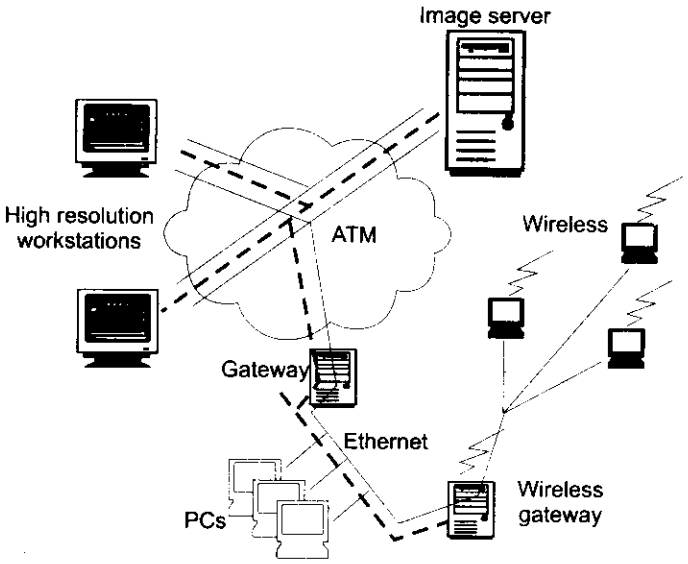

Figure 6.36 Layered networking in heterogeneous networks [6.117]. ©1997 IEEE.

Ethernet. The second enhancement layer is a high-bit-rate layer that the high-powered workstations on the ATM network can receive and decode.

The video source adjusts the bit rates transmitted across each of the layers and dynamically adapts them to the link capacity and receiver processing powers in the current scenario. If the source is transmitting live video, the encoding parameters can be directly manipulated to adjust the bit rates allocated to the different layers. If the information to be transmitted is recorded and pre-encoded video, the source can internally represent the video as a very large number of layers and can map it dynamically to a smaller set of transmission streams based on feedback from the receivers. This mechanism allows the source to match the bit rates of three streams correctly to the characteristics of the wireless, Ethernet and ATM receivers without knowing these characteristics beforehand.

The IP/ATM gateway project addresses the problems of bandwidth and protocol heterogeneity. The gateway is responsible for the following tasks:

- Translating connection set-up messages of the QoS signaling protocols between the two domains. This includes mapping the QoS and traffic parametric between the two domains.
- Forwarding data packets on the IP domain onto appropriate virtual circuits on the ATM side and vice versa. The gateway should perform data forwarding with QoS support, taking QoS parameters into account for scheduling packet transmission.

- Translating session advertisement messages from the two domains to allow a session in one domain to be visible in the other.
- Performing load balancing between multiple gateways connecting the two domains.

6.4 Multimedia Across IP Networks

Multimedia has become a major theme in today's information technology that merges the practices of communications, computing and information processing into an interdisciplinary field. In this Internet era, IP-based data networks have emerged as the most important infrastructure, reaching millions of people anytime and anywhere. They serve as an enabling technology that creates a whole new class of applications to enhance productivity, reduce costs and increase business agility. Anticipating that multimedia across IP will be one of the major driving forces behind the emerging broadband communications of the 21st century, we address the challenges facing the delivery of multimedia applications across IP in a cost-effective, ubiquitous, and quality-guaranteed manner.

6.4.1 Video Transmission Across IP Networks

The problem of sending video across IP has essentially two main components: video data compression and design of communication protocols (Figure 6.37).

Figure 6.37 Structure of a video-streaming system [6.118]. ©2001 IEEE.

One approach consists of designing a low-bit-rate coder, protecting the resulting bit stream with channel codes and using one of the standard Internet transport protocols to transmit the resulting datastream. If the source bit rate is low enough and the channel is not too congested, it is possible to use TCP, in which case no errors occur and therefore there is no need for channel codes. Otherwise, UDP is used with a constant packet injection rate, and low-redundancy channel codes are used to protect against infrequent lost packets. Figure 6.38 illustrates an approach centered around coding problems. In this case, all the intelligence goes into the design of good compression algorithms. It is assumed that the network is a black-box, or a fixed standard pipe, with no differentiation of packets at the socket level or lower.

Figure 6.38 An approach centered around coding problems [6.118]. ©2001 IEEE.

The main drawback of this approach is that it does not deal well with the time-varying nature of the channel. To avoid having to deal with these time variations, the channel is severely underused by using a low-bit-rate coder. This is because, at higher injection rates, fluctuations in the packet-loss rate make it very difficult to guarantee a low probability of decoding error. At higher bit rates, a careful matching of Reed-Solomon (RS) codes to the importance of different portions of an MPEG-2 stream has been proposed [6.118].

Another widely used approach consists of designing new transport protocols, but using either standard video-coding algorithms that generate a fixed syntax for the compressed bit stream or using layered coding techniques. This approach has advantages over the first one discussed previously, the most obvious being that flow control is part of the protocol. In addition, because the bit-stream syntax is known, it is possible to put all the blocks along a given motion trajectory into a single packet so that, if this packet is lost, the entire motion path is lost and therefore error propagation is limited. Also, it is possible to retransmit packets selectively depending on, for example, whether a lost packet contains intracoded blocks or not. These concepts are illustrated in Figure 6.39. In this case, all the intelligence goes into the design of good communication protocols. It is assumed that the coder is typically one of the standards (MPEG-x or H.26x).

Figure 6.39 An approach centered around the design of network protocols [6.118]. ©2001 IEEE.

The main drawback of this approach is it is limited in performance by the nature of the coders used. Video coders based on multiresolution techniques are inherently mismatched to a network that provides no form of packet differentiations. Also, because the modified protocols cannot ensure error-free transmission, when errors do indeed occur, the quality of the decoded signals suffers severely because of lack of robustness in the coders used.

