Packet Video and Its Integration into the Network Architecture

GUNNAR KARLSSON STUDENT MEMBER, IEEE, AND MARTIN VETTERLI, MEMBER, IEEE

Abstract—**Packet video is investigated from a systems point of view. The most important issues to its transmission are identified and studied in the context of a layered network architecture model. This leads to a better understanding of the interactions between network and signal handling. The functions at a particular layer can thereby be made least dependent on network implementation and signal format. In the model, the higher layers provide format conversion, hierarchical source coding, error recovery, resynchronization, cost/quality arbitration, session setup and tear-down, packetization, and multiplexing. Provisions from the network layers pertain mainly to real-time transmission. Special consideration is given to hierarchical source coding, error recovery, statistical behavior, and timing aspects. Simulation results are presented for a subband coding system.**

I. Introduction

TODAY'S fiber technology offers a transmission capacity that can easily handle high bit rates like those required for video transmission. This can lead to the development of networks which truly integrate all types of information services. By basing such a network on packet switching, the services (video, voice, and data) can be dealt with in a common format. Packet switching is more flexible than circuit switching in that it can emulate the latter (while the opposite is not possible), and vastly different bit rates can be multiplexed together. Also, the network's statistical multiplexing of variable rate sources may also yield a higher utilization of the channel capacity than what is obtainable with fixed capacity allocation. Most of these arguments have been brought forth and verified in a number of projects like MAGNET at Columbia University [1], [2], Prelude at CNET [3], [4], and PARIS at IBM [5]. Video coding for packet transmission has been considered by a number of authors [6]–[11]. Other issues of importance for packet video include error recovery [12], [7], [13], [9], [11], video resynchronization [14], [6], [11], and statistical analysis and modeling of the variable bit rate of coded video [15]–[19], [11]. The abstracts of the recent workshop on packet video give a good overview of new results in the field [20].

The amount of information in a video signal depends on the activity in the captured scene. When compressed, the resulting bit rate may be highly varying, which is referred to as variable bit rate coding. Most compression methods produce a variable output rate. Only when a fixed output rate is enforced, as with circuit-switched transmission, is the compression varied in order to keep the transmitted rate within the prescribed limit of the channel. Since this is done regardless of the information content in the signal, the quality of the received signal may vary with the compression. (Generally, the effect is reduced by appropriate buffering.) In contrast, statistical multiplexing in packet networks allows the transmission rate to vary in order to reflect the information contents of the signal. It is consequently expected that the receiver will perceive a constant video quality. Throughout the paper, we will assume that variable rate coding is being used.

In the case of variable-rate transmission of real-time information, we can no longer assume the separation of source, channel, and receiver as a valid paradigm. The transmitter of a compressed video signal will require various amounts of capacity over time and the packet network provides a channel whose capacity changes depending on the total load of the network. The receiver, in turn, has to function together with the channel where packets are lost or delayed. Hence, there are strong dependencies between the entities involved, which require consideration of the global system. A structured way of considering the entire system of packet video is to place the required functions in their proper context, a network architecture model, so that the interactions between functions can be determined [21], [2]. This model may also help to design a system which minimizes a function's dependency on the format of the video signal. The model is illustrated in Fig. 1. By using the Open Systems Interconnection model [22] of the International Standards Organization as the starting point for discussion, we are provided with an adequate terminology that is not contingent upon any specific network implementation. Note that the OSI model was not designed to support real-time transmission. The network architecture model will be considered mainly from a video processing point of view. The control flows will not be covered.

The paper is organized as follows. First some signal processing issues of packet video coding and transmission are analyzed in more detail. These are hierarchical source

Manuscript received April 15, 1988; revised December 16, 1988. This work was supported by the National Science Foundation under Grant CDR-84-21402. This paper was presented in part at the Second International Workshop on Packet Video, Torino, Italy, September 7–9, 1988.

G. Karlsson was with the Department of Electrical Engineering, Columbia University, New York, NY. He is now with IBM Research Division, Zürich Research Laboratory, CH-8803 Rüschlikon, Switzerland.

M. Vetterli is with the Department of Electrical Engineering, Columbia University, New York, NY 10027.

IEEE Log Number 8927586.

0733-8716/89/0600-0739$01.00 © 1989 IEEE

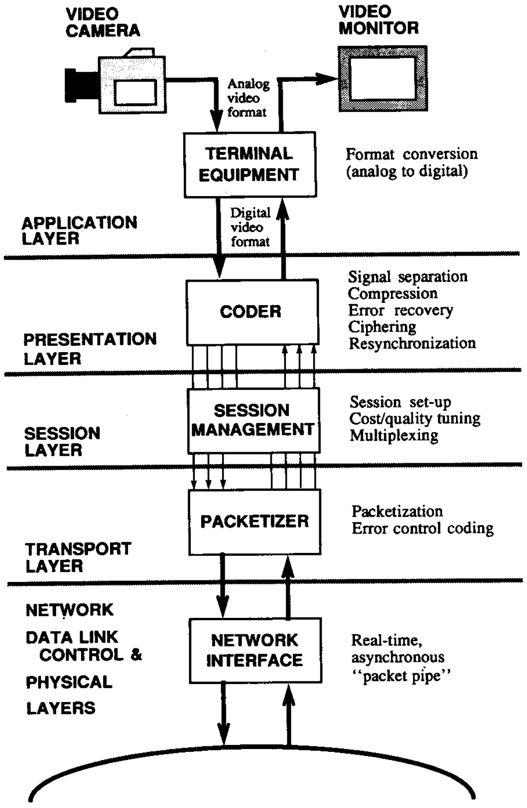


Fig. 1. The major functions of the higher layers of the packet video network architecture model. Note that the arrows indicate signal flows.

coding, error recovery, and the statistical behavior of variable-rate coded video signals. In Section III, video transmission considerations are covered pertaining to the lower layers of the model. Section IV presents the video-related functions fitted into the higher layers. Transmission delays and resynchronization of the video are discussed in Section V. Section VI presents simulation results of a packet video scheme based on subband coding with error recovery, indicating practical solutions to some of the issues raised throughout the paper.

II. Signal Processing Issues of Packet Video

A number of issues in packet video call for signal processing solutions. These issues include source coding and error recovery. This section describes them in more detail. (They are related to the network architecture model in Section IV.) Also in this section, the statistical behavior of a compressed video signal is discussed. The properties of the varying capacity requirement matter for the network design, while semantics and syntax of the video signal are irrelevant.

A. Hierarchical Source Coding

Hierarchical coding, known also as layered coding or embedded coding, is a technique first developed for packet voice transmission [23], wherein a signal is separated into subsignals of various importance to be coded and transmitted separately. This general technique, illustrated in Fig. 2, may be advantageously extended to video coding and transmission as well [7], [11]. In this way, the coding of a subsignal can be tailored to the information it carries and the subsignal can be transmitted with a priority reflecting its importance. Network congestion, where buffer overflow leads to packet discard, will affect mainly the subsignals of low importance. Thus, hierarchical coding offers a way of achieving error control by preventing loss of perceptually important information. Yet another reason is the possibility of cost/quality tuning for a session [7], [9]. This refers to the fact that high video quality can be traded for reduced transmission cost. A potential problem with hierarchical coding appears if some of the compressed subsignals yield low output rates; the packetization delay may become intolerable for fixed-length packets. The solution would be to multiplex low bit-rate subsignals, or, alternatively, to stuff them with dummy information to artificially raise the rate.

