Asynchronous Transfer of Video

Gunnar Karlsson
Swedish Institute of Computer Science (SICS)
Box 1263
S–164 28 Kista
Sweden

Video transfers across IP and ATM networks have received much research attention during the last ten years. Various video services are expected in the future, enabled by the rapid development in video coding and broadband network technology. This paper gives an introduction to the issues involved in asynchronous video transfers. Brief overviews of video coding, rate control, multiplexing, as well as delay, error and loss control are given.

Section 1. Background

A collection of the world's telecommunication operators and their equipment suppliers are currently standardizing the asynchronous transfer mode (ATM) within the International Telecommunication Union (ITU) as a basis for broadband integrated services digital networks. In parallel with the ITU, the ATM Forum is furthering the development process by including computer communication manufacturers, as well as end–users in the standardization process. Although the representations give different perspectives on the proposed network solutions, the overall goal of building networks capable of transferring multimedia efficiently is shared between the ITU and the ATM Forum.

In addition to the work on ATM, new developments for internet protocols lean towards supporting full–fledged multimedia communications. The simple internet protocol plus (RFC 1710) has been selected by the Internet Engineering Task Force as a basis for the next generation internet protocol (version 6, often denoted IPng; see RFC 1752). It features connection–oriented services, denoted flows, and signaling by means of the resource reservation protocol (RSVP) [103].

An implied assumption in the development of both ATM and IPng is that network users will get access to copious amounts of transmission capacity at affordable prices, even in the local loop. This capacity is needed to lower the latency of bulk data transfers and to enable audio and video communications. The provision of the latter has received considerable attention among researchers and is the topic of this paper.

Asynchronous transfer of video, which often is referred to as ‘‘packet video,’’ can be defined as the transfer of video signals over asynchronously time–division multiplexed (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 inherit time–constraints (a regular one and a half hour film in VHS quality is over a giga–byte of data). In addition to the requirement on pacing, there may also be bounds on the maximal transfer delay from camera to monitor if the video is part of an interactive conversation or conference. These limits are set by human perception and determine when the delay starts to impede the information exchange.

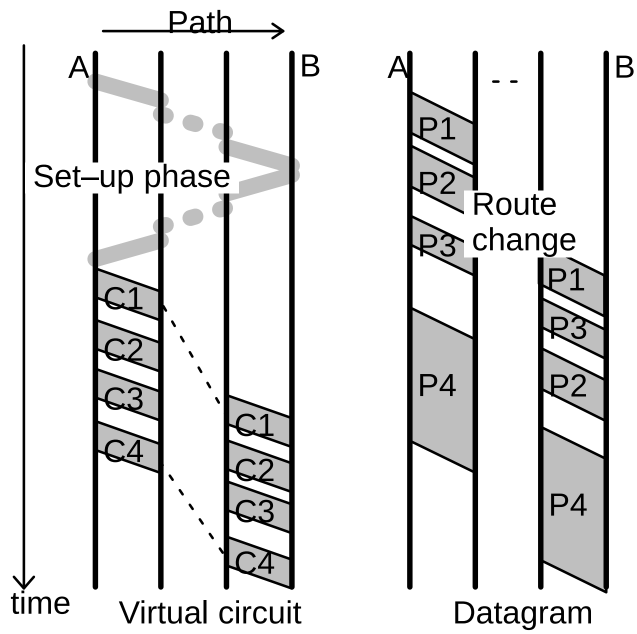
Parts of the signal may be lost or corrupted by errors during the transfer. This will reduce the quality of the reconstructed video and, if the degradation is serious enough, it may cause the viewer to reject the service.

The general topic of packet video is thus to code and asynchronously transfer video signals under quality constraints. The research field is active with its own set of workshops: the International Workshops on Packet Video the first one being held at Columbia University in 1987. There is also the more general International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV). There are, in addition, sessions on the topic held at many of the international signal and image processing, and communications conferences. Papers appear in the corresponding journals and transactions. Two books have to date been published in the field [61][74] and several general papers such as [20][45][99][104].

This survey attempts to introduce the issues and their possible solutions to a networking and signal processing audience and has the following structure: in Section 2 we take a brief look at asynchronous transfers by means of ATM and IP, Section 3 describes the digital video signal format and Section 4 the video communication system. Section 5 contains a summary of the perceptual requirements that a video application may place on a communication system. Section 6 contains a discussion of the role of video in applications.

The purpose of Section 7 is to provide insight into the tradeoffs faced in the systems design. The principles of video coding are reviewed in Section 8 and, as an example, the MPEG coding scheme is explained. We then progress to the communication system. Section 9 presents rate control, and Section 10 the information transfer which includes the multiplexing. Receiver functionality is covered with delay related issues in Section 11, and error and loss related issues in Section 12. Section 13 presents ATM and Internet protocols for video transfers. Section 14 concludes the paper.

Fig. 1 Illustration of virtual circuit and datagram routing. Bit–encoded information is sent in fixed–sized cells along pre–established routes in ATM networks, and in variable–sized packets without any pre–selection of a route in datagram IP networks.



Section 2. Asynchronous transfer

The asynchronous 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 5–octet header. Traffic on a particular link will consist 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 sent. The intention is that ATM networks should be able to guarantee the quality of service in terms of cell loss and maximum delay, as well as maximum delay variations. (An overview of the standards is given in [54].)

The internet protocol 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 the resource reservation protocol RSVP. IPng packets contain a 24–bit flow identifier in addition to the source and destination addresses which can be used in routers for operations like scheduling and buffer management to provide service guarantees (a flow is comparable to a virtual circuit in ATM). The difference between virtual circuits and datagram routing is illustrated in Fig. 1.

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 impulse noise. Loss of information is mainly caused by multiplexing overload of such magnitude and duration that buffers in the nodes overflow. Loss may also be caused by misrouting due to bit–errors in the addresses, but it is less probable.

Section 3. Digital video

Video in digital form is a three dimensional signal; it is a time sequence of equidistantly spaced two dimensional 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 out 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, then it is transformed into a YIQ format with one luminance component (Y) and two chrominance, or color, components (I and Q).

The color transformation compacts most of the signal energy into the Y component and the color components are at times sampled at a lower frequency than the luminance component. The resolution of a YIQ signal is commonly denoted by Y:I:Q where Y is 4 for the full sampling rate (13.5 MHz for ITU–R BT.601), and I and Q are 4, 2 or 1 for full, half or quarter resolution, respectively. Normally a pixel (picture element) is represented by 8 bits per component.

Fig. 2 Structure of the video signal

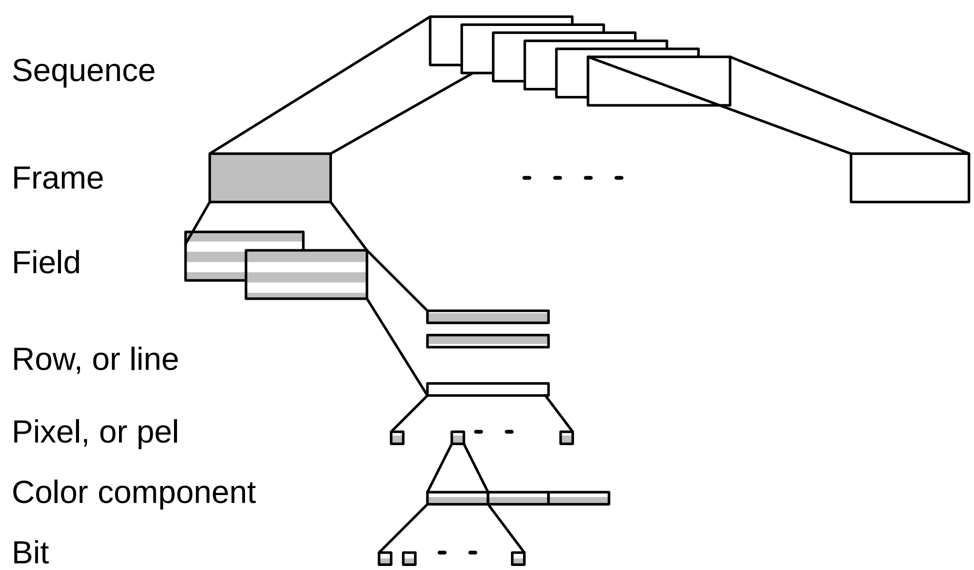
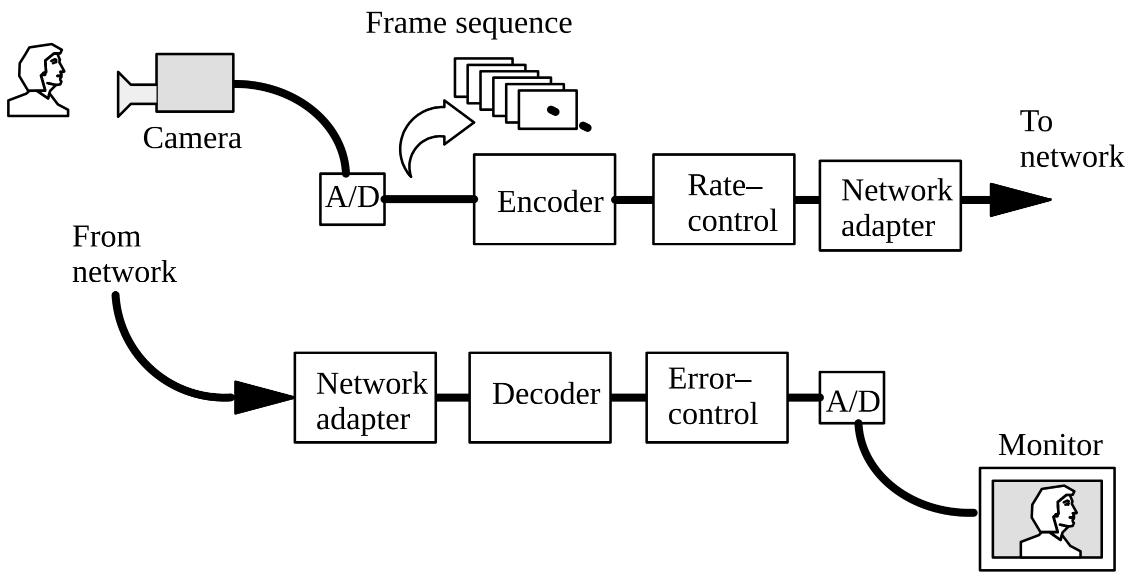


Table I. Some European digital video formats.

Format	Width [pixels]	Height [pixels]	Frames/s	Y:I:Q	Bit rate [Mb/s] Without and with coding	
HDTV	1920	1250	25 (50)	4:2:2	960	20—40
ITU–R BT.601	720	576	25	4:2:2	166	5—10
CIF	360	288	6.25—25	4:1:1	7.8—31	1—3
QCIF	180	144	6.25—25	4:1:1	1.9—7.8	0.064—1

Fig. 3 An example of a video transmission system: the camera continuously captures a scene; the signal is digitized, coded and forwarded to the network. The receiver side unpacks, decodes and converts the data to analog form to be displayed.



The structure of a video stream is illustrated in Fig. 2. The stream consists of frames which 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 of which each has a fixed number of bits.

Some common digital European video formats are given in Table I. The high–definition television format may have a frame–rate of 25 or 50 frames per second for interleaved or sequential scanning, respectively (ITU–R BT.709–1 ”Basic parameter values for the HDTV standard for the studio and for international program exchange”, see also [73][81]). The ITU–R BT.601 is a digital studio format for television. CIF—the common interchange format—has a quarter of the spatial resolution of ITU–R BT.601, with an optionally lower frame–rate, and QCIF is a quarter of the spatial size of CIF. Both CIF and QCIF are sequentially scanned. (American formats have 30 frames per second and 5/6 the height.) The coded bit rates are simple estimates.

Section 4. The video communication system

A video communication system is shown in Fig. 3. 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. It is needed when there are restrictions on the permissible bit rate from the coder, typically a common restriction is that of the access capacity to the network. The constraint need not, however, be a single upper limit on the rate but could be a more general function. It can also be affected by flow–control messages from the network as well as from the receiver.

Before or after 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 lead to the loss of one or more application segments. Further segmentation occurs at the network level, where the data are segmented into multiplexing units (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 which even may include error–control coding.

The receiver side performs functions which are the reciprocal of 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.

Section 5. Perceptual requirements

The communication system has to limit information loss and delay according to requirements placed by the video application. The expected perceptual quality of the received video should of course be adequate for its intended use. Whether perceptible loss is caused by encoding, buffer overflow or unsuccessful re–synchronization is irrelevant to a user. Most of the information loss objectively measured should be incurred at the encoding. The coding is done with awareness to the signal's information contents and possibly also its application context. The introduced error can therefore be made as imperceptible as possible. Information loss during the transfer may, however, interfere with the signal at any point and, although the probability of loss may be low, it could cause annoying artifacts in the reconstructed video.