The design of the interface between networks and applications is a problem that has received significant attention. Most of the open literature so far has focused on transmission and traffic regulation across ATM networks, of which perhaps the simplest example is the leaky bucket controller [6.90]. Video across ATM is of interest for two reasons: (a) ATM is one of the candidate transport technologies for future BISDN all the way to the end-user and (b), ATM networks are able to provide QoS guarantees for applications. However, current IP networks are inherently different from ATM networks in that they take a best-effort approach to packet transmission and routing, so no QoS guarantees are provided. As a result, there is no contract to be negotiated between the source and the network. Hence, there are no policing mechanisms applied by the network at its interface with the source.

6.4.2 Traffic Specification for MPEG Video Transmission on the Internet

To promote the evolution of the Internet from a simple data network into a true multiservice network, the IETF Integrated Services WG (ISWG) is defining an Integrated Services Internet, in which traditional best-effort datagram delivery and additional enhanced QoS delivery classes exist [6.119]. Although the IETF has considered various QoS classes, to date, only two of these, Guaranteed Service and Controlled-Load Service, have been formally specified. Guaranteed Service provides an ensured level of bandwidth, a firm end-to-end delay bound and no queuing loss for conforming packets in a data flow. Controlled-Load Service provides a service equivalent to the best-effort delivery on a highly loaded network. Therefore, Guaranteed Service is intended for real-time traffic, and Controlled-Load Service is intended for classes of applications, like adaptive real-time applications, that can tolerate a certain amount of loss and delay, provided that it is kept to a reasonable level.

In order for the new Internet to allow applications to request network packet delivery characteristics according to their needs, sources are expended to declare the offered traffic characteristics. Tspec and admission control rules have to be applied to ensure that requests are accepted only if sufficient network resources are available. Moreover, service-specific policing actions have to be employed within the network to ensure that nonconforming data flows. Policing at the network access point is performed through a token bucket device; packets revealed as nonconforming are marked and forwarded as best-effort traffic.

Traffic specification is a reference point allowing the source and the network to pursue separately two targets:

- To provide the agreed Tspec (source)
- To allocate source requests and to police source traffic (network)

In this context, it is necessary to provide applications with the capability of calculating the Tspec parameters to be declared to the network on the basis of both a limited set of parameters statistically characterizing the data source and the performance of the smoother used in the source to reshape its traffic according to the declared Tspec. The ISWG defined the following Tspec parameters [6.119]:

- Peak rate measured in bytes of IP packets per second, specifying the maximum rate at which the service can inject bursts of traffic into the network.
- Token bucket depth, measured in bytes.
- Bucket rate of the token buckets, measured in bytes of IP packets per second.
- Minimum policed unit (m), measured in bytes, specifying the minimum size of the network packets. All packets of a size less than m will be treated by the policer as being of size m.

- Maximum IP packet size, measured in bytes, specifying the maximum size for a packet that will confirm to the Tspec.

The video source can be modeled as a switched-batch Bernoulli Process (SBBP) taking into account both intra- and inter-GOP correlations [6.120]. Then, a discrete time-queuing system can be used to model video traffic smoothing of the source. After evaluating the loss probability and the average delay suffered in the smoother device, we model the traffic at the output of the smoother, that is, the traffic actually sent across the network. Finally, we model the token bucket at the access point of the network to calculate the marking probability for the packet that does not comply with the specifications.

We can use this paradigm as a tool with the following objectives:

- To design the buffer size of the video server smoother
- To calculate Tspec parameters that are sufficient to guarantee a) both the upper bound for the loss probability and the average delay suffered in the smoother and b) the upper bound for the marking probability in the token bucket at the network access point

6.4.3 Bandwidth Allocation Mechanism

Multimedia applications require the transmission of real-time streams across a network. Pay-per-view movies, distance learning and digital libraries are examples of multimedia applications that require the transmission of real-time streams across a network. Such streams (such as video) can exhibit significant bit rate variability, depending on the encoding system used, and can require high network performance [6.121]. Moreover, these streams require performance guarantees from the network, such as guaranteed bandwidth and loss rate. This poses significant problems when such streams are delivered across the Internet. In fact the real-time network applications that currently run across the Internet achieve a QoS that is far from what is desired. To solve these problems, a small set of differential services has been recently introduced. Among these, Premium Service is suitable for transmitting real-time stored stream (full knowledge of the stream characteristics) [6.122]. It uses a Bandwidth Allocation Mechanisms (BAM) based on the stream peak rate. Due to the variable bandwidth requirement, the peak rate BAM can waste large amounts of bandwidth. One possible approach to reduce bandwidth requirements is to reduce the video VBR. However, even using smoothing techniques, the variability is still present, and, hence, the BAM can still waste a large amount of bandwidth [6.123, 6.124]. Bit-rate variability can also be reduced by modifying the video QoS, but, in this case, the client must settle for a lower QoS [6.122]. Another approach is to allocate the bandwidth in a dynamic way instead of through a fixed bandwidth channel. For instance, one report [6.125] suggests using renegotiation mechanisms to avoid bandwidth waste. Although bandwidth effective, this technique may, at some point in time, require additional bandwidth while transmitting a video. This raises a potential problem because the required additional bandwidth may not be available, leading this technique not to provide the needed bandwidth guarantees. From these consider-

ations, a BAM should not modify while transmitting a video stream. For these reasons, a peak rate BAM is used. A new BAM that uses less bandwidth than the peak rate BAM, while providing the same service, was proposed [6.122]. This BAM does not affect the real-time stream QoS and does not require any modification to the Premium Service Architecture.

To avoid bandwidth waste, the proposed BAM uses dynamic bandwidth allocation and never asks for additional bandwidth. This substantially differs from other dynamic BAMs. For instance, the regeneration mechanisms described in Feng and Rexford [6.126] may make requests for additional bandwidth. They cannot guarantee that these requests will be satisfied by the network. Conversely, the BAM described in Furini and Towsley [6.122] provides bandwidth guarantees as well as the peak rate BAM while using less bandwidth. This is achieved by allocating the peak bandwidth to the premium channel, but progressively reducing this allocation results in the decrease of the peak rate of the remaining stream. This is possible because the streams with known characteristics are considered. Consequently, there is no need to ask for additional bandwidth while transmitting a stream. Hence, this BAM provides the same guarantees as the peak rate BAM while using less bandwidth.