In our definition of functionality, the process of separating (and recombining) the input signal does not include the compression of the resulting subsignals. The signal separation ought to be such that the total amount of data bits in the subsignals equals the amount of bits in the input. Moreover, it is desirable to require the signal separation and recombination to be lossless. This way the information loss is limited to the compression and transmission, which partly can be controlled through compression level and transmission priorities. Given these constraints, there exist several feasible ways of separating a signal into subsignals. The better methods are those which also lead to improved overall compression. Alternatives include bit-plane separation [23], subband analysis/synthesis [24], and unitary transforms [25].

Bit-plane separation means that the most important information consists of the most significant bits of the image, and, progressively down through the bit layers, the least important subsignal is composed of the least significant bits. This separation has the peculiarity that the most important subsignal would yield the highest compression, while the least important one, which contains virtually no correlation, may only be compressed to a slight degree. This is the reverse of the other methods. We have found subband analysis to be an attractive way of decomposing a signal for coding and transmission [7]–[9] since the subbands have a natural hierarchy. The use of subband coding as hierarchical coding will be presented in Section VI. For unitary block transforms, common block sizes lead to 64 and 256 channels, respectively, a prohibitively large number for independent transmission which would thus require some multiplexing. Since a transform is applied on subblocks of the image, all transform coefficients with the same index could be gathered from the subblocks to form a subsignal (commonly known as Mandala reordering). Zonal encoding may thereby eliminate entire subsignals by not allocating any bits to them.

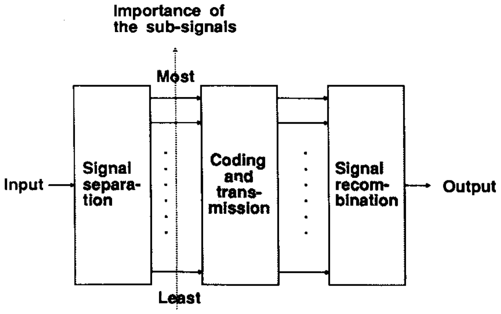


Fig. 2. A hierarchical coding system. The signal is separated into subsignals of various importance which may be coded and transmitted independently of one another. When received and decoded, the subsignals are recombined to form the output signal.

B. Error Recovery

Along with the advantages of packet transmission, such as services integration and variable bit rate transmission, inevitably comes the problem of lost packets. Packet loss is caused by bit errors in the packet's address field, leading the packet astray in the network, and network congestion, leading to discard of packets due to filled buffers. Statistical properties differ for the two causes: bit-error can realistically be modeled as an uncorrelated process which results from noise in the transmission channel [22]. Packet loss due to network congestion is not so straightforward; it depends on the transmission rate in relation to the total network capacity, as well as the resource allocation and the sizes of the network buffers. A thorough analysis of the topic in the case of packet voice can be found in [26].

There are mainly two approaches to error recovery: first, the use of error control codes, and second, error concealment by use of visual redundancy. The former offers perfect recovery from error until the number of errors exceeds the limit of the used code. In contrast, the latter method never gives perfect recovery but it can be engineered to be nearly imperceptible. Its advantage, however, is that it works, although with decreasing effectiveness, regardless of the number of errors experienced during transmission. It is worth noting that while complete and correct delivery are the foremost constraints on data transmission, video as well as voice signals can tolerate some information loss (which of course is exploited already at the compression stage).

1) Error Control Coding: If error control coding is applied along the bit stream of a signal, a lost packet means that a burst of hundreds of bits has to be corrected; a formidable task which would require codes of unreasonable lengths. Fig. 3 depicts a better solution. Here the signal is put into packets after which the error control coding is performed across the information field of the packets (i.e., bit interleaving) [12]. The error control codes used can be block codes or convolutional codes [27]. In both cases, the codes should preferably be *systematic* since it speeds up the decoding: if no error is present, the information packets remain unaltered.

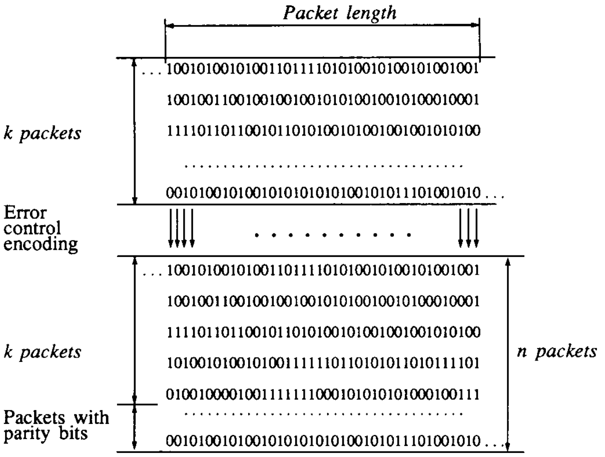


Fig. 3. Error correction encoding should be applied perpendicular to the segmented bit stream to achieve bit interleaving. The figure illustrates the case when a systematic block code is used.

For an (n, k) block code, the number of packets that need to be stored is equal to the number of information bits of the code, k. Since all k packets must be created before the encoding can take place, the coding introduces a periodically varying delay. This can be avoided when the code is systematic by using copies of the information-bearing packets in the calculation of the parity symbols. Note that the same delays are incurred at the decoder, where they cannot be avoided. The minimum distance that the code should provide is governed by the network's probability of packet loss and the correlation of such loss. In most packet-switched networks, there is no absolute bound on the packet loss, whereby the number of lost packets can exceeded what the code can correct. Consequently, some other correction, such as the one presented next, needs to be invoked to avoid breakdown of the video session.

2) Error Concealment by Use of Visual Redundancy: Error concealment by use of visual redundancy takes place after the source decoding but before the signal recombination of the hierarchical source coding system (see Fig. 2). The general requirements for this method are that the error propagation is limited and that the locations of the lost values are known, i.e., they can be treated as erasures [7], [9]. The way these two properties can be obtained will depend on the source coding method used for the subsignals. Generally, recursive coding methods (e.g., DPCM) are less suitable for this type of error recovery than FIR-based methods, such as transform and subband coding. This is due to the strong linkage created between samples in the compressed signal. Section IV-B presents how the erasure property may be obtained.

An erased area of an image may be approximately concealed by spatial and temporal interpolation, or statistical image reconstruction methods. However, the latter are usually too computationally complex to be performed at video rate. Whether information in the surrounding areas, and previous and following video frames should be used in the interpolation depends on the signal separation method used in the hierarchical coding. Regardless of the success of the concealment, in a perceptual sense, the limited error propagation guarantees that the session never needs to be terminated for any amount of lost packets. An encouraging performance of error concealment by use of visual redundancy is indicated by simulation results in Section VI-B.