Acceptable levels for transfer loss and error are difficult to determine since they depend on criteria such as the use and cost of the video transfer, the duration of the session, the quality of the source material, activity in the captured scene, and the appearance of the loss and errors in the reconstructed signal [31][39][93]. It is possible to reduce the visibility of loss by signal processing means (so called loss and error concealment). The probability of occurrence may also be reduced by forward error correction when the loss events are independent.

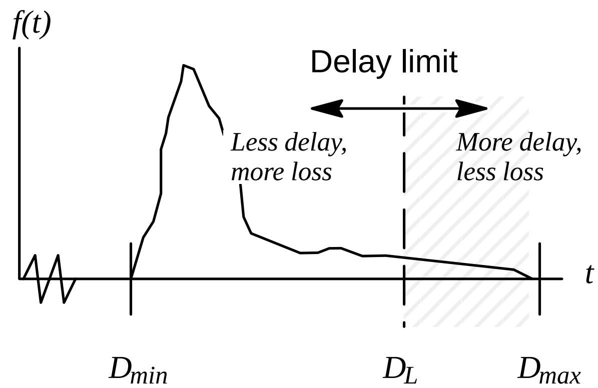
For ATM, the target loss probability for video is often taken to be 10^{-9}. This author has, however, not been able to find a justification for it in the research literature. The network will be operated uneconomically if it turns out to be overly precautious. There are overall few results reported on the perceptual aspects of video transfers and there are consequently few guidelines to follow, especially concerning acceptable loss levels.

In [40] there are, however, some results that can be used in designing a video transfer system. The factors that have greatest impact on quality are the number of lost cells or packets, number of pixels in an impaired region and its shape, as well as the ”burstiness” of the loss. For the latter, random cell or packet losses were found to yield greater quality degradation than clustered losses at equal loss ratios. Thus, for a given loss probability one may safely assume uncorrelated loss events which give an upper bound on the quality degradation.

The isochronal sampling of the signal imposes requirements on regular pacing of the signal at the digital to analog conversion. The applications may place delay constraints on the video transfer when it is bidirectional and part of an interactive conversation or conference. The recommendations on maximal delay follow those for voice conversations (ITU–T Rec. G.114: Mean one–way propagation time) [56]. There is consequently no or little impact below 150 ms one–way delay, and serious impact above 400 ms. Video can antecede the associated audio by up to 100 ms or follow it by at most 20 ms for one–way sessions [12]. Interactive sessions show less clear limits due to differences caused by the type and content of the conversation [56].

Are there perceptual constraints on the delay–variations? The most common approach to handle the variations is to impose a delay limit in accordance with the guidelines given above, and to delay all data up to that limit (see Fig. 4). Jitter is thus of little concern when the maximum delay in the network is below the tolerable end to end limit; when this is not the case, arrivals with delay above the limit are discarded at their reception and excessive delay is turned into loss. For a given delay distribution there is consequently a balance between the amount of delay and the probability of loss. User preferences in this balance have not been reported.

Fig. 4 The trade–off between delay and loss.



Clark *et alii* [15] discuss ‘‘adaptive applications’’ for which the enforced delay limit may be varied, which for video would mean that the scan–out time of a frame should be changed (from the fixed 33 or 40 ms per frame). This may, for instance, be accomplished by shortening or stretching the horizontal and vertical beam tracing periods of the display’s cathode ray. Adaptive scanning works under the provision that external control of the display’s beam tracing is possible.

If the magnitude of the delay variations is very large, then the adjustments may be made by skipping or repeating whole frames. It is, however, a crude form of control. It should be remembered that any deviation from the fixed sample period will result in spectral distortion (temporal aliasing) since the signal is isochronally sampled. There are no psycho–visual tests reported on the perceptual effects of ‘‘adaptive applications’’ to determine the limits of such a policy. User acceptance of various frame rates is discussed in [2].

A final note on perceptual issues: There are limits under which further improvements in delay and loss are unnoticeable to the eye. Passing these limits do not lead to improvements for the user while it may increase the complexity of the control functions and waste capacity in the network.

Section 6. The role of video

When viewed from the user’s perspective, the question arises about the role of video in communication applications. For what applications is video vital, for which ones is it useful, and furthermore where is it of no concern? Table II. summarizes the role of video for a selection of applications (this section is based on the work presented in [49]).

Some clarifications may be warranted: Television is assumed to provide the type of programming that is presently received via terrestrial or satellite broadcasts (basic and extended service, pay per view, and so on). Video on demand [19] includes movies, concerts, news, sports events, and other prerecorded material which can be retrieved from a server, as well as interactive television [6]. It allows for the type of viewing we presently have when using traditional video recorders. Games are similar to video on demand in that it is a retrieval from a server, which could be a rendering engine in case of virtual reality. Games can be assumed to have a higher degree of interaction with the server than video on demand.

Video is seen as an enabler if a particular application is not possible without it. Regular television, for example, without video is not radio. Video is therefore considered vital to television. Video is an enhancer if the application in question is fully operable without video, but the addition of video might enhance its quality or attractivness to the user. On one hand, it is interesting to note that none of the enabled applications require switched broadband networks. On the other, there are several applications for which video is relevant, but not vital, and require the services of a switched network.

Table II. The role of video in applications

Applications	Video as	
	enabler	enhancer
Television	❍	
Video on Demand	❍	
Shopping		❍
Games		❍
Medicine		❍
Surveillance		❍
Teaching		❍
Conferencing		❍
Telephony		❍

Telephony, for instance, is not necessarily enhanced by adding video; the picture–phone idea has been tried and failed, teleconferencing may have a similar fate. After all, one can do perfectly well with a wideband audio channel (eg, 7 kHz audio bandwidth according to ITU–T Rec. G.722), and a few shared documents. Shopping may be enhanced if one can show video clips of the wares instead of still images but it is not proven.

Note that the added utility in applications enhanced by video might not justify the extra cost of offering it. The minimum price to offer a video application may therefore not relate to its perceived usefulness. This should be kept in mind when developing multimedia communication systems. Cost–efficiency in the provision of video is germane.

Section 7. System Trade–offs

The new view on video coding and networking given by the asynchronous transfer is indicated by the trade–offs in Table III. As mentioned above, the user expects an adequate video quality at the lowest possible cost and this is the optimization to be performed for a session.

Table III. Trade–offs when transmitting video over asynchronous networks.

	Maximize	**Minimize**
Session	Quality	Cost
Encoding	Quality	Bit rate
Bit–rate control	Consistency of quality	Bit–rate variability
Transfer	Utilization	Queuing
Error control	Error recovery	Overhead

The role of the encoding is to maximize the quality for a given rate of reduction in average bit rate (or vice versa), this role is not directly dependent on the transfer mode. The network should minimize the queuing to reduce delay and loss while maximally utilizing network resources (such as the transmission capacity). This dual goal can actually be met by smoothing the flows prior to multiplexing. Hence, the role of the bit–rate control is implied: to minimize the variability of the encoded bit stream while retaining uniformity in quality.

Error control is needed to aid the recovery from losses and errors in the network while adding only a minimum amount of overhead information to the signal. Good error handling allows more loss caused by the queuing and thus the utilization can be increased on behalf of more queuing. The increased utilization must, of course, compensate the additional overhead in the framing of the data.

Consequently, a main issue is how to regulate the encoder to retain a good and consistent quality while at the same time allow good multiplexing performance in the network. In general, economizing on a single resource, such as the transmission capacity, may not give a total solution with the lowest cost. The discussed trade–offs should be considered together to find, at least in principle, suitable solutions close to an optimum for the complete system. The flexibility offered by asynchronous transfer of video is at the expense of a stronger dependency between the application and the network.

Section 8. Video coding

Video has prodigal capacity requirements. Consider the cases in Table I.: an uncompressed digital television signal (ITU–R BT.601) requires around 170 Mb/s. It can, however, be coded to 5–10 Mb/s with reasonably good quality that is sufficient for distribution. This should be taken as an indication that source coding (compression) is required to make digital video transfers to the public technically and economically feasible. Another reason is storage requirements as it may be advantageous to store and transmit the signal in a single format. In addition, storage will remain limited even if the networks' capacity becomes virtually unlimited.

The functional blocks of a video coding algorithm are:

1) energy compaction,

2) quantization, and

3) representation.

The following can be taken as an informal rationale for the listed functions. Assume a digital (video) signal with a fixed number of bits per sample. This means that all sample values are represented by the same number of bits even though the values may have very different likelihoods of appearing in the signal (see Fig. 5). Hence, a new *representation* which assigns shorter codewords using fewer bits to frequent sample values and longer codewords with more bits to infrequent values may actually represent the signal more efficiently (an example is given in Table IV.). A concise representation may lower the bit rate a few times depending on the distribution of the sample values. The procedure is called entropy coding, or variable length coding.

Fig. 5 Illustration of distribution of intensity values from 0 to 255 in a video sequence.

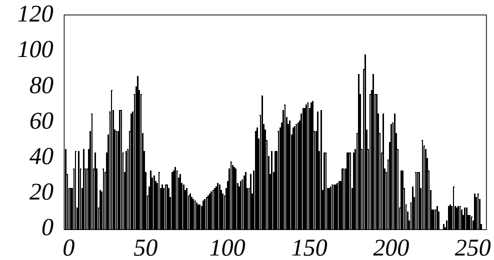


Table IV. An example of entropy coding: Huffman encoding, 2 bits per value in and 1.6 bits per value out.

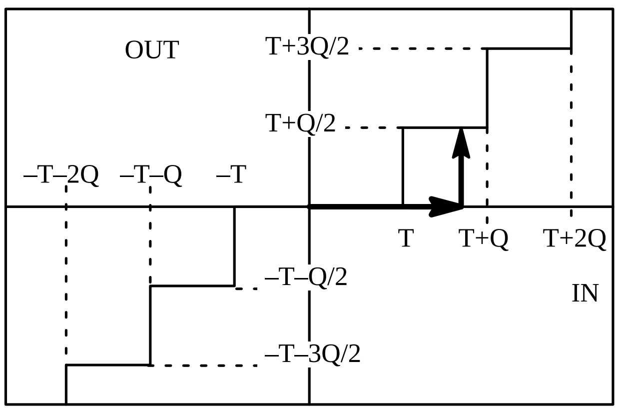
Quant–ization level	Input code–word	Frequency (prob.)	Output code–word
0	0 0	0.6	0
1	0 1	0.15	1 0 0
2	1 0	0.2	1 1
3	1 1	0.05	1 0 1

The lower limit of entropy coding is one bit per sample. To achieve further reduction, contiguous samples of identical values may be represented together as a duple giving the actual value and the length of the run. For example, the string of values '0 0 0 0 0 0 0 0 1 1 2 2 2 2 2' would yield the list of duples (0; 8), (1; 2), (2; 5). Not surprisingly, this method is called run–length coding. The duples are represented by a variable length code and the overall representation can average less than one bit per sample.

If yet greater reductions in bit rate are needed, then the admissible sample values of the input signal can be limited to, say, half or a quarter of the number of values in the original signal. This is done before the change of representation. Such *quantization*, when a value is rounded to its nearest permissible output value, distorts the signal. Fig. 6 shows a symmetric quantizer that is specified by three parameters: the width of the zero level, the step size and the number of steps.

With higher degrees of quantization the error becomes visible as artificial contours between intensity levels. It is especially noticeable in areas with smoothly varying shading. At this point, it is important to recognize that the original signal may not be well suited for coarse quantization, rather the signal should undergo an *energy compaction* before being quantized. The compacted signal form is often such that many values incur only a slight round off error while a few values get coarsely quantized. Hence, a higher quantization could be applied while still yielding a perceptually unobjectionable distortion.

Fig. 6 A quantizer. It is specified by an inner zone given by threshold T where all values are truncated to zero, by a quantization step size Q and by the number of steps.



It should be clear from this exposition that the distortion is fully and solely caused by the quantization; the energy compaction and the entropy coding are lossless, reversible processes. If distortion cannot be accepted, then the algorithm has to be constructed without quantization. The bit–rate reduction will consequently be severely limited. The entropy coding leads to varying numbers of bit per compressed frame; the maximum and the minimum can differ by up to orders of magnitude.

A full video coding algorithm takes the input signal and applies one or more forms of energy compaction, quantizes the resultant values and gives them a representation which aims at assigning the minimum amount of bits per value.