In order to describe BAM, a sender that provides the service and a receiver that desires the service were considered [6.122]. The receiver requests a video, composed of N frames from the sender. Without loss of generality, a discrete time model, where one time unit corresponds to the time between successive frames, was assumed. For a 24-fps full-motion video, the duration of a frame is 1/24 of a second. Denoted by a(i), the amount of data is sent at time i, i=1,2,...,N. We can introduce the bandwidth function, which will be used by the BAM.

$$band(i) = \max\{a(j),\ j \geq i\},\quad i = 1,...,N \tag{6.16}$$

If the bandwidth is allocated using this nonincreasing function, there is no need to ask for additional bandwidth. Conversely, it is possible to reduce the allocated bandwidth when it is no longer needed. At time *j*, just before sending the quantity a(j), a request to deallocate the bandwidth is sent if *band(j)<band(j-1)*, and the new allocated bandwidth will be band(j) instead of band (j-1). The overhead introduced by the deallocation messages is very small compared to the video transmission. Experiments show that at most 20 deallocation messages are sufficient for a 28-minute video. The bandwidth use, U, achieved using our bandwidth allocation mechanisms is

$$U = \sum_{i=1}^{N} a(i) \,/\, \sum_{i=1}^{N} band(i) \tag{6.17}$$

It is greater than what is obtained using classic BAM because $band(i) \leq Peak\ rate, i = 1,...,N$.

6.4.4 Fine-Grained Scalable Video Coding for Multimedia Across IP

Multimedia streaming and the set of applications that rely on streaming are expected to continue growing. A primary objective of most researchers in this field is to mature Internet video solutions to the level when viewing of good-quality video of major broadcast television events across the Web becomes a reality. One generic framework that addresses both the video-coding and net-

working challenges associated with Internet video is scalability. Any scalable Internet video-coding solution has to enable a very simple and flexible streaming framework. Hence, it must meet the following requirements [6.127]:

- The solution must enable a streaming server to perform minimal real-time processing and rate control when outputting a very large number of simultaneous unicast streams.
- The scalable Internet video-coding approach has to be highly adaptable to unpredictable bandwidth variations due to heterogeneous access technologies of the receivers or due to dynamic changes in network conditions.
- The video-coding solution must enable low complexity decoding and low memory requirements to provide common receivers, in addition to powerful computers, the opportunity to stream and decode desired Internet video content.
- The streaming framework and related scalable video-coding approach should be able to support both unicast and multicast applications. This eliminates the need for coding content in different formats to serve different types of applications.
- The scalable bit stream must be resilient to packet loss events, which are quite common across the Internet.

The previous requirements were the primary drivers behind the design of the FGS video coding scheme [6.128]. For example, the 3D wavelet/subband-based coding schemes require large memory at the receiver, and, consequently they are undesirable for low complexity devices [6.73, 6.129]. In addition, some of the methods rely on motion compensation to improve the coding efficiency at the expense of sacrificing scalability and resilience to packet losses [6.129]. Other video-coding techniques totally avoid any motion compensation and consequently sacrifice a great deal of coding efficiency [6.73, 6.130].

The FGS framework strikes a good balance between coding efficiency and scalability while maintaining a very flexible and simple video-coding structure. When compared with other packet-loss-resilient streaming solutions, FGS has also demonstrated good resilience attributes under packet losses [6.131]. After new extensions and improvements to its original framework, FGS has been adopted in the MPEG-4 Video standard as the core video-coding method for MPEG-4 streaming applications [6.132]. Since the first version of the MPEG-4 FGS draft standard [6.133], there have been several improvements introduced to the FGS framework. In particular, there are three aspects of the improved FGS method. First, a very simple residual-computation approach was proposed. Despite its simplicity, this approach provides the same or better performance than the performance of a more elaborate residual computation method. Second, an adaptive quantization approach was proposed, and it resulted in two FGS-based video-coding tools. Third, a hybrid all-FGS scalability structure was also proposed. This novel FGS scalability structure enables quality (that is, SNR), temporal or both temporal-SNR scalable video coding and streaming [6.128].

In order to meet the requirements outlined, FGS encoding is designed to cover any desired bandwidth range while maintaining a very simple scalability structure. Examples of the FGS

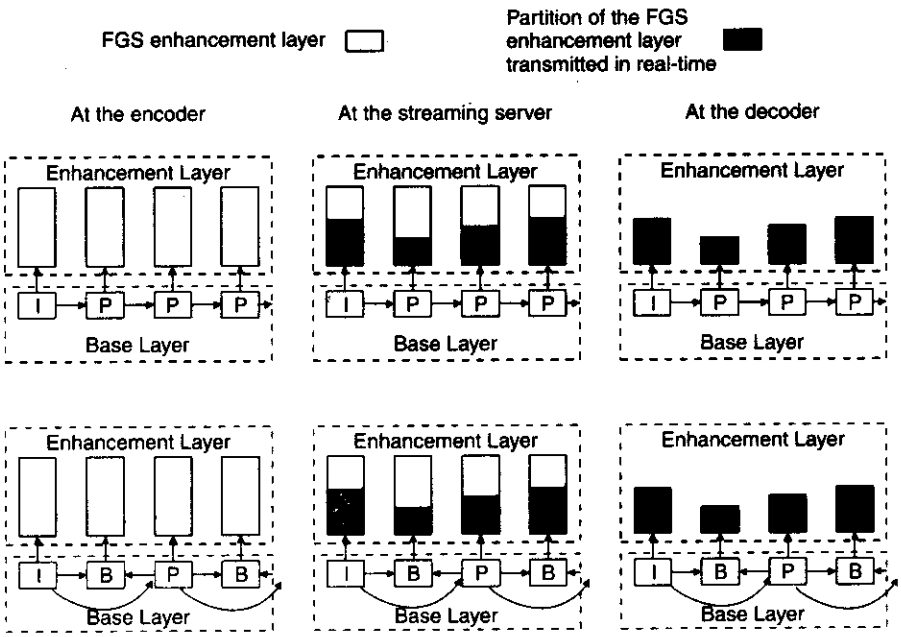

Figure 6.40 Examples of the FGS scalability structure [6.128]. ©2001 IEEE.

scalability structure at the encoder (left), streaming server (center) and decoder (right) for a typical unicast Internet-streaming application are shown in Figure 6.40. The top and bottom rows of the figure represent base layers without and with bidirectional (B) frames, respectively.