C. Statistical Behavior of Video Signals

The previously discussed issue shows how stochastic properties of the network, i.e., packet loss and variable delays, influence the development of the signal processing functions. Similarly, knowledge of the statistical behavior of variable rate coded video will be necessary for proper network design. This relationship is illustrated in Fig. 4, which also reinforces the argument about dependencies between source, channel, and receiver, as mentioned in the Introduction. However, variable rate coded video signals are problematic to describe since they are highly varying, with a bursty behavior. The bursts correspond to activity in the captured scene and they may therefore last several seconds. Consequently, the varying rate cannot be sufficiently smoothed out through buffering since it would introduce unacceptable delays. However, when the rates of all video sessions are summed in the channel, the total rate exhibits less burstiness as a result of the statistical multiplexing that takes place (assuming independent sources) [15]-[19], [11]. As the number of sources increases, the ratio of the standard deviation to the mean of the summed rate goes towards zero. Hence, statistical multiplexing may yield a higher channel utilization than what is possible in circuit switching.

Voice signals have been successfully modeled by a two-state (voice/silence) Markov chain. For video there is no direct analog such as motion/no motion. However, there has been a model derived from the voice model, in which the total output rate from N video sources is taken as the aggregate rate from several ($>> N$) independent on-off sources [18]. Thus, the rate variations are modeled as discrete steps, where each step corresponds to the output from an on-off source, and where only one such source may change state at a time (i.e., a birth-death process). While the model does not capture features of the video rate of a single source, such as its burstiness, the model may serve well for statistically multiplexed sources.

III. The Lower Layers of the Network Architecture Model

The previous section described some signal processing issues of packet video. Their corresponding functions will be part of the higher layers of the network model. First, however, the lower network layers need to be briefly discussed. These layers comprise the physical link layer, the data link control layer, and the network layer (see Fig. 1). They form a network node, as opposed to the higher layers which typically reside in equipment on the customer's premises. If the division of functionalities between the layers is kept as suggested in this paper, then all of the lower layers may be designed without consideration of the signal type. The only requirement is that the layers have to support real-time services; it is irrelevant whether it is video or voice (except in need for capacity). Generally, this requires the lower layers to be as simple as possible without any processing of the information; any sophisticated signal handling is better performed at the higher layers where the signal's format and importance are known. Thus, the network may be described as a real-time, asynchronous "packet pipe," where packets inserted at one end come out orderly at the other end, although some will contain errors and others might have been lost.

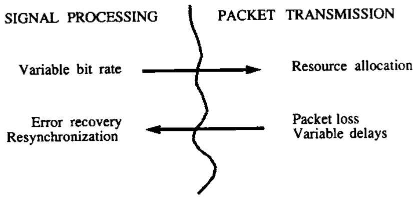


Fig. 4. The interactions between signal processing and transmission. The behavior of variable rate coded video will affect decisions on resource allocation. That, in turn, will influence the network's behavior in terms of packet loss and delay variations, which has to be remedied at the receiver by error recovery and resynchronization.

The requirements for the physical link are adequate capacity and low bit-error rate. It is, however, difficult to quantify these parameters—in general, they are determined by the physical limits of the state-of-the-art technology. To give a sense of the needs involved in a general network carrying video, consider transmitting 30 channels with broadcast video quality at 45 Mbits/s. This would require a link with a capacity in excess of 1.35 Gbits/s. An average of no more than one bit error per video frame for each of these sessions (i.e., 900 errors per second) would correspond to a bit-error probability of less than 7×10^{-7} for the 1.35 Gbit/s link.

Most of the tasks commonly associated with data link control for data transmission are incompatible with real-time services. For example, automatic repeat request is unsuitable as an error handling technique for real-time transmission. Even though the propagation and processing delays may be short, retransmission will introduce delay variations which are to be avoided. In short, there will foreseeably be no processing of data-link frames that contain real-time information. Therefore, data-link control is reduced to deal only with link-management issues.

The network layer provides end-to-end communication and shields the transport layer above from any physical aspects of the transfer medium [22]. The functions associated with the network layer during a data transfer phase include: routing (switching), congestion control, and packet duplication for broad- and multicast sessions. The necessity to keep end-to-end delays and packet delay jitter under control requires video transmission to be conducted over a fixed route, a virtual circuit, which also guarantees that packets are delivered in order. The logical channel number should be protected in order to avoid packet loss, or worse, intrusion (delivery of unwanted packets) due to bit error in the address field. Since the delivery of video packets cannot be warranted by retransmission, the network layer should, through the use of congestion control, maximize the probability of successful and timely delivery. In order to perform such a control in a sensible manner, the layer should provide transmission priorities. Priorities may not be necessary if capacity reservations are available. Lowered transmission quality due to congestion can thereby be avoided by allocating generous amounts of capacity to important signals.

In summary, the network layers should act as a real-time "packet pipe" where order is maintained. However, some of the delivered packets may contain bit errors while others have been lost. To minimize the loss of important data, the network layer ought to provide priority classes. Finally, for multiparty sessions, addressing has to allow multiple destinations and the network nodes must provide packet duplication.

IV. The Higher Layers of the Network Architecture Model

The higher layers are the transport, session, presentation, and application layers (see Fig. 1). In these layers, we place functions for which specific video issues come into consideration. As seen in the previous section, the lower layers are not video specific, but rather they provide general real-time service which could serve voice as well as video. Each of these layers would shield the layer above from some of the physical structure of the link, so that at the network layer the notion of a virtual network is created whose physical implementation is irrelevant. There is a duality between this and the way we would like to see the video-specific, higher layers: an upper layer has to shield its service provider from some of the peculiarities of the format of the video signal. The higher layers will therefore be presented top-down to illustrate how the format dependence is incrementally reduced.

While the lower layers are resident in the network nodes, the upper four are at the customer's premises. In that sense, a packet video coder is seen as being the set of functions associated with the upper four layers. With a network that provides real-time transmission, the user's choice of video format, compression method, and encryption can be made independently of the network. Hence, the network does not restrict the introduction of new signal formats or more advanced compression methods.

A. *The Application Layer*

The application layer forms the boundary between the user and the network. For the signal, it provides analog inputs and outputs which adhere to the standard of the user's choice and the analog-to-digital and digital-to-analog conversions. This layer is thus dependent on both the analog and digital video formats. To possibly obtain compatibility between different analog video formats, and to limit the variety of digital formats [see Fig. 5(a)], a highly reduced set of digital video formats could be used, as shown in Fig. 5(b). The allowed set of formats has to be such that any possible analog video format can be obtained from its members.

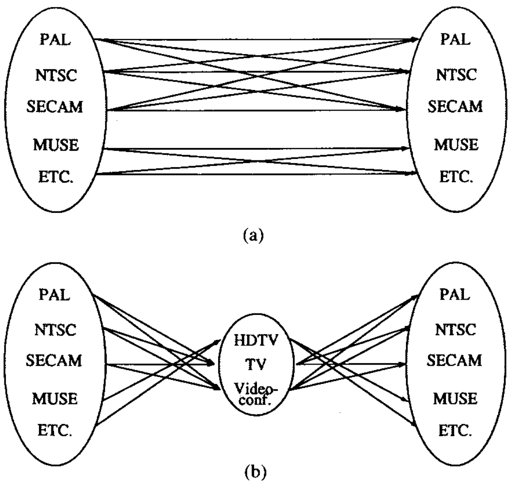


Fig. 5. Cross-compatibility between video formats: (a) direct format to format conversion, and (b) the use of a limited set of intermediate formats.