The three most common techniques for energy compaction are prediction, subband analysis [46], and orthogonal transformation. Orthogonal transforms are basically a subset of the subband analysis systems but are usually treated separately. Quantization can be done one value at a time, denoted by scalar quantization, or a group at a time, called vector quantization. Entropy coding includes Huffman, Lempel–Ziv and arithmetic coding as well as variations on these techniques.

We will illustrate the video coding with the scheme standardized by the ISO/IEC Motion Picture Experts Group (ISO/IEC JTC 1/SC 29/WG 11, *CD 13818 Generic coding of moving pictures and associated audio information*) .

Example: MPEG encoding

The MPEG coding scheme (see [60]) uses two forms of energy compaction: prediction is used to exploit temporal redundancy, and discrete–cosine transformation is applied spatially.

Fig. 7 Motion–compensated, uni–directional prediction.

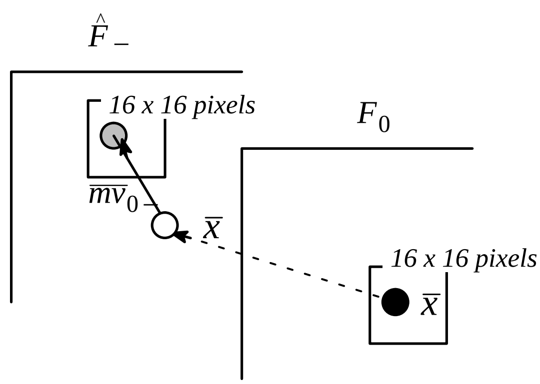
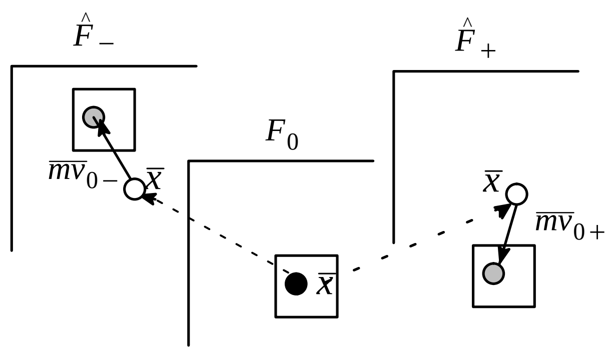
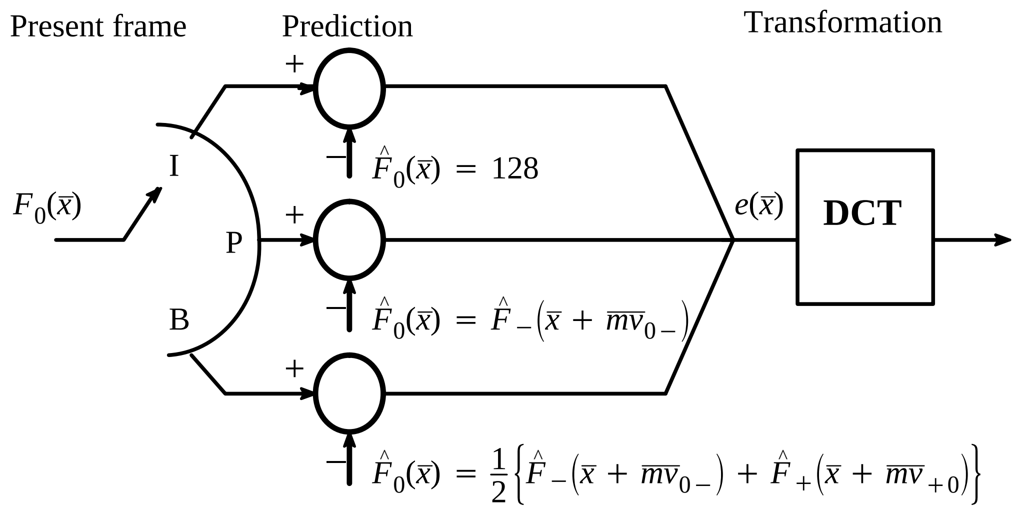


Fig. 8 Motion–compensated, bi–directional prediction.



There are two types of prediction, of which are both motion compensated. The first type of prediction, denoted P, is unidirectional. It is illustrated in Fig. 7. The location $\overline{x}$ refers to a pixel within a block of 16×16 pixels. An area around the same location $\overline{x}$ in a prior encoded and decoded frame $\hat{F}_-$ is searched to find the block that best matches the given block in frame F_0. A common measure for matching that is easily calculated is the sum of the absolute values of the pixel differences in the two blocks.

Fig. 9 The MPEG energy compaction. I stands for intraframe coded (DCT only), P for predicted and B for bi–directionally predicted frames (motion–compensated prediction and DCT).



The motion compensation works well for translational motion, for example resulting from camera pans. It does, however, approximate other types of motion reasonably well if the block size is small compared to the moving area and if the movement is minimal. For instance, a zoom can locally be seen as a radial translation of a block, and a rotation as a translation along the tangent.

The second type of prediction is bi–directional (B frame) and consequently not causal. A block in the frame is estimated, using a frame in the past and one in the future (see Fig. 8). The prediction is delayed until the future frame is present. The bi–directional prediction is more accurate than what is possible for P frames. For instance, uncovered areas that are not visible in the past frame can be properly predicted from the future frame. The bi–directional prediction consequently gives a very high degree of compaction.

Following the prediction the error signal is transformed into blocks of 8×8 pixels to compact the signal spatially. This prediction may be by–passed so that the frame is only transformed (I frame). In this case a bias of 128 is subtracted. The compaction system is shown in Fig. 9 and an illustration of the periodic interleaving of the I, P and B frames is shown in Fig. 10 (a period is called a group of pictures, GOP).

Fig. 10 MPEG's group of picture format. There is only one I–frame per GOP but number of P and B frames can be chosen freely.

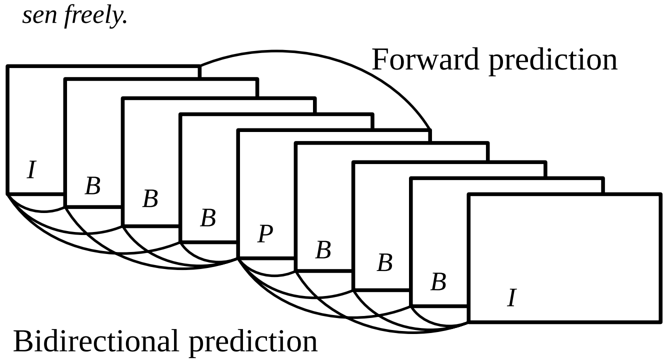
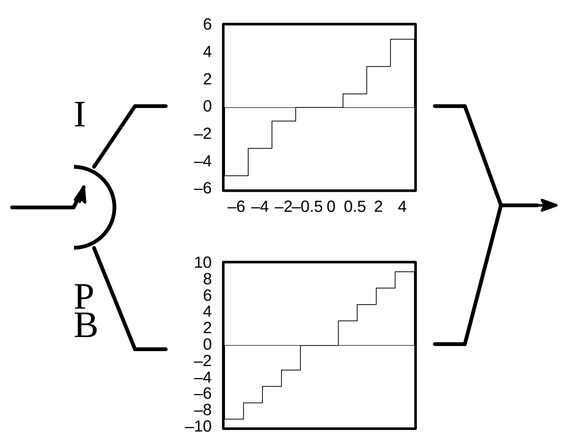


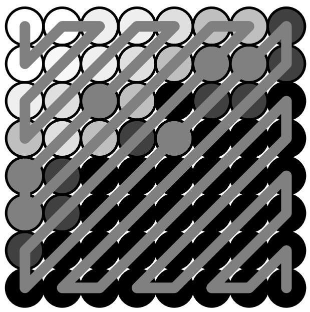
Fig. 11 The MPEG quantizers.



The compacted signal is quantized block by block by one of either two possible stair–case functions: one for blocks from I frames, and the other for blocks from P and B frames (Fig. 11). This is motivated by the differing statistical characteristics of the transformed values. A large number of the values are truncated to zero which constitutes a large percentage of the compression.

The values of each block are scanned out in zigzag order and run–length encoded (Fig. 12). The codewords for the runs and the motion–vectors from the prediction are represented by a Huffman–like code.

Fig. 12 The block of DCT values is scanned in zigzag order and run–length encoded.



Discussion

There is nothing inherent in the coding that must be changed when refitting a system which used synchronous to asynchronous transfer. The rate control affects the quantization levels by either scaling the values before the quantization (more get truncated to zero) or by switching between a set of quantizers with different degrees of coarseness. The quantization is consequently modulated by the rate control, and it may affect the quality consistency. The statistics of the quantized values are affected by the change in quantization but this is usually ignored and the same entropy code is applied irrespectively of the quantizer.

MPEG can, however, offer a further regulation mode of the compression ratio if the GOP format is relaxed: a lower bit rate can be achieved by increasing the number of B frames in relation to P frames, and P frames in relation to I frames.

MPEG as described is poorly suited for asynchronous transfer: the B frames cause long delay and the GOP concept introduces highly variable bit rate with strong periodicity [78]. Modifications to the scheme are discussed in [51]. In simulations, one scheme with only one B frame between any P frames and another with P' frames (predicted from a past frame but not used to code future frames) instead of B frames are found to yield reasonably good trade–off in encoding delay and compression. The periodicity of the I frames is avoided by replacing them with an intraframe coded column in each frame of the GOP.

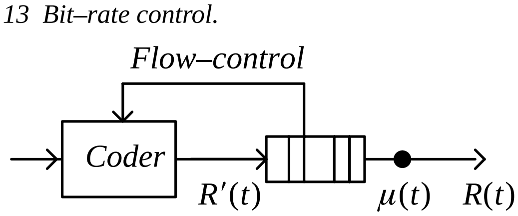
Non–standardized coding schemes are more of interest for proprietary systems (used for surveillance, secure communication etc.). Public communication is after all enabled by whole communities using compatible systems such as the standardized MPEG schemes and the ITU–T H.261 to H.263.

We refer to [61] and [74] for more discussion on coding, and to [92] for an overview of standardization activities (ITU–T Study Group 15 is responsible for video coding on ATM).

Section 9. Rate control

Source coding of video naturally produces a varying bit rate since not all video frames have the same entropy [38][85][89]. The network has to be informed about the particular behavior that may be anticipated from a source in order to offer delay and loss guarantees (call–acceptance control). Two approaches are common: the time–varying rate from the coder is characterized by a suitable stochastic model, or its envelope is given some specified bound. The bound could be stochastic or deterministic, although the latter is most common. The purpose of the rate–control is consequently to enforce the specification of the bit stream.

Fig. 13 Bit–rate control.



The general system is shown in Fig. 13. 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 bit 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 a practical implementation it may be smaller than what is needed to cope with all possible variations. It is therefore common to add a back pressure signal from the buffer to the encoder, so that the compression rate is increased when buffer overflow is at risk.

The flow control changes the quantization which may in turn reduce the quality. The recipient of the video may therefore notice disturbing variations in the quality if there are frequent changes in quantization levels. One of the argument used to promote asynchronous transfer of video has been that rate control would no longer be needed. The benefits would be a less complex encoder, less buffering delay, and a quality that would be constant since the transfer would use whatever capacity is needed for the unrestrained bit stream.

Even though it is possible to allow such ”freewheeling” transfers, it would nevertheless be poor network engineering. Firstly, the provision of constant or—as we prefer to call it—consistent subjective quality still allows some bit–rate regulation. In fact, in [76] it is shown that a fixed quantizer does not result in constant quality since sections of the video signal that are easy to encode should be more coarsely quantized than the average to maintain a uniform quality level. Secondly, it would be difficult to find an accurate and concise description of the uncontrolled bit rate in order to provide quality guarantees. This means that the bit stream from the coder should be regulated even for asynchronous transfers. As pointed out in Section 7, the resource sharing in the network is improved when the flows are smooth.

The issue is thus to reduce the variability of the rate–function $R(t)$ while minimizing the effects on the consistency of the perceptual quality [76][82]. The joint problem of traffic characterization and rate–control is thus to find a suitable description of the bit stream that is sufficiently terse to be useful to the network and that can be enforced without overly throttling the compression rate.

Modeling

We first look at modeling as a means to describe a bit–rate function to the network. There are several attempts to model the variable–rate coded video bit stream $R'(t)$ in the literature, such as references [16] [22] [23] [25] [34] [42] [59] [65] [67] [70] [85] [90] [94] [96]. (Most of these studies combine the modeling with some queuing analysis to determine multiplexing performance. See comment on usefulness of this procedure on page 16.)