The FGS structure consists of only two layers: a base layer coded at a bit rate Rb and a single enhancement layer coded using a fine-granular scheme to a maximum bit rate of Re. This structure provides a very efficient, yet simple, level of abstraction between the encoding and the streaming processes. The encoder only needs to know the range of bandwidth [Rmin = Rb, Rmax = Re] over which it has to code the content. On the other hand, the streaming server has a total flexibility in sending any desired portion of any enhancement layer frame, without the need for performing complicated real-time rate-control algorithms. This enables the server to handle a very large number for unicast streaming sessions and to adapt to their bandwidth variations in real-time. On the receiver side, the FGS framework adds a small amount of complexity and memory requirements to any standard motion-compensation-based video decoder.

For multicast applications, FGS also provides a flexible framework for the encoding, streaming and decoding processes. Identical to the unicast case, the encoder compresses the content using any desired range of bandwidth [Rmin = Rb, Rmax =Re]. Therefore, the same compressed streams can be used for both unicast and multicast applications. At time of transmission, the multicast server positions the FGS enhancement layer into any of the preferred number of multicast channels, each of which can occupy a desired portion of the total bandwidth. Example of an FGS-based multicast scenario is given in Figure 6.41. The distribution of the base layer is

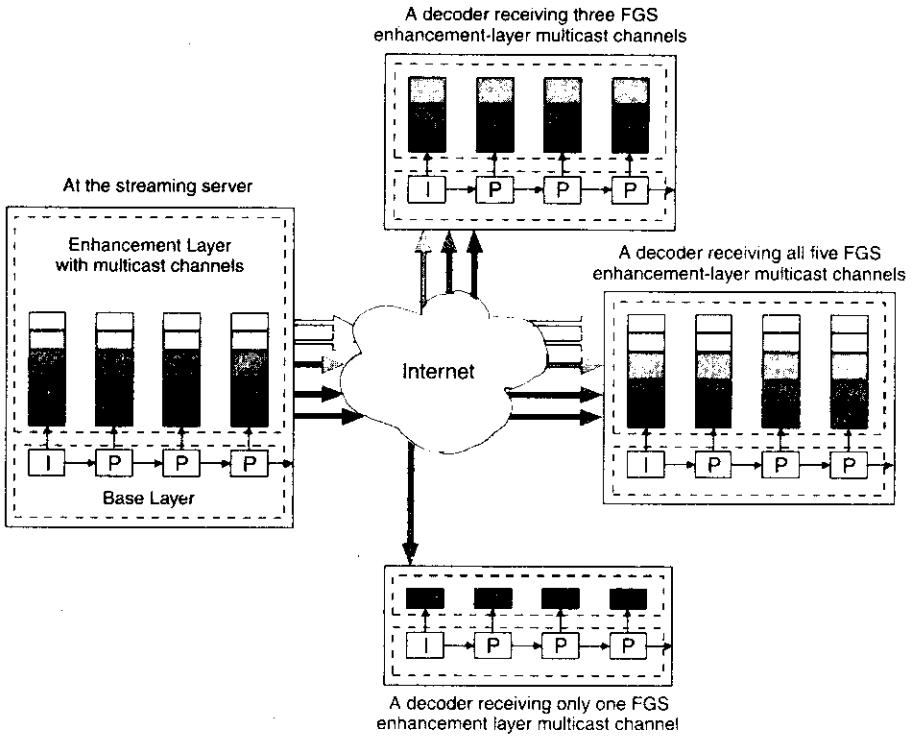

Figure 6.41 Examples of the FGS-based multicast scenario [6.128]. ©2001 IEEE.

implicit and therefore is not shown in the figure. At the decoder side, the receiver can subscribe to the base-layer channel and to any number of FGS enhancement layer channels that the receiver is capable of accessing. It is important to note that, regardless of the number of FGS enhancement layer channels that the receiver subscribes to, the decoder has to decode only a single enhancement layer.

The FGS framework requires two encoders, one for the base layer and the other for the enhancement layer. The base layer can be compressed using any motion-compensation video-encoding method. The DCT-based MPEG-4 Video standard is a good candidate for the base layer encoder due to its coding efficiency especially at low bit rates. Prior to introducing FGS, MPEG-4 included a very rich set of video-coding tools, most of which are applicable for the FGS base layer [6.134].

The FGS enhancement layer encoder can be based on any fine-granular coding method. When FGS was first introduced to MPEG-4, three approaches were proposed for coding the FGS enhancement layer: wavelet, DCT and matching pursuit-based methods. This led to several proposals and extensive evaluation of these and related approaches. In particular, the performances of different variations of bit-plane DCT-based coding and wavelet compression methods were studied, compared and presented [6.135]. Based on an analysis of the FGS enhancement

layer SNR, the study concluded that both bit-plane DCT coding and EZW based compression provide very similar results.

6.5 Multimedia Across DSLs

The Internet with all its applications is changing the way we work, live and spend time. However, today the Internet is facing a major problem. Growing demand for access has produced bottlenecks and traffic jams, which are slowing down the Internet. In an attempt to overcome these restrictions, access has pushed the technology of traditional telephony to new and innovative heights with the emergence of Asymmetric DSL (ADSL) technology. High-speed ADSL eliminates bottlenecks, giving all subscribers quick and reliable access to Internet content. Telecom service providers have yet to realize the full potential of ADSL. Traditional telephone and Internet services are only the beginning, but the ability to offer broadcast video services is a reality. Cable TV operators are beginning to offer voice and data services. There is increasing competition from Competitive Local Exchange Carriers (CLEC) and other carriers, making it imperative that traditional telecom service provides video services. By offering a range of services, established service providers can generate additional revenue and can protect their installed base. Direct Broadcast Satellite (DBS) providers, particularly in Europe and Asia, are offering a compelling Multichannel Video Program Distribution (MVPD) service [6.142].