B. *The Presentation Layer*

The presentation layer holds functions which perform some type of signal conversion. Both the signal separation and the compression of hierarchical source coding would be performed within the context of the presentation layer. The other functions include ciphering, error concealment by use of visual redundancy, and video resynchronization. Resynchronization will be covered separately in Section V. Owing to the reduced number of digital video formats stemming from the application layer, presentation layer implementations can be restricted to one class (or standard) for each format.

Error concealment performed by utilizing visual redundancy, as explained in Section II-B2, relies entirely on the assumption that error propagation is limited and that the locations of lost data are known. However, one cannot expect lost data to be aligned to particular points in a video frame, such as the beginning of a scan line; nor can the number of erased values be deduced. The information about the ideal splitting points of the bit stream is local to the presentation layer where the video format is known, while the packet length is known only at the transport layer. Consequently, limited error propagation has to be achieved locally at the presentation layer by adding control information to the bit stream. A feasible way of doing this is to insert synchronization flags after every *i*th sample in a subsignal [13]. The distance between flags should be chosen so that the values may be concealed reasonably well if lost, while adding little overhead. The flags have

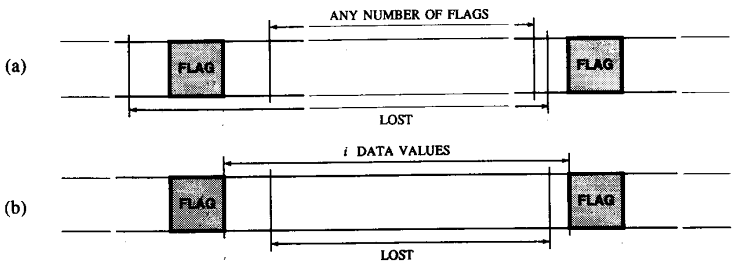


Fig. 6. Two cases of lost data: (a) one or more flags are missing, and (b) no flags are missing but intermediate data values have nevertheless been lost.

to consist of a unique identifier followed by a sequence number to indicate location and, in case of error, the number of erased segments. The flag's uniqueness has to be guaranteed by bit stuffing or reserved codewords. The latter appears advantageous if variable length codewords are being used: the flags are inserted before the variable length coding and they are transformed, as any other value, into their reserved codeword. One lost flag would imply that two data segments have been erased at least partly, as in Fig. 6(a), while the case shown in Fig. 6(b) would be detected since the segment would not yield *i* samples. All correctly received segments of a video frame would be decoded before the concealment is made, which in turn is done before the signal recombination.

C. *The Session Layer*

The session layer is mainly responsible for session setup and tear-down. Additionally, in our model, the session layer implements the functions which would be invoked only a limited number of times over an entire session. These functions are in contrast to continuously invoked functions, such as the compression. The session layer should provide not only different types of sessions, but also flexibility in the quality of the sessions. The functions of the session layer are completely freed from the format of the video signal. The layer receives a set of subsignals belonging to a single session and, owing to the format independence, some of these may carry sound information. Hence, it is here, at the session layer, that a complete session of integrated real-time services is created.

1) Session Types: One can foresee several types of video sessions, which can be end-to-end negotiated or locally negotiated. The former can be the common point-to-point session, but also multicast and multidrop sessions. We define a multicast session as a one-to-many communication where transmission to each receiver has been negotiated, and all destinations are explicitly addressed during the data transfer phase. Multidrop is analogous, except the communication is from many to one. A conference session would typically consist of multicast outgoing transmission and multidrop return. Both these sessions could be setup and torn-down together and all session management could apply to them equally. Only one type of locally negotiated session is conceived, namely broadcast. It is considered to be based on permanent virtual circuits. The session setup is thus negotiated only with the local network node. To receive a broadcast channel, the associated virtual circuit is tapped at the node. (For more details see [28].) Multidrop is similar to broadcast in that the receiver chooses, at any point in time, which virtual circuit to tap by switching between the incoming circuits at the local node.

2) Session Quality: The session quality depends on two parts: source coding and transmission. In conjunction with hierarchical coding, a desirable quality and output bit rate may be achieved through transmission of an appropriate set of subsignals. The greater the number of subsignals, the higher the quality and the output rate. If the coding method can provide more than one compression mode, a greater flexibility in the cost/quality tradeoff may be obtained by changing compression of the subsignals. For transmission, at issue is the degree to which the session may be affected by network congestion. This is contingent upon the resource sharing policy used by the network management. A prodigal resource allocation may be likened to a circuit-switched system while a parsimonious one will yield a higher capacity utilization, but also more delay and packet loss. For a given resource allocation policy a more consistent quality is obtained if a higher number of the transmitted subsignals is given high transmission priority, rather than only the most important one. The transmission quality may be raised further by requesting the transport layer to perform error control encoding on a desired set of subsignals. So, the lossless session quality is set through the number of transmitted subsignals, and the allowed deterioration under network congestion is decided through the use of the transmission priorities and error control coding.

It is our opinion that packet video, compressed with hierarchical coding, would be advantageously transmitted as a hybrid of synchronous and asynchronous time division [28]. The hierarchy's most important subsignal would be transmitted with a fixed reservation of network resources, while the less significant subsignals would have to vie for their share of capacity. This hybrid of allocation policies reduces the stress on the error recovery mechanisms while it still yields higher channel utilization than does pure synchronous time division. The feasibility of such a scheme is discussed in conjunction with the simulation results given in Section VI-B1.

D. The Transport Layer

Functions associated with the transport layer are the segmentation of a data stream into packets, the reassembly of the stream at the receiver, and error control coding. Since the network service is restricted to virtual circuits, all packets are guaranteed an orderly delivery and reordering is therefore not of concern. The transport layer serves all subsignals emanating from the hierarchical coding at the presentation layer and other associated signals, such as sound, which have been added to the session at the session layer. Each such signal would be independently segmented, but the packetized signals would be multiplexed onto the same route at the network layer. Sound and video information will thereby follow a single path so that the delay difference between the two is minimized. Note that the segmentation process does not have information about the video format (such as beginning of frame or scan line); the bit stream may therefore be cut at any point.

For error recovery relying on error correcting codes, the amount of data received has to equal the amount transmitted, or it would place the error correcting encoder and decoder out of phase. Consequently, the transport layer at the destination node has to detect lost packets and replace them with dummy packets so that synchronization can be maintained. Since orderly delivery is guaranteed, end-to-end sequence numbers may suffice for this purpose; a detected gap in the sequence indicates the loss of one or more packets. The necessary range of the sequence numbers has to be determined in relation to the channel code used, so that all gap lengths can be detected which can be corrected by the code. Note that packet intrusion would cause a similar problem by increasing the number of packets received. However, we require the network layer to protect the address field of the packet so that intrusion cannot occur (see Section III). Delays introduced by the channel coding depend on the code length n and the packet size: long packets and codewords give longer delays but less overhead. However, this tradeoff can be resolved without affecting the functions in the layers above, which are independent of packet format. Consequently, the choice between the use of variable or fixed length packets can be made locally as well. While fixed length packets simplify segmentation and packet handling along the transmission path, variable lengths of the packets could be used to keep the packetization delay constant.