There is good evidence that this bit–rate process exhibits long–range behavior [4][27]. This means that accurate characterization seems intractable by most conventional stochastic processes, such as various Markov processes. As discussed in [21], the apparent long–range behavior could also be explained by lack of stationarity. In either case, the modeling is problematic.

Provided that a model has been chosen, the user should then 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 specification (so called *policing*). Most of the models cannot be easily enforced and verified, and are thus of limited value to the network for call–acceptance control.

Heeke describes a model–based characterization that can be enforced without noticeably affecting the video quality [37]. The method forces the bit stream to obey a Markov chain model. The admissible rate will be restricted to a few levels with geometrically distributed holding times. A policing function that verifies the constant–rate levels and their average holding times is given.

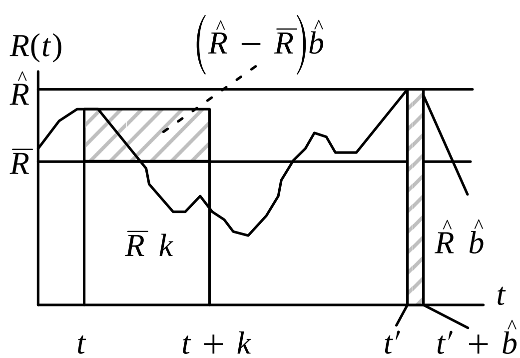
Bounding

A more easily enforceable and verifiable specification of a bit–rate function is to bound its envelope. The simplest is a single value that limits the rate function. The limit may be anywhere in the range of values that $R'(t)$ may assume (usually above the average). The limit, $\hat{R}$, does not have to be fixed for the duration of the session but could be renegotiated when permitted by the network control [13][33]. Several bit–rate control schemes suitable for enforcing a constant rate are presented in the literature [11][52][62][75][101]. The decisions that the algorithms must take are when to apply back–pressure and to what degree.

The leaky bucket is the most commonly suggested form of deterministic bounding with more than a single limit [72]. It is characterized by three parameters: the sustainable and the peak rates ($\bar{R}$ and $\hat{R}$, respectively) and the maximum burst duration ($\hat{b}$). The output rate $R(t)$ from the buffer, measured over some interval at time t, is restricted by

$$\begin{aligned} &R(t) \le \hat{R}, \ \forall t, \ \text{and} \\ &\sum_{i=t}^{t+k} R(i) \le \left(\hat{R} - \bar{R}\right) \hat{b} + \bar{R}\, k, \ \forall t, k. \end{aligned} \qquad (1)$$

Fig. 14 The accumulated bit rate within a window of k time–intervals is bounded by two areas, one fixed and one proportional to k.



The worst behavior of a constrained source is a sustained rate of $\bar{R}$ with a single burst of $\hat{R}\hat{b}$ bits at some instance; an on–off behavior with periodic bursts at peak–rate is also possible. The leaky bucket has been studied for realistic but unregulated video sequences [83]. There is, however, scant evidence that it is the limiting function best suited to the characteristics of video sources. The leaky bucket is however promoted within both the ATM and the IPng communities in order to specify traffic of all types uniformly. That may indeed be more important than having several bounding functions, each with a claimed optimality for a specific type of traffic.

If the coder cannot be regulated, as is the case for playback of stored video, then some smoothing is still possible and the server rate is adjusted according to the slow drift of the buffer occupancy level. The goal is to minimize the probability of buffer overflow and the frequency of rate changes while meeting some prescribed characterization that the network uses for call acceptance. References [58][77] describe how this class of control would work.

The network or receiver may send congestion or flow–control notifications to the sender [8][44]. These messages could be used in order to regulate the service rate, $\mu(t)$. If the restriction on the bit rate persists then it will eventually lead to buffer feedback and increased compression. It is however preferable that the flow and congestion control can be handled within the existing framework of bit–rate control.

A final note on the classifications: constant and variable bit rate (CBR and VBR, respectively). These terms have of course obvious definitions: CBR as constant ($R(t) = \hat{R}, \ \forall t$) and VBR without shaping ($R(t) = R'(t), \ \forall t$). These definitions are however only valid for the extreme points of operation. Moderately shaped bit streams are not well covered by them. We would therefore like to suggest the following definitions.

CBR should be taken to mean constant service rate, $\mu(t) = \mu \Rightarrow R(t) \le \hat{R}, \ \forall t$, and
VBR as the time–varying service rate $\mu(t) \le C_{link} \Rightarrow R(t) \le R'(t), \ \forall t$.

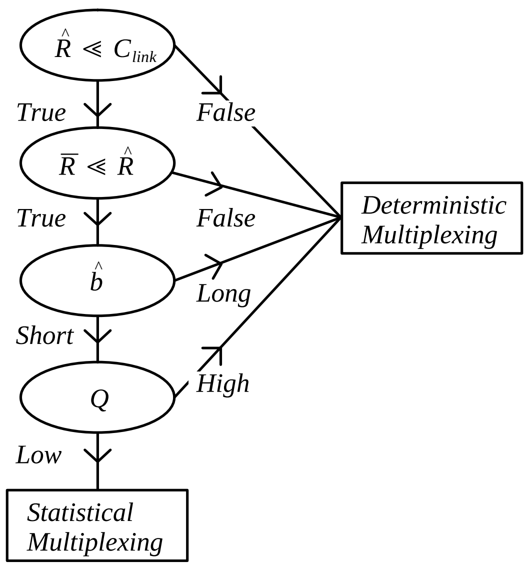
By classifying the bit streams with respect to the service rate we find that the definitions better match the service classes for ATM.

Section 10. Information transfer

The network performs the generic functions of multiplexing and routing in order to transfer the information to the destination. The routing functions, to provide connectivity, are not dependent on the information type in the transfers. Multiplexing, on the contrary, is highly dependent on the requirements imposed by the information type and application context since multiplexing determines much of the transfer quality in the network.

The optimization criteria for the transfer is to minimize the queuing and to maximize the utilization. The importance of a high utilization may, however, be disputable in cabled networks where transmission capacity should become a bulk commodity. A joint optimization is possible if the multiplexed streams are shaped to minimize the temporal variability: remember that an *N*D/D/1* queue (*N* deterministic streams multiplexed) is always shorter than an *M/D/1* queue at equal loading [88].

Fig. 15 Choice of multiplexing mode. The link has capacity C_{link} *and the source has peak–rate* $\hat{R}$*, mean rate* $\overline{R}$*, and max burst length* $\hat{b}$*. The requested quality is denoted Q.*



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. In both cases there is a division of responsibility: 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. ATM and IPng with RSVP break this division by asking the network to provide quality guarantees for statistically multiplexed channels. Quality guarantees have traditionally been accomplished by deterministic multiplexing (synchronous TDM).

For the network provider the interesting issue is the video session's need for quality guarantees and the best way for providing it in the network [57]. Capacity allocation could thus be deterministic or statistic. The choice of multiplexing mode for asynchronous transfers depends on several issues [86][100], as illustrated in Fig. 15. Deterministic multiplexing is the natural choice when the peak–rate is close to the link–rate, when the mean is close to the peak–rate, when traffic bursts are long compared to the buffers in the network, or when the quality requirements are stringent.

We may define three general service classes:

1) Deterministic multiplexing with fixed quality guarantees
2) Statistical multiplexing with probabilistic quality guarantees.
3) Statistical multiplexing without quality guarantees

For ATM, in the terminology of the ITU–T Draft Recommendation I.371, these classes are referred to as ATM–layer bearer capabilities. The listed classes are called deterministic bit rate (DBR), statistical bit rate (SBR) (called constant and variable bit rate by the ATM Forum), and unspecified bit rate (UBR). In the Internet world they correspond to guaranteed, predictive and best–effort service [15], only the latter is offered by IP today (version 4).

Video may be transferred in all of the listed service classes.

Deterministic multiplexing

Deterministic multiplexing means that all flows are (deterministically) bounded and that enough capacity and buffers are reserved in the network to assure complete absence of overflow. The quality that can be guaranteed is therefore an absence of packet or cell loss and bounded maximum delay [55]. The delay variations may also be limited if need be. The case when all queues are first come, first served and the flows are bounded at the network ingress is analyzed in [18]. Tighter bounds on delay and buffer space can be achieved if more general service disciplines are allowed which shape the resultant multiplexed flow (non–work conserving) [32][50], or which maintain the shape of the individual flows through the multiplexing [80][102].

The common objection to deterministic multiplexing is that the reserved capacity is poorly used when only loose bounds such as a peak–rate limit, can be placed on the flows. The traffic control architecture can however be designed so that service classes with statistical multiplexing (*eg*, UBR and SBR) can be offered in addition to the deterministic service. Traffic in the statistical classes can thus expend any slack in the reserved capacity. The issue is therefore one of tariffs: can guaranteed service be offered at nearly equal cost to a statistical bit–rate service, for a fixed level of perceived quality, given that slack in the reserved capacity can be resold to other traffic classes? This question will remain open until the tariff structures are determined.

Statistical multiplexing without quality guarantees

The simplicity of the deterministic multiplexing is equalled by that of statistical multiplexing without quality guarantees. Since the network does not offer any guarantees of the multiplexing performance (reliability guarantees are not considered) the amount and characteristics of the traffic entering is ignored. The perceived quality may therefore vary dramatically depending on the momentary load in the network [7].

Despite the total absence of guarantees, this service class may still be of use for video transfers. The tools *IVS* from INRIA [98], and *nv* from Xerox [26] are used for video conferences over the Internet, often reaching wide audiences via the MBone [24][66]. Many of the applications using video are unidirectional and do not pose stringent delay limits, allowing at least in principle retransmissions to be used in order to reach a specified perceived quality. The quality can also be increased by means of forward error–correction [3], especially in conjunction with spatial dispersion of the traffic to randomize losses [68].

The viability of offering interactive video services at acceptable quality (sufficient for the task at hand) is still unclear. It is determined by the supply of and demand for capacity, as well as the characteristics of the traffic streams. An individual sender may only affect the latter (disregarding the option of transmitting more rarely). This would mean that no bit rate greater than what the quality demands should be used and that voluntary bit–rate regulation is used to smooth the flow.

For ATM, there is a variant defined on the unspecified bit–rate service called available bit rate (ABR). It is basically an unregulated service but the cell–loss ratio will be minimized for sources which obey congestion notifications from the network. Such messages would naturally be used to regulate the service rate of the buffer at the encoder. ABR could be offered with a low amount of reserved capacity which, for instance, could be chosen according to the bit rate needed for the bare necessity of quality.

Statistical multiplexing with quality guarantees

SBR service for ATM and predictive service for IPng are probably what most researchers would consider the traffic classes of choice for video [35][79][87]. A variable bit rate might be more efficiently handled by a statistical capacity allocation, but this is not always the case [36][63][84][97].

In order to offer guarantees, albeit probabilistic ones, a network must know its current inflow of traffic. New flows are allowed entry if they can be guaranteed the quality they request and their characteristics do not violate the quality of already accepted flows. This means that the network has to estimate the multiplexing behavior with a precision that matches the connection with highest quality.

For a predictive service, the procedure is to measure the load on all links in the network and to estimate the amount of capacity that can safely be allocated [15][43]. A call to be added is specified by its peak–rate. If the unused portion of the capacity on all links along a route is above the requested level then the call is accepted. It is subsequently a part of the measurements needed for future requests. The advantage is that all calls are specified uniformly by a single parameter and that the acceptance decision is straight forward. The open issues are how to best estimate the available capacity and to determine what kind of quality guarantees can be offered (currently there are no guarantees suggested for predictive service, only a stated intention to offer acceptable quality).

The SBR bearer capability for ATM is equally simple to formulate. The call–request specifies the traffic that will be sent and states its desired quality. The network control computes the multiplexing performance that is obtained if the call is added to a specific route. If the resultant quality is sufficient for both the requested call and for existing ones, then the call will be accepted over the route. The network subsequently monitors the traffic to verify that indeed the traffic of the new call behaves according to its specification.

The complexities with this procedure should not be underestimated. We have already concluded that traffic characterizations that are not verifiable by the network are useless in the context of call–acceptance control. The multiplexing performance has to be computed for a heterogeneous set of traffic characterizations. It is not sure that all types of video may be characterized by a single model and, even more so by a single set of parameters. In addition video has to share resources with other traffic types within a given service class. This is not what is reported in the literature where independent streams characterized by a single model with an identical set of parameters are multiplexed and the performance is derived.