A key factor contributing to the successful deployment of ADSL access systems has been the facility for overlying data services on top of existing voice service without interfering with the voice service. For the users, this offers the following:

- Always-on service capability. There is no need to dial up because the IP connection is always available and so is the office networking model in which network resources are available all the time.
- Virtual second voice line. Unlike when the user is connected through a modem, the voice line remains available for incoming and outgoing calls.

For the operator, the service overlay allows ADSL to be installed throughout the network, irrespective of what types of narrow band switches are installed. After the initial success of ADSL, it became apparent that it could be used to offer multiple phone lines together with a greater range of services (for example, VPNs) targeted at specific markets. This has been made possible by the high bandwidth of ADSL, backed up by progress in voice compression, echo cancelling and digital signal-processing technologies. ADSL offers a high data bandwidth, of which a portion can be used to offer additional voice services integrated with the data services. Symmetric DSL techniques, such as Single Pair High-Speed DSL (SHDSL) cannot be deployed as an overlay to existing analog telephone services, so the delivery of voice and data services using a single facility requires voice to be carried directly on the DSL link. The techniques used to transport voice and data in an integrated way across DSL, whether ADSL or SHDSL, are referred to as Voice over DSL (VoDSL).

With VoDSL, two main market segments are of interest to service providers. The first is small- to medium-sized businesses, a significant percentage of which need to be able to send and receive data of around 500 Kb/s. The voice needs of these customers are typically met by 4 to 12 outgoing lines. Using, for example, ADPCM voice coding, at peak times these phone lines consume only 128 to 256 Kb/s of the ADSL bandwidth, which is typically in excess of 2 Mb/s downstream and more than 500 Kb/s upstream. The second market interested in VoDSL services is residential users who will appreciate the extra two to four voice lines that VoDSL offers [6.143].

From the service provider's perspective, ADSL offers considerable opportunities in terms of providing source of incremental revenue and a way of reducing costs. Regardless of the type of operator, there are compelling reasons for the success of VoDSL services. Advantages for the user include ISDN voice quality; automated provisioning, which greatly reduces the time taken to add or remove services, handle data and voice service with one-stop-shopping; a single bill and a common helpdesk. The keys to success are the bundling of both data and voice lines and pricing flexibility [6.136 through 6.141].

ADSL will be delivering multimedia services to millions of users. The transmission of digital multimedia data requires the existing systems to be augmented with functions that can handle more than just ordinary data. In addition, the high volume of multimedia data can be handled efficiently only if all available system services are carefully optimized.

6.5.1 VODSL Architecture

The architecture deployed by a telecom service provider to deliver video services will vary. A typical example is shown in Figure 6.42. In the access network, the ATM provides layer 2 connectivity across ADSL.

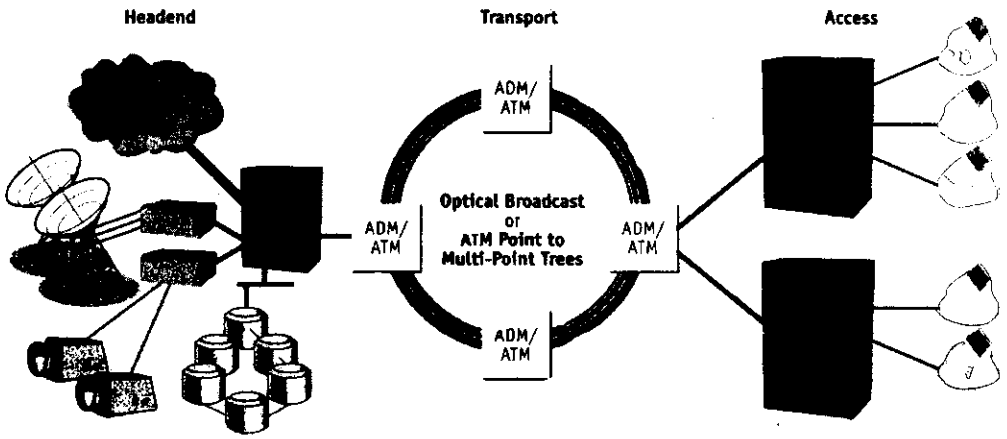

Figure 6.42 Architecture deployed to deliver video services.

Thus, each DSL Access Multiplexer (DSLAM) is either an ATM multiplexer or switch. As a result, video programs must be delivered in either MPEG-over ATM format or MPEG-over-IP-over-ATM format. Both technologies are currently available, but the market appears to be favoring the IP as the network layer delivery vehicle. Although IP adds some overhead to the video stream, it greatly simplifies in-home distribution across Ethernet-compatible media. Also, more applications are available for IP, broadening its audience. In both cases, the headend and transport networks are similar. The term headend originated in the cable industry and is used here to denote a location where content is aggregated for TV channels, VoD, e-commerce portals, Internet access and so on. The location of headend, and even whether it is centralized or distributed, is an architectural choice. As the video content is delivered to the user across the ATM access network, content can be injected into the network at almost any location. In the case of a broadcast TV service, video arrives from various services across diverse media, including DBS, local off-air broadcast and studios. Content from all these services has to be fed to an encoding platform and converted into MPEG format, if not already in this format. Given that the end-delivery network is ADSL, it is highly recommended that the output video signals should be shaped to optimize link use and to ensure that the ADSL line is not overloaded. The output channels are delivered to an ATM network using either MPEG-over-IP-over-ATM or MPEG-over-ATM encapsulation. Interactive services, such as VoD and network-based time-shifted TV, are delivered from servers, which store content in MPEG format and deliver a copy at the subscriber's request. The server must be dimensioned for both the amount of content it must store and the number of active subscribers retrieving data. Single large servers or multiple distributed servers can be used to meet this requirement. The trade-off is between transport costs, replicated server costs and management complexity. Other servers for a variety of video services can also be collected at the headend. As for the headend in a VoDSL architecture, it can be centralized or distributed. Because the content is distributed using IP and/or ATM, connectivity is very flexible.

The role of the transport network is to deliver the content from the headend locations to the appropriate DSLAMs, or their attached switchs/routers, in the access network. The network must transport two specific types of traffic: multicast and unicast, corresponding to the broadcast and interactive services.