Note that end-to-end retransmission is not a possible error recovery method. First, multicast and broadcast sessions would not be feasible if a retransmission scheme was in effect. Consider that different packets may be lost on the various paths of transmission which can result in unreasonable requests for retransmission, proportional to the number of recipients. Second, there is a risk of positive feedback: assume a congested network where all video providers experience packet loss. The strict delay constraint would bar random waiting times before retransmission. So if all providers rely on retransmission for error recovery, the congestion could be aggravated and might lead to more severe performance degradation [7], [9].

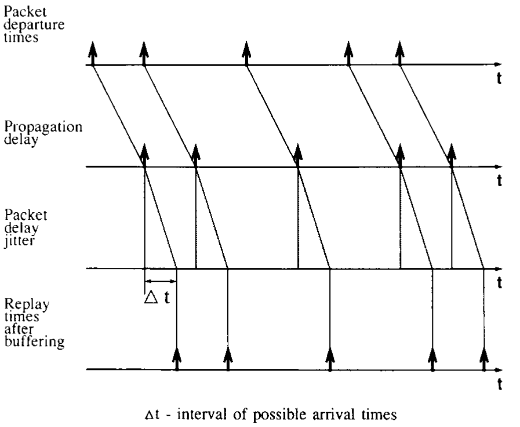

Packet departure times
t
Propagation delay
t
Packet delay jitter
Δ t
t
Replay times after buffering
t
Δt - interval of possible arrival times

Fig. 7. The delays in a packet-switched network are composed of a fixed part due to the propagation delay, and a variable (jitter) part which is due to waiting time in buffers.

V. Resynchronization of Video

The timing problems of packet video are twofold. First, there are variations in transmission delays, referred to as packet delay jitter. Analogous to packet loss, these variations are an inherent problem with packet transmission. Packet delay jitter can be removed by buffering the packets at the receiver, whereby the delay becomes fixed and equal to the maximal value. This is illustrated in Fig. 7. Transmission delays are irrelevant for one-way sessions, such as broadcast TV. For video conversations (video and voice), in contrast, end-to-end delays are of foremost importance, since long delays impede information exchange. Increased packet loss is therefore accepted in order to achieve reduced delays. Video, as an information carrier, is subordinate to sound, and the video delay has to be bound only to provide lip-synchronization. Lip-synchronization error appears to be unobjectionable for video-to-voice lags in the range of -90 to $+120$ ms [29].

Second, the absence of a common time reference for the encoder and the decoder adds further complications to the reconstruction of a synchronous video signal. According to its clock, the decoder might thus expect data at a higher or lower rate than is being transmitted. Fig. 8 shows the various cases: (a) the clock frequencies are equal, (b) the receiver clock is fast compared to the transmitter clock, and (c) the receiver clock is slow relative to the one at the transmitter. Commonly, the transmission clock frequency is deduced from the arriving packets [14],

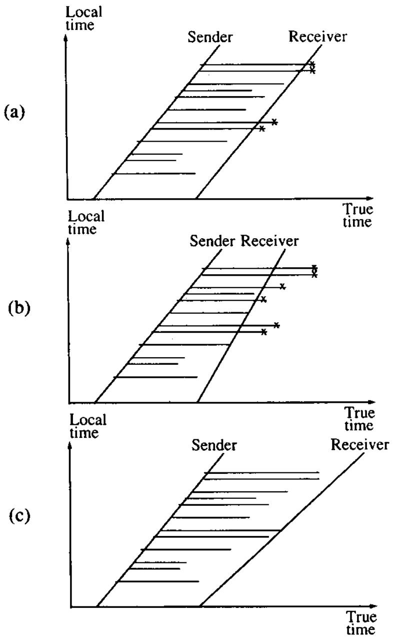


Fig. 8. In (a) the clocks have equal frequencies, and only the packets which are unduly delayed by the network are considered lost (marked by x). In (b) the receiver clock is fast compared to the sender, thus more packets are considered lost. If the receiver clock is slow, as in (c), too much time is allowed for arriving packets.

[6], [11]. This can be done by monitoring the level of the input buffer [14], or by synchronizing the receiver clock to time information in the packets (by use of a phase-locked loop) [6], [11]. Both methods are, however, complicated by packet delay jitter and, if used, variable rate coding. There are also cases when the receiver clock cannot adapt to the received data rate. For example, when video is received from more than one sender (as with multidrop in Section IV-C1), only one of the signals can be used for clock adjustment. So, the frame rates of the other received video signals have to be altered to match the playout rate. This could be performed by occasionally repeating or skipping frames, possible in combination with motion-compensated temporal filtering. (For more details see [28].)

VI. Results from a Packet Video Simulation

In this section, results are presented pertaining to hierarchical source coding with error recovery, and the statistical behavior of its output rate. The simulation involved a subband coding scheme and error concealment by use of visual redundancy. The latter was simulated by random discard of packets; no actual network was used in the simulation. The full details of the coding scheme and associated results can be found in [7]-[9], [28]. The coding scheme is designed to provide quality sufficient for video conferences. The investigation was aimed at robust transmission and low complexity, rather than maximal compression.

A. Subband Coding of Video—The Implementation

We have investigated subband coding as hierarchical coding for packet networks. In this method, the signal separation of Fig. 2 is referred to as subband *analysis* and the recombination as *synthesis*. The analysis of a video signal is, in our scheme, achieved by splitting the frequency spectrum in all three dimensions (i.e., temporally, vertically, and horizontally) to obtain a total of 11 three-dimensional frequency regions. This is illustrated in Fig. 9. Following the filtering, the subbands are obtained by subsampling the signals in each dimension to the new Nyquist frequencies. As pointed out in Section II-A, signal separation and recombination should ideally be lossless, and should not increase the amount of data that need to be coded. The filters used enable a perfect reconstruction of the analyzed signal in the absence of coding and transmission loss [30], and the sum of data in the subbands equals that of the input. Also, the information in each subband is amenable to well-tailored quantization schemes which may be adjusted according to perceptual criteria. Thereby compression can be highest in subbands where distortion is least visible.

Subband 1, resulting from low-pass filterings in all three dimensions, retains a high variance of its intensity distribution, which is similar to that of the input. All of the other bands have greatly reduced variance and they can be sufficiently encoded by PCM. Subband 1, in contrast, needs more powerful encoding, which must be weighed against a possible reduction in robustness. As a compromise, the band is encoded with first-order one-dimensional DPCM. The prediction error and the PCM values are run-length encoded. The coding scheme has a low complexity, and can actually be implemented without multiplications [8]. It is worth noting that subband analysis may also be used for size reduction if up-sampling and synthesis filtering is bypassed [31].