Bounding offers a method of specifying all sources not only video but also audio, data and mixed ones by a common set of parameters. The most popular is the aforementioned leaky bucket. The parameters for it may howeever be difficult to estimating before the call has commenced. A method for determining traffic parameters by measurements and the associated call–acceptance are presented in [1]. One remaining problem is that multiplexing can make a traffic stream burstier than specified at the network access. This problem may be solved by shaping multiplexed flows or by accepting lower utilization at nodes inside the network.

Discussion

There is a common misconception that video must be given quality guarantees in the network. Many video services are however one–way and do not have any limits on end–to–end delay that go beyond those of data transfers. There is also the possibility that the video application could adapt to various degrees of loss. At the same time it is clear that the coding schemes developed for a synchronous TDM environment do not handle variations in transfer quality well. A video system developed directly for asynchronous transfers is *vic*, developed by McCanne and Jacobson [69].

The quality in ‘‘best effort’’ networks is determined by the amount of capacity and the users’ demand and behavior. The capacity cannot be expected to be lavished on users since an operator would want to economize on the resource. Furthermore, a few ill–behaving users could obliterate the quality for all. The network service for interactive applications is therefore likely to contain some type of quality guarantees, deterministic or statistical. A strong incentive for this is also the operators’ desire to charge users if the provided quality is indeed sufficient for interactive services. With best–effort service there is little premium paid to an operator for a better quality level.

Section 11. Delay

Information is inevitably delayed in networks and, since we consider asynchronous transfers, these delays will not be constant (not even for deterministic multiplexing). This delay has to be considered end–to–end since delay limits are posed by the application. The video signal is delayed as protocol functions are executed and when the signal is transmitted across the network.

The following instances may cause the bulk of the end–to–end delays:

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

The acquisition is the time 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 then the delay can be on the order of a single pixel instance.

The encoding can be structured to minimize delay. In the case of MPEG, the coding can start as soon as 16 lines have been acquired (a stripe). Similarly, each received stripe can be decoded without waiting for consecutive ones, each encoded area of a frame may yield a variable amount of bits. If the bits are transferred at a constant rate then there is no longer a constant delay. In fact, all system functions between the signal acquisition and display can be considered asynchronous with respect to the sampled signal.

The functions closer to the network are the segmentation of service data–units or streams into protocol data–units and their re–assembly. The time to fill a packet or cell might be excessive at low rates (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, to send partially filled cells or packets of restricted length. Re–assembly delay depends on message length and transfer rate. There is for instance no delay for unstructured (stream–oriented) data. Otherwise there may be restrictions on the maximum transfer unit (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 look–ups in hosts and switches (routers). The UNIX process scheduling is a major problem that can be relieved by using operating systems with real–time scheduling, or by taking the operating system out of the information transfer all together. In general, protocols should be implemented to reduce maximum delay and not only to maximize throughput.

Wave propagation is limited by 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 a 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 comes. The transmission delay is restricted by increasing the link capacity and by reducing the number of links per route.

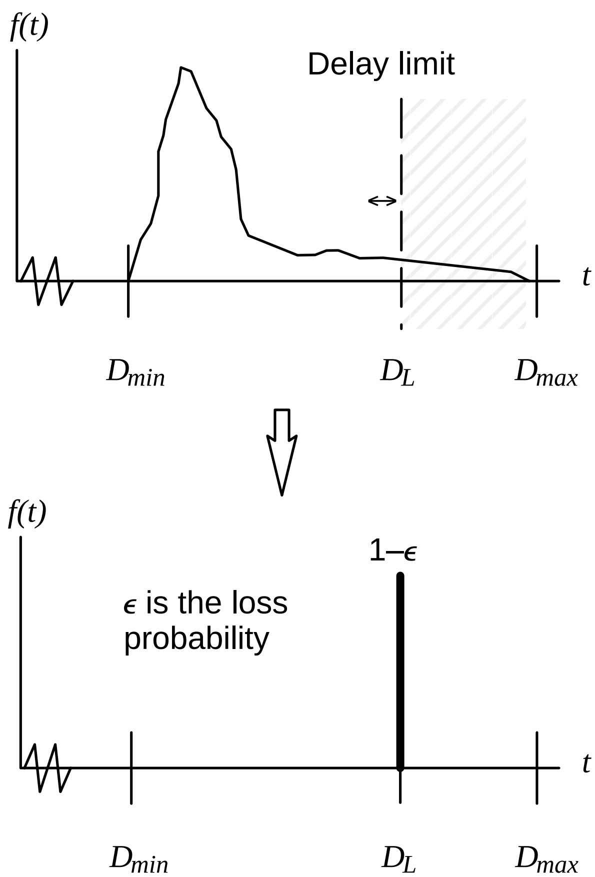
Since the multiplexing is asynchronous there will be queuing in the network. Queuing delays in the network vary dynamically from cell to cell and packet to packet for a given route. The delay depends on the instantaneous load in each multiplexer, number of multiplexing hops on the route, amount of buffer–space per node and whether deterministic or statistical multiplexing is used. The scheduling discipline affects the distribution of the delays. If it is work–conserving (never idle when there are packets to serve) it does not affect the average, however.

Delay control

There are two control issues regarding delay: the variations must be equalized to maintain the isochronal sample–rate, and the absolute value must be limited for interactive applications. The absolute value can only be capped by careful design of all the functions from video source to monitor. Most articles that discuss the delay aspects of video aim at reducing the queuing delay. Note that basically any of the functions listed above can yield delay that exceeds the acceptable level for interactive applications.

It is possible to limit the delay variations (jitter) in the network, or to remove it from an arriving stream at the ATM adaptation layer or the transport layer. The delay can also be equalized at the application layer where the signal is reconstructed. Jitter control in the network comes at a cost of having either a high minimum delay or more complex scheduling. Since not all applications require jitter–free service, jitter removal should preferably be done outside the network when needed.

Fig. 16 The delay is equalized by buffering data up to the acceptable limit. Segments that are delayed by more than the limit are treated as if they were lost.



Equalization at the network interface of the receiver is not sufficient unless all subsequent 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. Ideally, this would be the only instance of it and it is, in fact, the most appropriate location since the bit stream can be synchronized to the display system (the digital–to–analog converter). It should also be noted that each stage of equalization introduces more delay.

Equalization of delay variations is basically done by buffering data delivered by the network to a predetermined limit before delivery (see Fig. 16). Late data is discarded (there is no loss if $D_L \geq D_{max}$). The general problems 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 D_L when displayed.

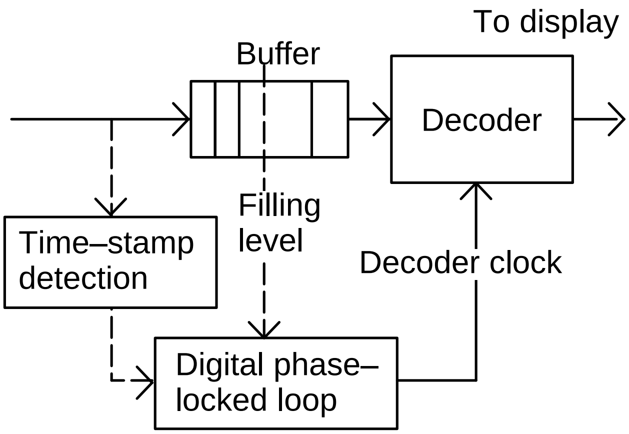
A common simplification is to equalize queuing delay at the re–assembly point (the adaptation layer in case of ATM and at the transport layer in case of IP). Jitter introduced in the end system is then removed before or after the decoding to obtain signal synchronization.

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 but differ in *ppm* values. The jitter is thus 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 by Mills (RFC 1305) [71]. 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 (see Fig. 17). The input signal to the loop can be either time–stamps carried in the cells or packets, or the buffer–fill level [95].

Once the clocks are (sufficiently) synchronized and the data–stream is sent completely isochronally then the delay is equalized by simply reading the application frames from the buffer with the same time–intervals as sent. It is evident that variable–rate video complicates the equalization since 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.

Time–stamps in every cell or packet is therefore needed to mark their generating instances.

Fig. 17 The sender–clock frequency can be estimated by means of a phase–locked loop. It may take the buffer–fill level or explicit time–stamps as input.



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 display's cathode ray.

Synchronization immediately before the display will be needed also for another reason. When several video streams emanating from different 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 time–base [45].

Section 12. Errors and loss

The video communication system causes information loss. First the camera has a limited bandwidth, the analog–to–digital conversion quantizes the amplitude of the samples, and the encoding introduces controlled amounts of distortion (in order to compress the signal). The 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 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 above the acceptable threshold. Undetected loss in a signal can place encoders and decoders out of phase. For forward error–correction this leads to an irrecoverable state. Burst–errors caused by loss of synchronization and by equipment failures have durations 20 – 40 ms. Their likelihood has been estimated to be below 10^{-7} [91]. Loss, especially due to multiplexing overloads, appears to be the most common signal corruption caused by the network.

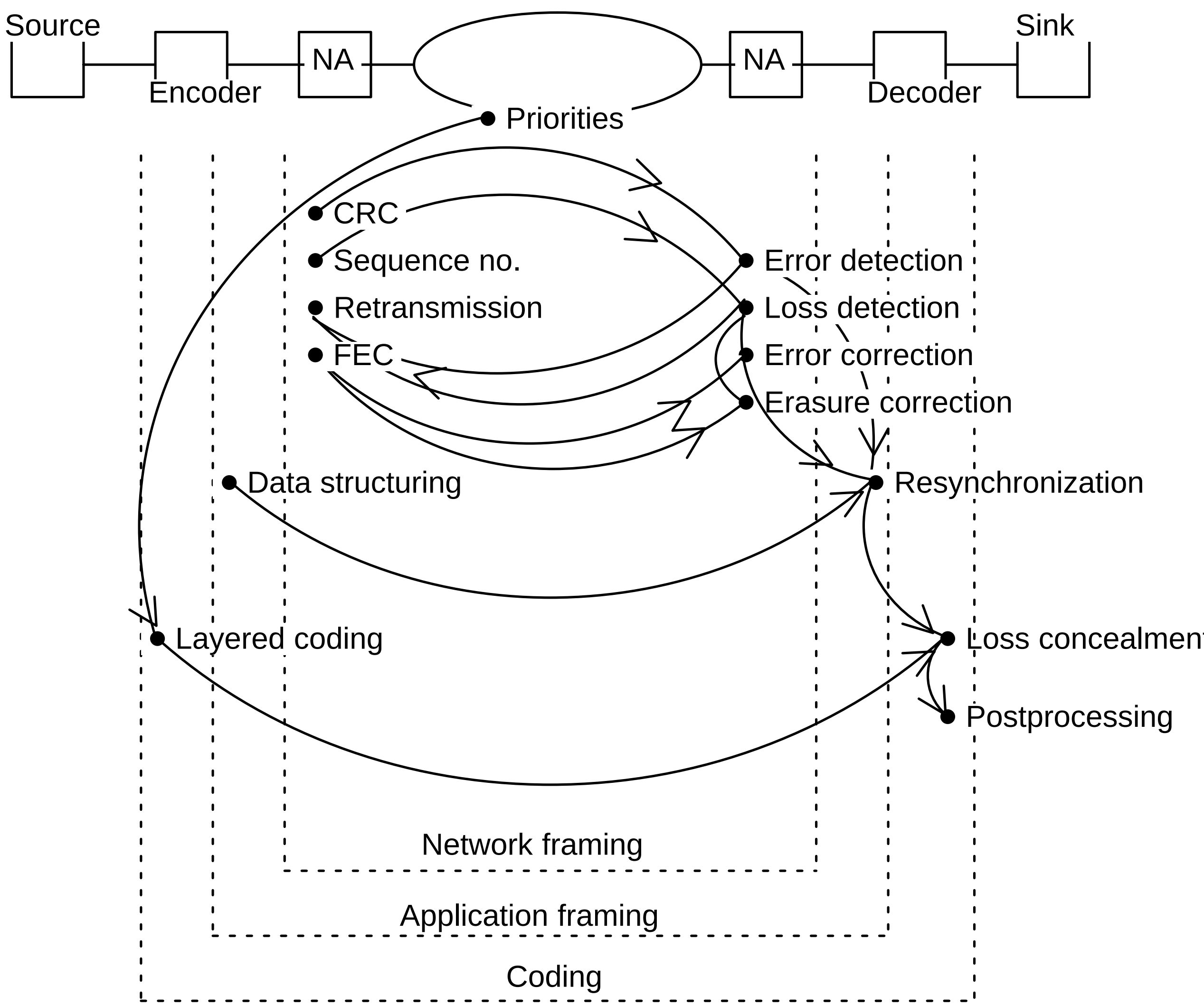
Error recovery is based on limited error propagation, and correction or concealment of the missing portion of the signal. Error propagation is restricted by proper framing of the bit stream so that errors and loss can be detected. Correct signal reception may then be re–initiated at a later point in the bit stream.