Broadcast traffic is transported as IP multicast, ATM point-to-multipoint or a combination of the two. IP multicast overlay using ATM is shown in Figure 6.43. Traffic must be delivered to all DSLAM locations in the network, essentially emulating a cable service that delivers all channels at all locations. Given that the traffic is either IP or ATM, the choices for constructing the distribution network are ATM point-to-multipoint or IP multicast. A good solution for an overlay network is to use ATM point-to-multipoint connections in an ATM-switching environment. ATM is stable technology with the proven ability to replicate high bandwidth data. This approach will work across almost any transport network, such as SONET/Synchronous Digital Hierarchy (SONET/SDH) or Dense Division Multiplexing (DDM), and supports native MPEG-over-ATM and MPEG-over-IP encapsulation. The links that carry the broadcast channels can also be used to transport other data, such as interactive content. The downside to this multicast overlay

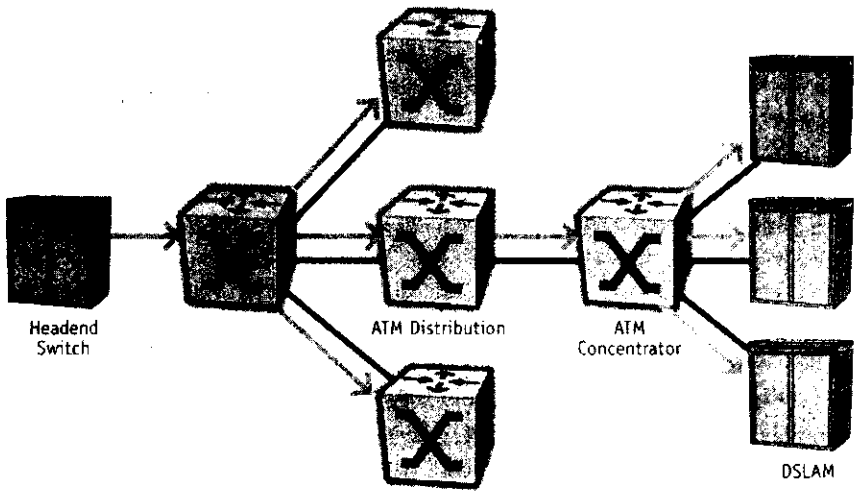

Figure 6.43 IP multicast overlay using ATM.

approach is the increased cost of supporting multiple optical transport links and any intermediate ATM switches required to complete the point-to-multipoint. ATM switches attached to the DSLAMs are not included as extra costs, because these switches will typically exist within the network.

IP multicast-capable routers can also be used to distribute the broadcast TV channels if IP is the network layer chosen for the service (Figure 6.44). If an existing IP network provides the required capacity and performance for multicast replication, then it may be feasible to add broadcast television streams. The end-delivery encapsulation is ATM, so the IP multicast streams must be encapsulated into ATM virtual circuits for the final leg of the journey. To ensure high-quality video, IP networks must also be properly engineered to deliver QoS.

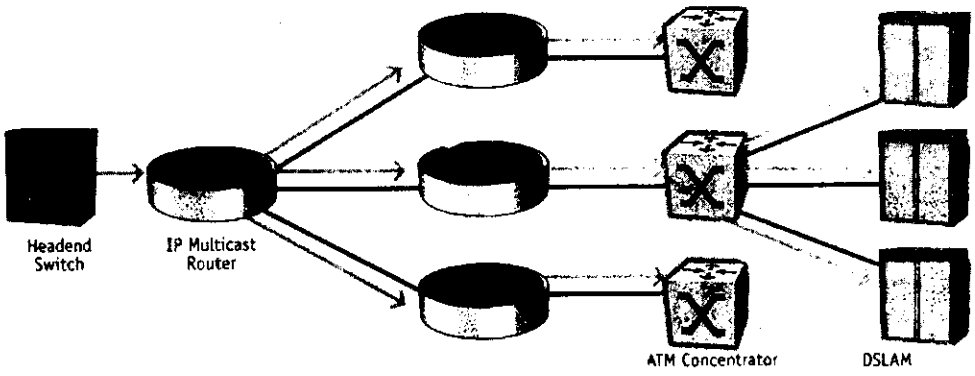

Figure 6.44 IP multicast overlay using routers.

The ADSL access network is well suited to point-to-point IP architectures. Many different architectures can be chosen for building a unicast network, including existing Broadband Remote Access Servers (BRAS), ATM switch/routers and IP cards within the DSLAM connected to routers.

Unicast and broadcast services can be delivered across the same network infrastructure. For example, the ATM concentrator nodes that aggregate the DSLAMs support both point-to-multipoint and point-to-point virtual circuits. Bidirectional, interactive traffic across ATM point-to-point virtual circuits can be aggregated at either BRAS or router, depending on the service requirements. These routing devices, located within certain central offices, can then be connected to a data center across the same optical transport medium that delivers the broadcast traffic.

The DSLAM is the last element in the access network before the subscriber's home. It is responsible for switching the video channels delivered to the subscriber. In the interest of service response (rapid channel changing) and bandwidth savings, the nearer the multicasting device is to the subscriber, means the better the offered service. To meet the performance requirements, the DSLAM must always support multicasting in hardware. DSLAM multicast replication is shown in Figure 6.45.

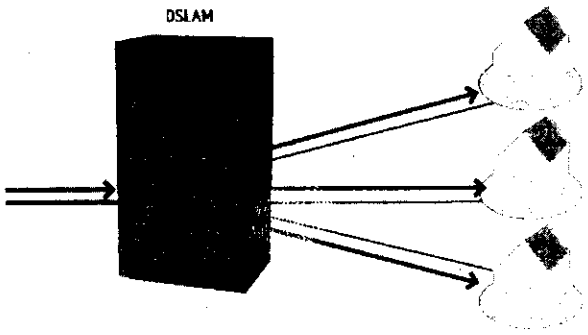

Figure 6.45 DSLAM multicast replication.