B. Simulation Results

The subband coding scheme was simulated using a monochrome sequence consisting of 100 512 × 480 images, with 8 bits per pixel, which corresponds to a bit rate of 60 Mbits/s. According to our results, the argument that variable rate may give a constant quality appears to hold. The average rate of the total output is 2.7 Mbits/s with a standard deviation of 0.62 Mbits/s. In contrast, the mean signal-to-noise ratio (SNR) is 30.4 dB, with a standard deviation of only 0.3 dB!

The subband coding yields a separation of the image information, which results in vastly different behavior of the output rates. In Fig. 10, the mean rate and standard deviation have been plotted for all subbands. As shown, the temporal low-pass bands are generally less variant than the temporal high-pass bands, with the exception of band 5. This band is vertically high-pass filtered, which yields a large variance due to the interlaced format of the video. Fig. 11 shows the temporal behavior (i.e., burstiness and constancy) of four subbands. Owing to the temporal sub-

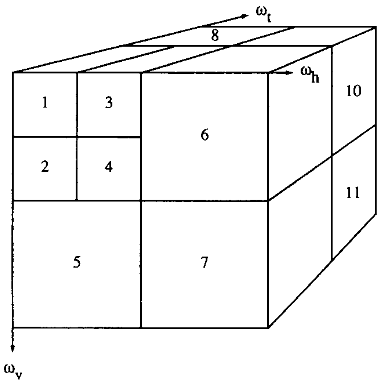


Fig. 9. The 11 frequency regions of the subband analysis. The total region is initially split along the midpoints of each frequency axis. The one which contains both low temporal and low spatial frequencies is split into four spatial-frequency regions.

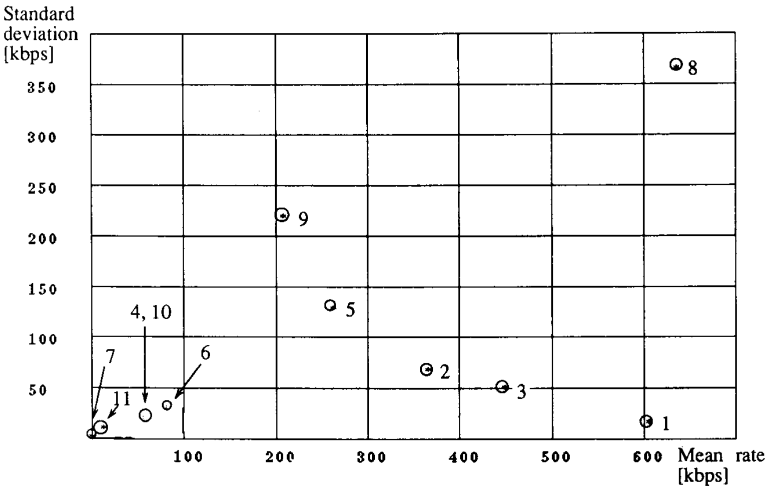


Fig. 10. Mean rate and standard deviation of all subbands (for the band's indexes see Fig. 9).

sampling, the frame rate is only 15 frames per second. Note that the bursts last for several frames. Hence, only limited smoothing can be obtained by buffering, if the end-to-end delay should meet the real-time constraint (Section V-A). In addition, the figure hints that rates from the subbands are correlated. The standard deviation of the summed rate is 0.62 Mbits/s. If the bands were independent, the sum-of-variances would give 0.45 Mbits/s as standard deviation.

1) Session Quality: In Fig. 12, the output rate is plotted against the SNR for selected subsets of all subbands. This can serve as an indication of how degradation can vary with available transmission capacity. The pictures in Fig. 13 represent the quality associated with the four cases in Fig. 12. Cost/quality tuning (see Section IV-B2) may be achieved by omitting less important subbands in order to yield a desired output rate. The possible range of quality and output rates corresponds to those of Fig. 12. Note that packet loss is a temporary quality decrease along this curve, while cost/quality is a permanent one for the session.

The DPCM encoded subband 1 will require the highest possible transmission priority and, if possible, capacity reservation according to its maximum rate. Since its output rate is nearly constant, such a policy would not waste resources. In fact, its maximum rate in the simulation is 619 kbits/s as compared to a mean of 604 kbits/s. The other subbands can be transmitted with lower priority since they are not necessary for the continuity of the session; only its quality. A resource allocation could, in the case, guarantee capacity according to the mean rates of the bands. Thus, they will have to compete for unassigned portions of the network capacity when need be. An even lower allocation class would correspond to subbands which have no guaranteed capacity and therefore have to vie for their entire bandwidth. Priority assignment and ca-

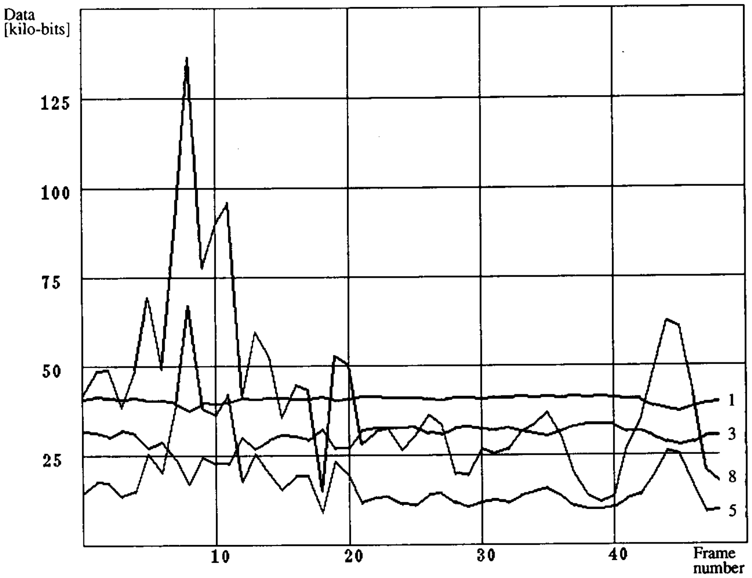


Fig. 11. The amount of data (in kilobits) per frame for four subbands (1, 3, 5, and 8). Note that the sequence length is halved due to the temporal subsampling.

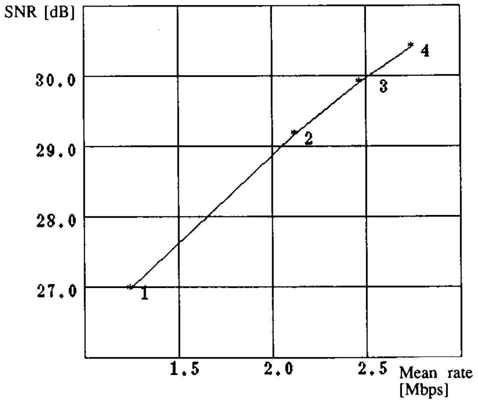


Fig. 12. The SNR as a function of bit rate. Case 1 consists of subbands 1 and 8; case 2: 1-4, and 8; case 3: 1-6, and 8; and case 4 includes all subbands.