Framing

The multiplexing unit in traditional TDM networks is a call (a session). Asynchronously multiplexed networks, such as those based on ATM and IP, have cells and packets, respectively, as multiplexing units which are shorter than a full session. Network framing means that appropriate control information is added to each multiplexing unit. There can also be framing added on segments of the signal at the application layer. These segments are created with knowledge of the signal's syntax and typically they constitute the loss unit (at least one application frame is lost when corruption has occurred). An example of application framing is the MPEG slice layer which packs bits for 16 consecutive lines together.

The purpose of the network framing is to detect and possibly correct lost and corrupted multiplexing units (Fig. 18).

Fig. 18 Error and loss handling.



NA — network adaptation

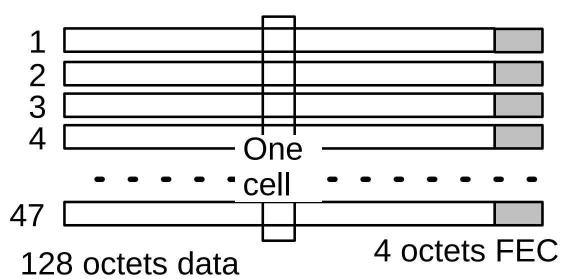
Errors may be detected by a cyclic–redundancy check of sufficient length [91]. Loss is detected by means of sequence numbers which turn it into erasures (known location, unknown values). It is important that the sequence number is based on the number of transferred data–octets. Knowing that a cell or a packet has been lost does not tell how much data it contained [48].

Errors and also loss can be identified by a CRC on the application frame after re–assembly. It is important that frame–length is known a priori since the length of a faulty frame cannot be ascertained. The failed CRC could be caused by a bit–error, which would not affect its length, or by a lost packet or cell. Note that a length–field in the frame cannot be used to tell the two cases apart since it could have been affected by a bit–error (actual length of the frame would not match the value of the length–field even though loss has not occurred).

A lost or corrupted network–frame would, in case of regular data communication, be re–transmitted. There are complications with the use of retransmissions for video: first the delay requirements might not allow it since 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 data–stream must be buffered until the missing frame eventually is received (assuming a selective reject policy). Note that this case is much worse for video than for a data transfer for which flow–control is enforced (eg, at most a full TCP send–window needs buffering).

One may simply discard network frames with bit–errors since it would add little to the loss caused by multiplexing overload. If we disregard retransmission, the question is whether the loss should be corrected by forward error–correction. Since a lost cell or packet results in a long burst of erasures there is a need to interleave the data before the coding. Interleaving disperses the erasures over several code–words so that they hopefully can be corrected. Fig. 19 shows how the interleaving and coding is applied to the bit stream for ATM adaptation layer 1. The matrix is filled row by row and the cells are read out column by column with one octet from each row. The Reed–Salomon code can correct 4 erasures per code–word; each lost cell creates one erasure per code–word.

Fig. 19 Interleaving and forward–error correction for handling cell–loss in ATM. The Reed–Salomon (128, 124) corrects four erasures



There are several reasons to be cautious against forward–error correction 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 (copies of the data can be used for the encoding if structured differently than in Fig. 19 [45]). Third, loss caused by multiplexing overload is likely to be correlated since the overload is caused by traffic bursts and more loss may thus occur than what the code can correct. If an interleaving matrix cannot be corrected then the full matrix is useless (the de–interleaving spreads the erasures) and the loss–situation is in fact made worse. The interleaving matrix could of course be made larger to cope with burst losses but, again, it increases the delay. Fourth, the coding adds overhead.

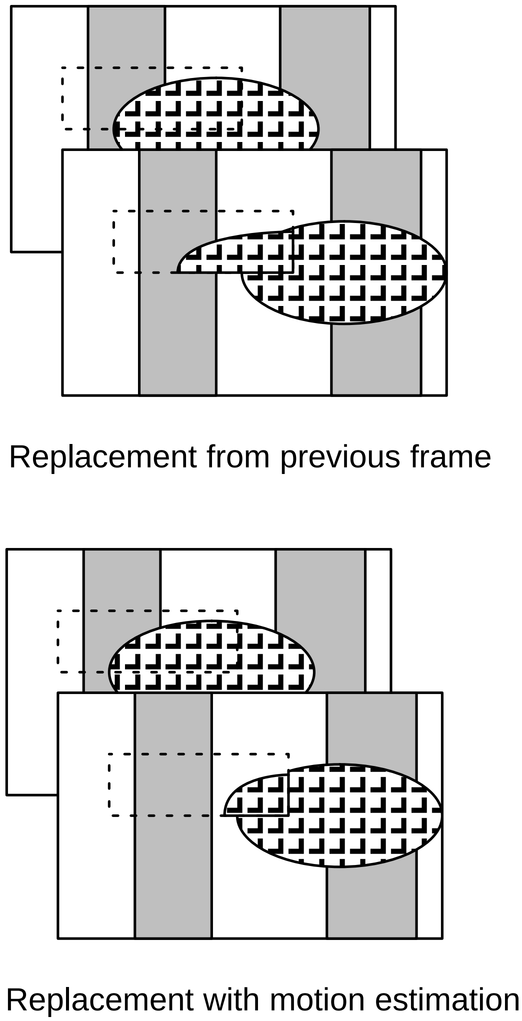
Despite these fully valid points of criticism, the foremost argument against FEC is that it is overly ambitious: absolute delivery is not needed. The user is happy with recovery that renders loss imperceptible. This level of recovery is offered by concealment after the decoding .

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 however 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 codewords are not split by segment boundaries [30]. This can however be guaranteed since the segmentation is done where full information about the signal syntax is available.

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 [29][45][53]. For instance, the corresponding area in the previous frame can be used [64]. (It might be best to repeat the full frame if the corruption is severe.) When the coding uses motion estimation and the motion vectors are correctly received, then 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). These two options are illustrated in Fig. 20. The motion–vectors should be sent separately from the prediction–error for the latter to work.

Fig. 20 Loss concealment.



The residual error after concealment will however propagate into future frames when temporal prediction is used. It will stop once the prediction is restarted, as for a correctly received I frame in the MPEG scheme. That can however be several frames into the future and the visibility of the propagating error may be disturbing. In analogy with the retransmission at the transport layer, the decoder could send a refresh demand to the encoder. The next frame to be coded after the request has been received would then be intraframe coded. Even partial refresh could be possible. How to best cope with error–propagation in conjunction with interframe prediction coding is largely an open problem.

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

Layered (hierarchical) coding

Layered coding means that the signal is separated into components with differing visual importance [10][45][53]. The idea is that each layer can be transferred in the network with a quality that matches its importance. Vital layers may thus be transferred in a class with guaranteed quality (even deterministic guarantees), while a signal layer that enhances the quality could be sent as ‘‘best effort’’. The hope is that the overall transfer is more economical than if the transfer was done over one channel with a service quality determined by the most sensitive part of the information (such as the motion vectors). It is important that the layers are sent along the same route in the network in order to ensure similar delays for them.

Fig. 21 Layering after energy compaction. The layers are formed from the signal components and are independently quantized, variable–length encoded and transferred.

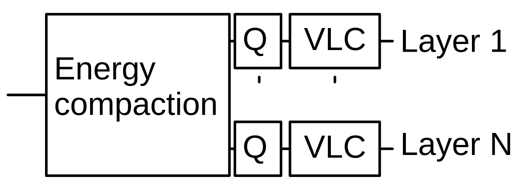
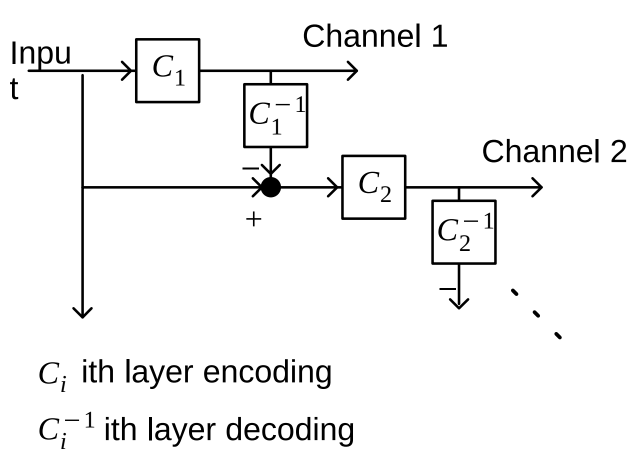


Fig. 22 Iterative coding scheme.



The layering can be done after the energy compaction, which fits well for subband and transform coding (Fig. 21). One or a few subbands are combined to form a layer, or one or a few of the transform values in all blocks of a frame for transform coding. It is also possible to define iterative coding schemes where each layer is coded by a method tailored to its properties, as in Fig. 22. The so–called pyramid scheme is an example of this for which the energy compaction at each level consist of a low–pass filtering followed by subsampling [9].

The objections one may have to the layered coding are that it may be difficult to ensure quality consistency. Remember that perceived quality should primarily be determined by the coding. The average quality is, loosely speaking, given by the basic layer and the consistency will be determined by the enhancement layers. If the enhancement layers incur loss frequently then there will be quality variations.

Another serious reservation is that statistical multiplexing gain improves only slowly with increasing loss probability. This puts efficiency gain in question: enhancement layers may not necessarily be more efficiently multiplexed than the basic layer. Heeke reports a 20% improvement in multiplexing gain for a change in loss–rate from 10^{-9} to 10^{-2} [36]. Such moderate improvements may not be enough to offset the overhead added by the layering.

It is not yet firmly established how to chose quality level in the network for a single layer. How then should the sender match transfer quality to several layers? Finally, layering assumes that a set of connections of differing capacity and quality of service is cheaper than one connection for the aggregate stream. This cannot be firmly established until tariff–structures for broadband networks are in place.

Section 13. ATM and Internet Protocols

The protocols are the placeholders for the functions we have discussed in the previous sections. The pertinent parts of these protocols in the context of video transfer are the end–to–end functions. These belong in the transport and adaptation layers for IP and ATM, respectively. These protocol functions are what we collectively have referred to as 'network framing'. Application framing is, by definition, part of the application.

ATM adaptation layers

The role of ATM adaptation is to ameliorate the behavior of the network for the communication applications. The main adaptation functions are segmentation and re–assembly of user–data, and detection of bit and burst errors and of cell losses. There are additional functions suggested for adaptation layers, namely delay equalization and cell–loss recovery.

The formats of the protocol information for adaptation layers 1 and 5 are shown in Fig. 23 (ITU–T Rec. I.363). Layer 1 is aimed at constant–rate real–time services. It supports transfer of partially filled cells in the, so called, P format. The sequence number has three bits which may detect seven or less contiguously lost cells. It is protected by four bits with the possibility of correcting a single bit–error (even though such a rare event could be treated as cell loss). Detected cell–losses become erasures since all cells are filled to the same level (not necessarily 47 octets).

Fig. 23 Formats for control information in ATM Adaptation Layer 1 (a) and 5 (b).

a)

CSI	SN	SNP	*Header*
1	*3*	*4*	*[bits]*

CSI: Convergence Sublayer Identifier
SN: Sequence Number
SNP: Sequence Number Protection

b)

UU	CPI	Length	CRC	*Trailer*
1	*1*	*2*	*4*	*[octets]*

UU: Common Part Convergence Sublayer User–to–User Indication (CPCS–UU)
CPI: Common Part Indication

The CS–indication bit may be used to send residual time–stamps, which may be used for synchronization of the sender and receiver clocks in relation to the network clock. The synchronization would be used to remove jitter and to regain the original pacing of the stream. Forward error–

correction with Reed–Salomon (124, 128) coding may be used with interleaving to correct at most four erasures out of 128 cells. The interleaving creates delay of 128 cells (36 ms at 1.5 Mb/s) and the coding adds 3 percent load. It may however give insufficient protection when losses occur in bursts. The de–interleaving turns in practice non–correctable erasures into octet–long burst–errors of rate $n/124$ (for $n > 4$) over the block of 5828 octets.

Layer 5 is frame–based to suit data communication. Each frame is protected by a 32–bit CRC for error detection. A length field gives the amount of user data in the protocol data unit, and may help detect cell loss. The convergence sublayer has to collect a specific amount of octets into a frame if data is delivered in a stream. The longer the frame is, the lower the overhead but the higher the delay at the receiver where the full frame has to be reassembled for verification by the CRC.