However, the integrated DSLAM approach is not ideal in cases where the service provider has an installed base of DSLAMs that do not support such features. Also, broadband Digital Loop Carriers (DLC) with ADSL links are unlikely to provide any multicast switching capabilities. Thus, it is also necessary to offer multicast switching using an external device. Typically, this will be either an IP multicast router or ATM switch supporting logical multicast (multiple point-to-multipoint leaves from the same connection on the same port), or a combination of the two. Multicast for existing DSLAMs is shown in Figure 6.46. Note that the uplink from the DSLAM/DLC to the switching device will constrain the number of video subscribers supported by the DSLAM, because all content channels are treated as unicast from the switching device onward. Unicast interactive traffic must also travel through the DSLAM, so both multiple virtual circuits and QoS guarantees must be available within the DSLAM to support both broadcast and interactive services concurrently. The strength of the ATM access networks lies in its use of virtual circuits (Figure 6.47.)

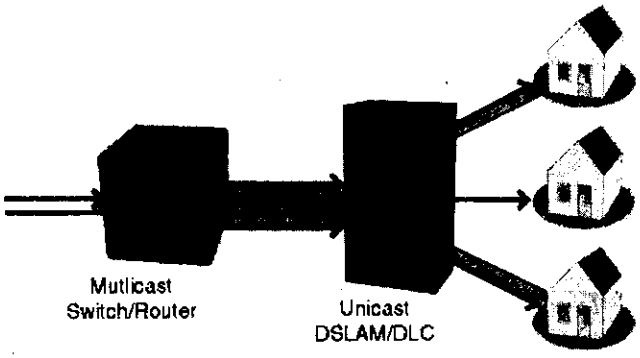

Figure 6.46 Multicast for existing DSLAMs.

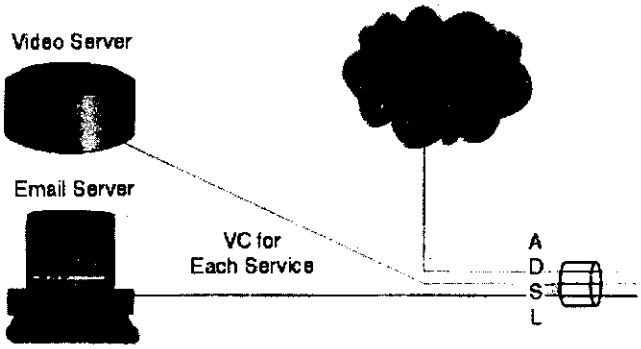

Figure 6.47 Multiple services to a subscriber.

A single subscriber might require multiple services, each of which is best served by a unique device. For example, high-speed Internet access traffic might best be delivered through a BRAS, which provides a rich feature set for accounting. Multiple services to a subscriber are presented in Figure 6.47.

After the VoDSL channel is terminated at the subscriber's premises by a DSL modem, it is necessary to distribute the content to the set-top box so that it can be viewed on the television. This is typically done via Ethernet, which can also be connected to the PC.

When the video is encapsulated as MPEG-over-IP-over-ATM, there are more options for in-home distribution. A variety of Ethernet-compatible media are available or under development, including wireless Ethernet, wired Ethernet, Home Phoneline Networking Alliance (HPNA) and Powerline technologies. Obviously, media that do not require new in-home wiring are very attractive because they considerably reduce the cost of home installations and the need to send an engineer to the subscriber's premises.

Wireless Ethernet is one of the most promising emerging technologies for rapid home installation. Use of wireless Ethernet in the home is represented in Figure 6.48.

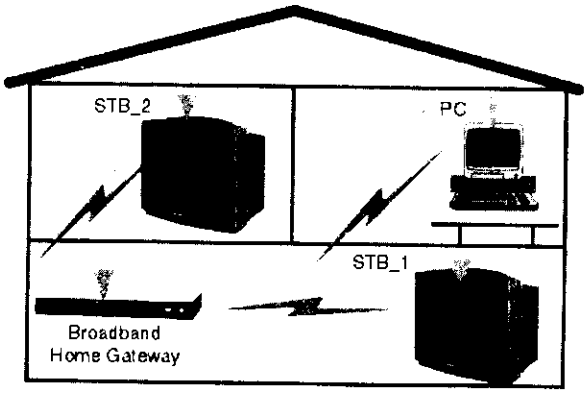

Figure 6.48 Use of wireless Ethernet in the home.

The DSL modem becomes an integral part of a broadband home gateway, which supports home links, such as wireless Ethernet, to communicate with IP devices in the home. A television set-top box could be one of these devices, so the IP video stream is directed from the home gateway to the set-top box. This is possible using the IEEE 802.11b standard, which can support up to 11 Mb/s, sufficient to supply two remote set-top boxes. Future technologies promise to increase the bandwidth of the wireless connection to 20 or 30 Mb/s.

6.5.2 Delivering Voice Services Across DSL

Delivering VoDSL offers a lucrative opportunity for both established and emerging providers. Today ADSL users can receive one or more digital voice lines on top of their existing analog telephone line. Depending on the target ADSL penetration and the network application, it will become economically attractive to implement data-only access networks, based on pure ATM DSLAMs.

Several benefits of VoDSL are reinforced by a network solution that brings about the convergence of voice and data in both the access and core networks. Several architectures have been proposed in the context of VoIP. The term "Voice over Packets (VoP)" is used to refer to these architectures because in many cases the top-level features and advantages remain valid whether voice and data are carried in ATM, IP frames on IP on top of ATM or the access or core network. At the transport level, much of the final architecture will depend on the deployment scheme followed by the service provider. Evolution from VoDSL to next generation network is shown in Figure 6.49. In a next generation network, voice calls are no longer handled by exchanges, but by a central high capacity call server. This server controls the gateways that perform the conversion to VoP. Already next generation network architectures are widely deployed in the core network. Trunk gateways are typically positioned between the local exchange and a packet-based network (ATM or IP). The transit exchanges are then replaced by more cost-effective router equipment. Control servers centralize control, thereby reducing operating costs, and voice compression saves on bandwidth, particularly on long distance and intercontinental links. The initial VoDSL scenario introduces gateway functionality at or in the CPE and access nodes.