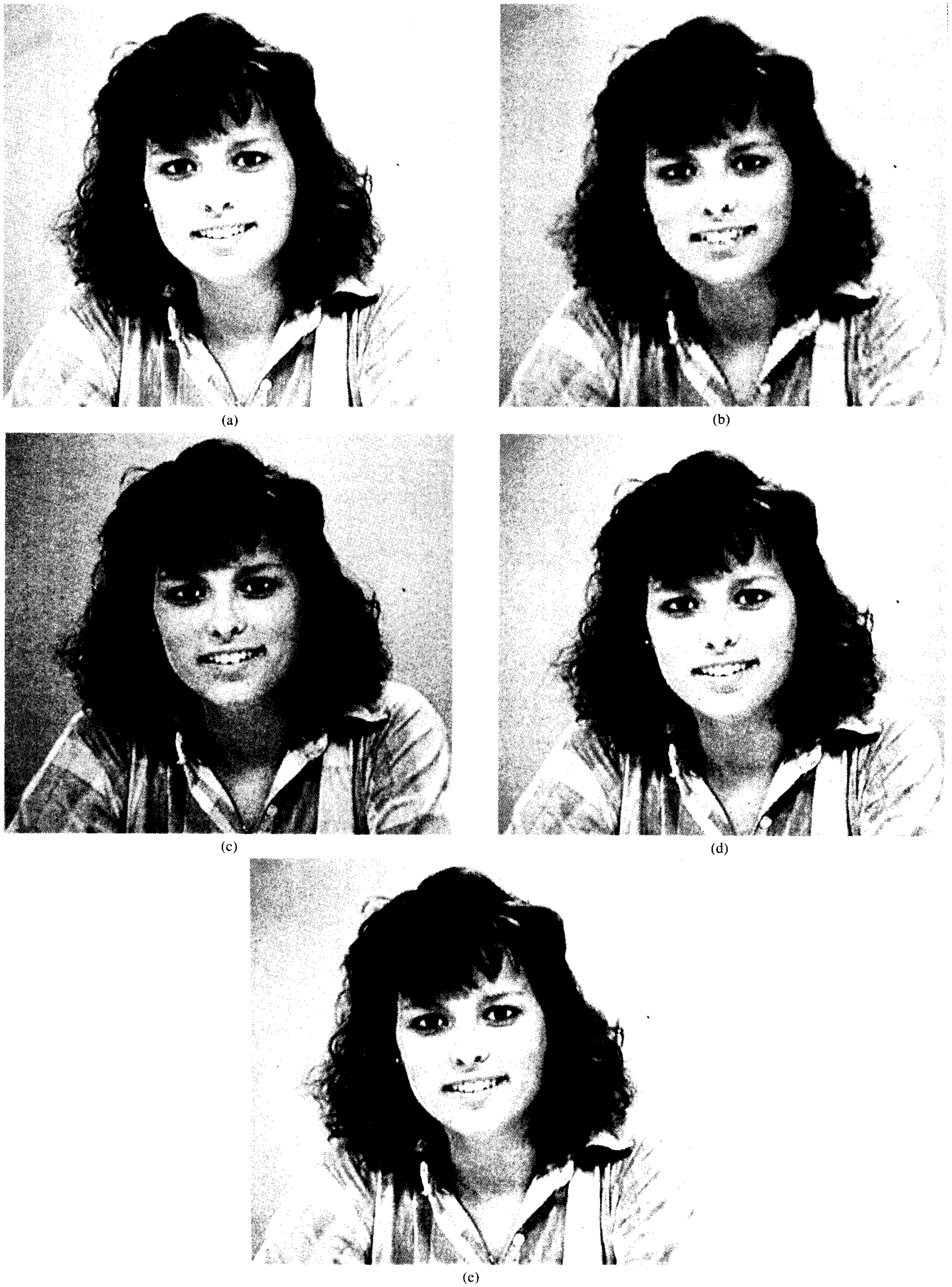
pacity allocation for a subband could be made according to mean, variance, and visual importance, as given by Figs. 10 and 12.

2) Packetization Delays: The packetization delays, corresponding to the subbands' output rates, have been tabulated in Table I for 1024-bit packets (as used in [1] and [2]). The maximum delay is not an exact figure since it is not calculated by integration of the variable bit rate, but it is calculated from a constant rate, equal to the minimum of the variable rate. The table illustrates a potential problem with hierarchical coding: excessive packetization delay. When calculating the permissible end-to-end delay, the maximum packetization delay has to be used. Bands such as 7 and 11 may be either multiplexed, or omitted permanently. However, bursty subbands which at any point in time may yield a low bit rate are potential problems, such as subbands 4, 6, 9, and 10 in the table. While multiplexing can eliminate the maximum delay, the added rates of these bursty bands may require unacceptable processing needs at rate peaks. One solution would be to avoid transmission of a packet that required more than the allowed time for its packetization.

3) Modeling of the Variable Output Rate: The output of each subband may be more easily modeled than the total rate (see Figs. 10 and 11). For example, subbands 1, 2, and 3 could be modeled as constant rate sources. The subbands 5, 8, and 9 could be modeled as simple, but cross-correlated, on–off sources, with the average time in the on-state equal to the average duration of the bursts. The remaining subbands could be excluded from the modeling because of their minor contribution to the total bit rate. An investigation of the statistical properties of the output rates of a subband coding scheme is reported in [19], where the simulation results include ten video sequences, each of 2 min duration.

C. Error Recovery

Our study covers error concealment by use of visual redundancy, as discussed in Section II-B2; it does not include the use of error correcting codes. We have found that subband coding offers a good framework for performing error recovery [7], [9]. First, synchronization flags are added according to the method of Section IV-B so that the location and number of lost packets are known. The encoding is memoryless, except for subband 1, which will need only one PCM value per synchronization flag in order to limit the error propagation (hence, the simple encoding). It is therefore possible to obtain restricted error propagation with limited overhead. Second, only subband 1 requires some form of computed concealment of the era-



(a)

(b)

(c)

(d)

(e)

Fig. 13. Pictures representing the cases in Fig. 12. (a) The original. (b)–(e) show cases 1–4.

TABLE I
PACKETIZATION DELAYS FOR THE SUBBANDS. THE PACKET SIZE WAS 1024 BITS. (FOR BAND 7, THE MINIMUM BIT RATE IS 0, AS CALCULATED FROM THE MINIMUM NUMBER OF BITS PER FRAME. HENCE, THE CORRESPONDING MAXIMUM PACKETIZATION DELAY IS NOT VALID).

Sub-band	Mean [ms]	Max [ms]
1	1.7	1.8
2	2.8	4.0
3	2.3	4.0
4	16.1	93.9
5	3.9	7.5
6	12.3	280.1
7	4053.8	----
8	1.6	5.7
9	4.9	59.5
10	17.6	38.6
11	79.4	397.3

sures. The other subbands recover to near invisibility of the errors by setting the erasures equal to zero (the mean). For subband 1, the erasures were patched over by using the corresponding area from the previous frame. The performance of this error recovery method was simulated by randomly erasing 1024-bit packets. For lack of a better model, the simulated loss was taken to be uncorrelated over time and between subbands. Representative images are shown in Figs. 14 and 15; in Fig. 14, five randomly chosen bands, not including subband 1, have suffered loss of one packet each, and Fig. 15 shows a case when subband 1 has been affected by a lost packet. Apparently, the method works well for areas without motion, but subband 1 leaves visible error in areas with motion. This inconsistency can be eliminated by using motion estimation to find the appropriate area to be used as concealment. Spatial filtering can then further reduce the visibility of error by smoothing along the border of the replaced area. This was demonstrated in a related study [28].