The maximal data loss resulting from single bit error is one frame; two frames may be lost for a (short) burst of errors and a single cell loss (when the trailer of the frame is lost). There is no way of telling how much user data has been affected once a bit or burst error, or cell loss has been detected (unless all frames are of the same length). ITU–T Rec. J.82 contains the specifications of AAL 1 and 5 for carrying MPEG–2 transport streams at fixed bit rate.

There is an AAL 2 which is designated to support variable–rate coded video. Current discussions seem to start from AAL 1 and refit it for more general transfers. At this point there are no definite suggestions published. A simplified adaptation layer for video transfers has been proposed in [48].

Internet transport layer

Video is transferred in internets by means of the user datagram protocol (UDP). It is an unreliable protocol that at most could be used to verify the frame; it contains an optional 16–bit check sum and a 16–bit length field. UDP can be supplemented by application specific framing, as discussed in [14][69], or by the real–time transport protocol (RTP).

RTP is being standardized within the IETF audio–video transport working group (AVT WG). The functions that it can perform are re–sequencing, loss detection, jitter removal, clock synchronization, and intermedia synchronization. RTP does not guarantee quality of service for real–time services. Signaling for resource reservation in IPng is supposed to be done by RSVP [103]. The information transfer is augmented by a control protocol (RTCP). Note that RTP does not mandate a RTCP or UDP as an underlaying transport layer, and could equally well be combined with ATM AAL 5, for instance.

Section 14. Final remarks

We have in this introductory paper tried to outline the issues involved when video is to be carried in an asynchronous network. The research on the various signal processing and networking aspects of asynchronous transfer of video does not lack prolificacy. However, parts of it lack a certain engineering realism, and despite much work not all of it can be put into practice. For instance, the only standardized system is for fixed–rate coding by means of ISO MPEG–2 carried in ATM AAL 1 or 5 according to ITU–T Rec. J.82. A comparable set of standards for variable bit–rate video is still missing. There are some working systems available in the Internet but their usefulness is hampered by their low quality.

Asynchronous transfer of video is an exciting field that unites the fields of networking and digital signal processing. The possibility to obtain results that are widely appreciated should now be greater than ever, considering the high number of experimental broadband networks available throughout the world [5][17][41][47].

Section 15. Acknowledgments

The author is grateful for proof–reading and helpful comments by Ian Marsh, guest researcher at SICS, and Eric Lin at the Royal Institute of Technology.

Section 16. References

[1] S. Abe and T. Soumiya, ‘‘A Traffic Control Method for Service Quality Assurance in an ATM Network,’’ IEEE Journal on Selected Areas in Communications, Vol. 12, No. 2, February 1994, pp. 322–331

[2] R. T. Apteker, et alii, ‘‘Video Acceptability and Frame Rate,’’ IEEE Multimedia, Vol. 2, No. 3, Fall 1995, pp. 32–40.

[3] E. Ayanoglu, *et alii*, ”Performance Improvement in Broadband Networks Using Forward Error Correction for Lost Packet Recovery”, Journal of High–Speed Networks, Vol. 2, No. 3, 1993, pp. 287–303.

[4] J. Beran, *et alii*, ‘‘Long–Range Dependence in Variable–Bit–Rate Video Traffic, IEEE Transactions on Communications, Vol. 43, No. 2/3/4, February/March/April 1995, pp. 1566–1579.

[5] E. W. Biersack, *et alii*, ”Gigabit networking research at Bellcore,” IEEE Network, vol. 5, March 1992, pp. 42–48.

[6] D. E. Blahut, *et alii*, ‘‘Interactive Television,’’ Proceedings of IEEE, Vol. 83, No. 7, July 1995, pp. 1071– 1085.

[7] J–C Bolot, ‘‘End–to–End Packet Delay and Loss Behavior in the Internet,’’ ACM Computer Communication Review, Vol. 23, No. 4, October 1993, pp. 289–298.

[8] J–C Bolot, *et alii*, ‘‘Scalable Feedback Control for Multicast Video Distribution in the Internet,’’ ACM Computer Communications Review, Vol. 24, No. 4, October 1994, pp. 58–67.

[9] P. J. Burt and E. H. Adelson, ‘‘ The Laplacian Pyramid as a Compact Image Code,’’ IEEE Transactions on communications, Vol. 31, No. 4, April 1983, pp. 532–540.

[10] T. Chiang and D. Anastassiou, ‘‘Hierarchical Coding of Digital Television,’’ IEEE Communications Magazine, Vol. 32, No. 5, May 1994, pp. 38–45.

[11] C–T Chien and A. Wong, ‘‘A Self–Governing Rate Buffer Control Strategy for Pseudoconstant Bit Rate Video Coding,’’ IEEE Transactions on Image Processing, Vol. 2, No. 1, January 1993, pp. 50–59.

[12] C. Cavé, *et alii*, ‘‘Perception of Sound–Image Synchrony in Cinematographic Conditions,’’ in *Proceedings of the Fourth Workshop on Rhythm Perception and Production*, Bourges, France, June 1992.

[13] S. Chong, *et alii*, ‘‘Predictive Dynamic Bandwidth Allocation for Efficient Transport of Real–Time VBR Video over ATM,’’ IEEE Journal on Selected Areas in Communications, Vol. 13, No. 1, January 1995, pp. 12–23.

[14] D. D. Clark and D. L. Tennenhouse, ‘‘Architectural Considerations for a New Generation of Protocols, ACM Computer Communication Review, Vol. 20, No. 4, September 1990, pp. 200–208.

[15] D. D. Clark, *et alii*, ‘‘Supporting Real–Time Applications in an Integrated Services Packet Network: Architecture and Mechanism,’’ ACM Computer Communication Review, Vol. 22, No. 4, October 1992, pp. 14–26.

[16] D. M. Cohen and D. P. Heyman, ‘‘Performance Modeling of Video Teleconferencing in ATM Networks,’’ IEEE Transactions on Circuits and Systems for Video Technology, Vol. 3, No. 6, December 1993, pp. 408–420.

[17] J. R. Cox, Jr., ”Project Zeus,” IEEE Network, vol. 7, March 1993, pp. 20–30.

[18] R. Cruz, ‘‘A Calculus for Network Delay, Part I: Network Elements in Isolation,’’ IEEE Transaction on Information Theory, Vol. 37, No. 1, January 1991, pp. 114–131, and ‘‘A Calculus for Network Delay, Part II: Network Analysis,’’ *Ibidem*, pp. 132–141.

[19] D. Deloddere *et alii*, ‘‘Interactive Video on Demand,’’ IEEE Communications Magazine, Vol. 32, No. 5, May 1994, pp. 82–88.

[20] S. Dixit and P. Skelly, ‘‘MPEG–2 over ATM for Video Dial Tone Networks: Issues and Strategies,’’ IEEE Network, September/October 1995, pp. 30–40.

[21] N. G. Duffield, *et alii*, ‘‘Predicting Quality of Service for Traffic with Long–Range Fluctuations,’’ in *Proceedings of IEEE International Conference on Communications*, Seattle, Washington, USA, June 18–22, 1995.

[22] A. Elwalid, *et alii*, ‘‘Fundamental Results on the Performance of ATM Multiplexers with Applications to Video Teleconferencing,’’ ACM Performance Evaluation Review, Vol. 23, No. 1, May 1995, pp. 86–97.

[23] A. Elwalid, *et alii*, ‘‘Fundamental Bounds and Approximations for ATM Multiplexers with Applications to Video Teleconferencing,’’ IEEE Journal on Selected Areas in Communications, Vol. 13, No. 6, August 1995, pp. 1004–1016.

[24] H. Eriksson, ‘‘MBONE: The Multicast Backbone,’’ Communications of the ACM, Vol. 37, No. 8, August 1994, pp. 54 60.

[25] M. R. Frater, *et alii*, ‘‘A New Statistical Model for Traffic Generated by VBR Coders for Television on the Broadband ISDN,’’ IEEE Transactions on Circuits and Systems for Video Technology, Vol. 4, No. 6, December 1994, pp. 521–526.

[26] R. Fredrick, ‘‘Experiences with real–time software video compression,’’ in *Proceedings of Sixth International Workshop on Packet Video*, Portland, OR, Sept., 1994, pp. F1.1–4.

[27] M. W. Garrett and W. Willinger, ‘‘Analysis, Modeling and Generation of Self–Similar VBR Video Traffic,’’ ACM Computer Communications Review, Vol. 24, No. 4, October 1994, pp. 269–280.

[28] M. W. Garrett and M. Vetterli, ‘‘Joint Source Channel Coding of Statistically Multiplexed Real–Time Services on Packet Networks,’’ IEEE/ACM Transactions on Networking, Vol. 1, No. 1, February 1993, pp. 71–79.

[29] M. Ghanbari and V. Seferidis, ‘‘Cell–Loss Concealment in ATM Video Codecs,’’ IEEE Transactions on Circuits and Systems for Video Technology, Vol. 3, No. 3, June 1993, pp. 238–247.

[30] M. Ghanbari and C. J. Hughes, ‘‘Packing Coded Video Signals into ATM Cells,’’ IEEE/ACM Transactions on Networking, Vol. 1, No. 5, October 1993, pp. 505–509.

[31] M. Goldstein, *et alii*, ‘‘Cell Loss Effects on Perceived Audio/Video Quality and Quality of Service,’’ in *Proceedings of Sixth International Workshop on Packet Video*, Portland, OR, September 26–27, 1994, pp. I3.1–4.

[32] S. J. Golestani, ‘‘A Stop–and–Go Queueing Framework for Congestion Management,’’ ACM Computer Communications Review, Vol. 20, No. 4, September 1990, pp. 8–18.

[33] M. Grossglauser, *et alii*, ‘‘RCBR: A Simple and Efficient Service for Multiple Time–Scale Traffic,’’ ACM Computer Communications Review, Vol. 25, No.4, October 1995, pp. 219–230.

[34] R. Grünenfelder, *et alii*, ‘‘Characterization of Video Codecs as Autoregressive Moving Average Processes and Related Queueing System Performance,’’ IEEE Journal on Selected Areas in Communications, Vol. 9, No. 3, April 1991, pp 284–293.

[35] B. G. Haskell, ‘‘Buffer and Channel Sharing by Several Interframe Picturephone Coders,’’ Bell Systems Technical Journal, Vol. 51, No. 1, January 1972, pp. 261–289.

[36] H. Heeke, ‘‘Statistical Multiplexing Gain for Variable Bit Rate Video Codecs in ATM Networks,’’ International Journal of Digital and Analog Communication System, Vol. 4, 1991, pp. 261–268.

[37] H. Heeke, ‘‘A Traffic–Control Algorithm for ATM Networks,’’ IEEE Transactions on Circuits and Systems for Video Technology, Vol. 3, No. 3, June 1993, pp. 182–189.

[38] D. P. Heyman, *et alii*, ‘‘Statistical Analysis and Simulation of Video Teleconference Traffic in ATM Networks,’’ IEEE Transactions on Circuits and Systems for Video Technology, Vol. 2, No. 1, March 1992, pp. 49–59.

[39] C. J. Hughes, *et alii*, ‘‘Modeling and Subjective Assessment of Cell Discard in ATM Video,’’

IEEE Transactions on Image Processing, Vol. 2, No. 2, April 1993, pp. 212–222.

[40] S. Iai and N. Kitawaki, ‘‘Effects of Cell Loss on Picture Quality in ATM Networks,’’ Electronics and Communications in Japan, Part 1, Vol. 75, No. 10, pp. 30–41.

[41] IEEE Computer, ”Gigabit network testbeds,” Special Report, September 1990.

[42] B. Jabbari, *et alii*, ‘‘Statistical Characterization and Block–Based Modelling of Motion–Adaptive Coded Video,’’ IEEE Transactions on Circuits and Systems for Video Technology, Vol. 3, No. 3, June 1993, pp. 199

[43] S. Jamin, *et alii*, ‘‘A Measurement–based Admission Control Algorithm for Integrated Services Packet Network,’’ ACM Computer Communications Review, Vol. 25, No.4, October 1995, pp. 2–13.

[44] H. Kanakia, *et alii*, ‘‘An Adaptive Congestion Control Scheme for Real–Time Packet Video Transport,’’ ACM Computer Communication Review, Vol. 23, No. 4, October 1993, pp. 20–31.

[45] G. Karlsson and M. Vetterli, ‘‘Packet Video and Its integration Into the Network Architecture,’’ IEEE Journal on Selected Areas in Communications, Vol. 7, No. 5, June 1989, pp. 739–751.