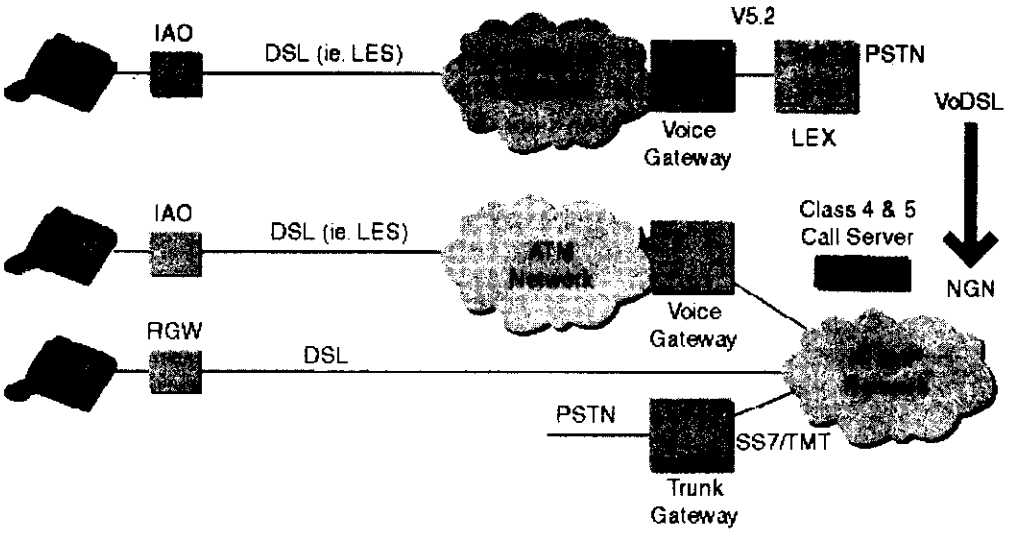

Figure 6.49 Evaluation from VoDSL to Next Generation Network (NGN).

6.5.3 Multimedia Across ADSL

The primary motivation behind ADSL is the delivery of multimedia services. Given the fast growth of Internet and multimedia applications, the widespread acceptance of ADSL systems will depend on the ability to provide efficient delivery and refined quality of multimedia data to the subscribers. Multimedia data has quite different characteristics compared to general data. One major characteristic is the layer coded structure where multimedia data is constructed into separate data streams, each representing a layer. The layers have different QoS requirements, that is, data rate and error performance (Bit Error Rate [BER] and Symbol Error Rate [SER]).

With limited communication resources, for example, bandwidth and transmitted power, a key design issue in multimedia communications is to handle the layers differently and, therefore, efficiently. A larger amount of channel resources should be assigned to the layers with higher importance. For example, it is well known that the use of scalability can enhance the error robustness of a video service. There are two transmission schemes for multimedia data across ADSL: serial transmission and parallel transmission.

Serial Transmission: TDM

ADSL transmission is divided into time slots. In each time slot, only data from a single layer is transmitted. The layers are time-division multiplexed. The design task is finding a time slot for layer assignment to achieve high-efficiency transmission and to provide an acceptable QoS to the users. Such a system is named serial transmission [6.144]. The time slot as well as subchannel structure for the serial transmission is shown in Figure 6.50. For these source layers, time slots 1 and 2 are assigned to transmitting layer 1, and slots 3 and 4 are assigned to transmitting layers 2 and 3, respectively. The subchannel power and bit rate distributions are different in slots

1 and 2 compared to 3 and 4. It is important to note that, within a single time slot, the power and bit rate are allocated so that all the usable subchannels perform at the same error rate. For two time slots transmitting different layers, the subchannels' error performances are completely different. An error performance distribution across the time slots and subchannels is illustrated in Figure 6.51. The system configuration is with four layers from single or multiple sources, 256 subchannels and 10 time slots. The BER is constant within the same time slot and different across the time slots.

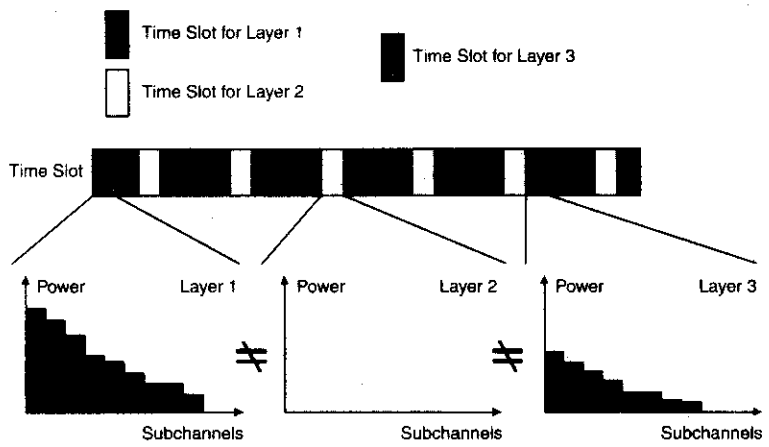

Figure 6.50 Serial transmission for multimedia data across ADSL [6.145]. ©2000 IEEE.

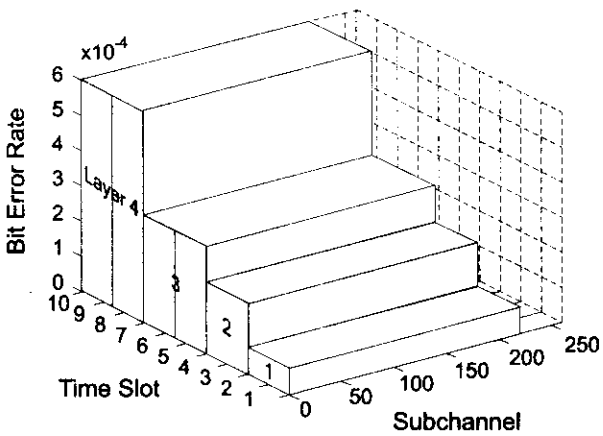

Figure 6.51 Error performance across the time slots and subchannels for serial multimedia data transmission across ADSL [6.145]. ©2000 IEEE.