[46] G. Karlsson and M. Vetterli, ‘‘Sub–band coding of video for packet networks,’’ Optical Engineering, Vol. 27, No. 7, July 1988, pp. 574–586.

[47] G. Karlsson, ‘‘Stockholm Gigabit Network,’’ in *Proceedings of the First IEEE Symposium on Global Data Networking*, Cairo, Egypt, December 12–15, 1993, pp. 7–14.

[48] G. Karlsson, ‘‘ATM Adaptation for Video,’’ in *Proceedings of Sixth International Workshop on Packet Video*, Portland, OR, September 26–27, 1994, pp. E3.1–5.

[49] G. Karlsson and G. Djuknic, ‘The Role of Video in Broadband Communications: Enabler and Enhancer of Applications, or Hype?,’’ in *Proceedings of the Second International Workshop on Community Networking: Integrated Multimedia Services to the Home*, Princeton, New Jersey, June 20–22, 1995, pp. 163–170.

[50] G. Karlsson, ‘‘Capacity Reservation in ATM Networks,’’ Accepted for publication: Computer Communications, Special issue on Algorithms for ATM Networks (Ed. B. Rajagopalan), July 25, 1995.

[51] M. Kawashima, *et alii*, ‘‘Adaptation of the MPEG Video–Coding Algorithm to Network Applications,’’ IEEE Transactions on Circuits and Systems for Video Technology, Vol. 3, No. 4, August 1993, pp. 261–269.

[52] G. Keesman, *et alii*, ‘‘Bit–Rate Control for MPEG Encoders,’’ Signal Processing: Image Communication, Vol. 6, 1995, 545–560.

[53] L. H. Kieu and K. N. Ngan, ‘‘Cell–Loss Concealment Techniques for Layered Video Codecs in an ATM Network,’’ IEEE Transactions on Image Processing, Vol. 3, No. 5, September 1994, pp. 666–676.

[54] J. B. Kim et al., ‘‘International Standardization of B–ISDN,’’ Computer Networks and ISDN Systems, Vol. 27, No. 1, October 1994, pp. 5–27.

[55] E. W. Knightly, *et alii*, ‘‘Fundamental Limits and Tradeoffs of Providing Deterministic Guarantees to VBR Video Traffic,’’ ACM Performance Evaluation Review, Vol. 23, No. 1, May 1995, pp. 98–107.

[56] T. Kurita, *et alii*, ‘‘Effects of Transmission Delay in Audiovisual Communcations,’’ Electronics and Communications in Japan, Part 1, Vol. 77, No. 3, 1994, pp. 63–74.

[57] J. Kurose, ‘‘Open Issues and Challenges in Providing Quality of Service Guarantees in High–Speed Networks,’’ ACM Computer Communication Review, Vol. 23, No. 1, January 1993, pp. 6–15.

[58] S. S. Lam, *et alii*, ‘‘An Algorithm for Lossless Smoothing of MPEG Video,’’ ACM Computer Communications Review, Vol. 24, No. 4, October 1994, pp. 281–294.

[59] A. A. Lazar, *et alii*, ''Modeling Video Sources for Real–Time Scheduling,'' Multimedia Systems, Vol. 1, 1994, pp. 253–266.

[60] D. J. Le Gall, ''The MPEG video compression algorithm,'' Signal Processing: Image Communication, Vol. 4, 1992, pp. 129–140.

[61] J–P Leduc, **Digital Moving Pictures–Coding and Transmission on ATM Networks**, Advances in Image Communcation, Vol. 3, Elsevier, 1994.

[62] J–P Leduc, ''Bit–Rate Control for Digital TV and HDTV Codecs,'' Signal Processing: Image Communication, Vol. 6, 1994, pp. 25–45.

[63] J–P Leduc, ''Multiplexing Digital Television Sources on ATM Networks,'' Signal Processing: Image Communication, Vol. 6, 1994, pp. 435–462.

[64] J. Lee and B. W. Dickinson, ''Temporally Adaptive Motion Interpolation Exploiting Temporal Masking in Visual Perception,'' IEEE Transactions on Image Processing, vol. 3, No. 5, September 1994, pp. 513–526.

[65] D. M. Lucantoni, *et alii*, ''Methods for Performance Evaluation of VBR Video Traffic Models,'' IEEE/ACM Transactions on Networking, Vol. 2, No. 2, April 1994, pp. 176–180.

[66] M. R. Macedonia and D. P. Brutzman, ''MBone Provides Audio and Video Across the Internet,'' IEEE Computer, April 1994, pp. 30–36.

[67] B. Maglaris, *et alii*, ''Performance Models of Statistical Multiplexing in Packet Video Communications,'' IEEE Transactions on Communications, Vol. COM–36, No. 7, July 1988, pp. 834–844.

[68] N. F. Maxemchuck, ''Dispersity Routing in High–Speed Networks,'' Computer Networks and ISDN Systems, Vol. 25, 1993, pp. 645–661.

[69] S. McCanne and V. Jacobson, ''vic: A Flexible Framework for Packet Video,'' in *Proceedings of ACM Multimedia*, San Francisco, CA, November 5–9, 1995, pp. 511–522.

[70] D. L. McLaren and D. T. Nguyen, ''Variable Bit–Rate Source Modelling of ATM–Based Video Services,'' Signal Processing: Image Communication, Vol. 4, 1992, pp. 233–244.

[71] D. L. Mills, ''Improved Algorithms for Synchronizing Computer Network Clocks,'' IEEE/ACM Transactions on Networking, vol. 3, No. 3, June 1995, pp. 245–254.

[72] G. Niestegge, ''The 'Leaky–Bucket' Policing Method in the ATM (Asynchronous Transfer Mode) Network,'' International Journal of Digital and Analog Communcation Systems, Vol. 3, 1990, pp. 187–197.

[73] Y. Ninomiya, ''High Definition Television Systems,'' Proceedings of IEEE, Vol. 83, No. 7, July 1995, pp. 1086–1093.

[74] N. Ohta, **Packet Video: Modeling and Signal Processing**, Artech House, 1994.

[75] A. Ortega, *et alii*, ''Optimal Trellis–Based Buffered Compression and Fast Approximation,'' IEEE Transactions on Image Processing, Vol. 3, No. 1, January 1994, pp. 26–39.

[76] A. Ortega, *et alii*, ''Rate Constraints for Video Transmission over ATM Networks Based on Joint Source/Network Criteria,'' Annales des Télécommunications, Vol. 50, No. 7–8, 1995, pp. 603–616.

[77] T. Ott, *et alii*, ''A scheme for Smoothing Delay–Sensitive Traffic Offered to ATM Networks,'' in *Proceedings of the IEEE INFOCOM*, Florence, Italy, May 6–8, 1992, pp. 776–785.

[78] P. Pancha and M. El Zarki, ''MPEG Coding for Variable Bit Rate Video Transmission,'' IEEE Communications Magazine, Vol. 32, No. 5, May 1994, pp. 54–66.

[79] P. Pancha and M. El Zarki, ''Bandwidth–Allocation Schemes for Variable–Bit–Rate MPEG Sources in ATM Networks,'' IEEE Transactions on Circuits and Systems for Video Technology, Vol. 3, No. 3, June 1993, pp. 190

[80] A. K. Parekh and R. G. Gallager, ''A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The

Single–Node Case,'' IEEE/ACM Transactions on Networking, Vol. 1, No. 3, June 1993, pp. 344–357, and ''A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Multiple Node Case,'' *Ibidem*, Vol. 2, No. 2, April 1994, pp. 137–150.

[81] E. Petajan, ''The HDTV Grand Alliance System,'' Proceedings of IEEE, Vol. 83, No. 7, July 1995, pp. 1094–1105.

[82] M. R. Pickering and J. F. Arnold, ''A Perceptually Efficient VBR Rate Control Algorithm,'' IEEE Transactions on Image Processing, vol. 3, No. 5, September 1994, pp. 527–531.

[83] E. P. Rathgeb, ''Policing of Realistic VBR Video Traffic in an ATM Network,'' International Journal of Digital and Analog Communication System, Vol. 6, 1993, pp. 213–226.

[84] A. R. Reibman and A. W. Berger, ''Traffic Descriptors for VBR Video Teleconferencing over ATM Networks,'' IEEE/ACM Transactions on Networking, Vol. 3, No. 3, June 1995, pp. 329–339.

[85] D. Reininger, *et alii*, ''Variable Bit Rate MPEG Video: Characteristics, Modeling and Multiplexing,'' in *Proceeding of the ITC–14*, Elsevier, 1994, pp. 295–306.

[86] J. W. Roberts, ''Variable–Bit–Rate Traffic Control in B–ISDN,'' IEEE Communications Magazine, Vol. 29, No. 9, September 1991, pp. 50–56.

[87] J. W. Roberts, *et alii*, ''Network Performance Considerations in the Design of VBR Codec,'' in *Proceeding of the ITC–13*, Elsevier, 1991, pp. 77–82.

[88] J. W. Roberts and J. T. Virtamo, ''The Superposition of Periodic Cell Arrival Streams in an ATM Multiplexer,'' IEEE Transactions on communications, Vol. 39, No. 2, February 1991, pp. 298–303.

[89] R. M. Rodriguez–Dagnino, *et alii*, ''Prediction of Bit Rate Sequences of Encoded Video Signals,'' IEEE Journal on Selected Areas in Communications, Vol. 9, No. 3, April 1991, pp 305–314.

[90] O. Rose and M. R. Frater, ''A Comparison of Models for VBR Video Traffic Sources in B–ISDN,'' in Proceedings of IFIP TC6 International Conference on Broadband Communciations, Paris 1994, North–Holland, pp. 275–287.

[91] J. M. Simmons and R. G. Gallager, ''Design of Error Detection Scheme for Class C Service in ATM,'' IEEE/ACM Trans. on Networking, Vol. 2, No. 1, Feb. 1994, pp. 80–88.

[92] R. Schäfer and T. Sikora, ''Digital Video Coding Standards and Their Role in Video Communication,'' Proceedings of IEEE, Vol. 83, No. 6, June 1995, pp. 907–924.

[93] N. B. Seitz, *et alii*, ''User–Oriented Measures of Telecommunication Quality,'' IEEE Communications Magazine, Vol. 32, No. 1, January 1994, pp. 56–66.

[94] C. Shim, *et alii*, ''Modeling and Call Admission Control Algorithm of Variable Bit Rate Video in ATM Networks,'' IEEE Journal on Selected Areas in Communications, Vol. 12, No. 2, February 1994, pp. 332 – 344

[95] R. P. Singh *et alii*, ''Jitter and Clock Recovery for Periodic Traffic in Broadband Packet Networks,'' IEEE Trans. on Communications, Vol. 42, No. 5, May 1994, pp. 2189–2196.

[96] P. Skelly, *et alii*, ''A Histogram–Based Model for Video Traffic Behavior in an ATM Multiplexer,'' IEEE/ACM Transactions on Networking, Vol. 1, No. 4, August 1993, pp. 446–459.

[97] T. Tanaka, *et alii*, ''A Study on Comparison Between VBR and CBR Video Service in ATM Environment,'' *Proceeding of the IEEE International Conference on Communications (ICC)*, 1992, pp. 551–555.

[98] T. Turletti, ''H.261 software codec for videoconferencing over the Internet,'' INRIA Research report 1834, France, January 1993.

[99] W. Verbiest, *et alii*, ''The Impact of the ATM Concept on Video Coding,'' IEEE Journal on Se-

lected Areas in Communications, Vol. 6, No. 9, December 1988, pp. 1623–1632.

[100] G. M. Woodruff and R. Kositpaiboon, ‘‘Multimedia Traffic Management Principles for Guaranteed ATM Network Performance,’’ IEEE Journal on Selected Areas in Communications, Vol. 8, No. 3, April 1990, pp. 437–446.

[101] J. Zdepski, *et alii*, ‘‘Statistically Based Buffer Control Policies for Constant Rate Transmission of Compressed Digital Video,’’ IEEE Transactions on Communications, Vol. 39, No. 6, June 1991, pp. 947–957.

[102] H. Zhang, ‘‘Service Disciplines for Guaranteed Performance Service in Packet–Switching Networks,’’ Proceedings of the IEEE, Vol. 83, No. 10, October 1995, pp. 1374–1396.

[103] L. Zhang, *et alii*, ‘‘RSVP: A New Resource Reservation Protocol,’’ IEEE Networks, September 1993, pp. 8–18.

[104] Y–Q Zhang, *et alii*, ‘‘Variable Bit–Rate Video Transmission in the Broadband ISDN Environment,’’ Proceedings of IEEE, Vol. 79, No. 2, February 1991, pp. 214–222.