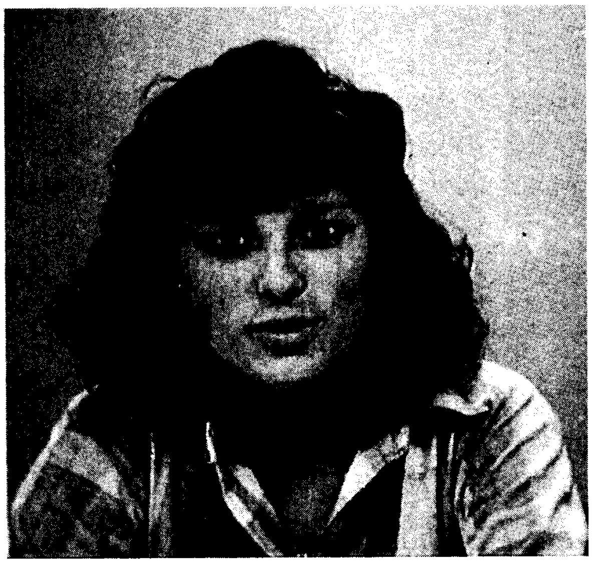


Fig. 14. The effect of packet loss. Five packets, of 1024 bits each, were lost, which corresponds to 2.9 percent of all the packets in the frame. The affected subbands (2, 5, 9, 10, and 11) were chosen at random, as were the particular packets.

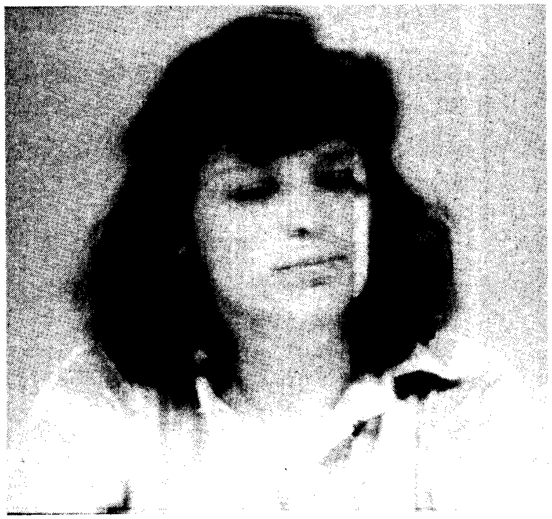


Fig. 15. Packet loss in subband 1 cannot be satisfactorily substituted from the corresponding area in the previous frame when there are high amounts of motion.

VII. CONCLUSIONS

Packet video has been studied from a system point of view and the functions necessary for video communications have been addressed. On signal processing issues, particular consideration was given to coding, error recovery, and the statistical properties of the transmission rate, for which simulation results were also presented. In terms of the network model, the lower layers would provide realtime transmission without regard to the signal's format. It is believed that network sessions should be conducted over virtual circuits, and that the network should provide for priorities if congestion control is exercised. The functionality should also include provisions for multiple destinations. Video-specific functions were discussed in the context of the higher layers. The functions could be arranged so that the dependence on the video signal's format is stepwisely reduced. The issues included format conversion, hierarchical coding, provisions to limit error propagation, session types, session quality and cost, and packetization. The timing problems inherent in an asynchronous network were also covered. Network delays are irrelevant for one-way sessions, but critical for two-way sessions, where the video delay has to be bound to give lip-synchronization. In addition to the network delays, there might be a disparity between the sender and the receiver clocks, which might be remedied by repeating or skipping video frames. Simulation results were presented for a hierarchical packet video coder based on the technique of subband coding. It appears that subband coding is a promising way of doing hierarchical source coding. The complexity is low and, in terms of quality and compression, the scheme promises a performance similar to transform coding. The simulation also verified various conjectures, like variable rate constant quality coding, cost/quality tradeoffs, and recovery from packet loss.

Acknowledgment

The authors thank Bell Communications Research for providing the image sequence; J-Y Cochennec of CNET, A. Lazar and S-Q Li of Columbia, J. Ma of IBM, and M. Garrett of Bellcore for helpful discussions; and the reviewers for their constructive comments.

References

[1] A. A. Lazar *et al.*, "MAGNET: Columbia's integrated network testbed," *IEEE J. Select. Areas Commun.*, vol. SAC-3, pp. 859-871, Nov. 1985.

[2] A. A. Lazar and J. S. White, "Packetized video on MAGNET," *Opt. Eng.*, vol. 26, pp. 596-602, July 1987.

[3] P. Gonet, "Fast packet approach to integrated broadband networks," *Networks*, vol. 9, pp. 292-298, Dec. 1986.

[4] M. Devault *et al.*, "The prelude ATD experiment: Assessments and future prospects," *IEEE J. Select. Areas Commun.*, vol. 6, pp. 1528-1537, Dec. 1988.

[5] I. Cidron and I. S. Gopal, "PARIS: An approach to integrated high-speed private networks," *Int. J. Digital Analog Cabled Syst.*, vol. 1, pp. 77-85, 1988.

[6] W. Verbiest and M. Duponcheel, "Video coding in an ATD environment," in *Proc. Third Int. Conf. New Syst. Services Telecommun.*, Liège, Belgium, Nov. 1986, pp. 249-253.

[7] G. Karlsson and M. Vetterli, "Subband coding of video signals for packet-switched networks," in *Proc. SPIE Conf. Visual Commun. Image Processing II*, vol. 845, Cambridge, MA, Oct. 1987, pp. 446-456.

[8] ——, "Three dimensional sub-band coding of video," in *Proc. ICASSP '88*, New York, Apr. 1988, pp. 1100-1103.

[9] ——, "Sub-band coding of video for packet networks," *Opt. Eng.*, vol. 27, pp. 574-586, July 1988.

[10] N. Ohta *et al.*, "Variable rate video coding using motion compensated DCT for asynchronous transfer mode network," in *Proc. IEEE 1988 Int. Commun. Conf.*, Philadelphia, PA, June 1988, pp. 1257-1261.

[11] W. Verbiest *et al.*, "The impact of the ATD concept on video coding," *IEEE J. Select. Areas Commun.*, vol. 6, pp. 1623-1632, Dec. 1988.

[12] P. J. Lee "Forward error correction coding for packet loss protection," presented at First Int. Packet Video Workshop, Columbia Univ., New York, May 1987.

[13] G. Aartsen *et al.*, "Error resilience of a video codec for low bit rates," in *Proc. ICASSP '88*, New York, Apr. 1988, pp. 1312-1315.

[14] J-Y Cochennec *et al.*, "Asynchronous time-division networks: Terminal synchronization for video and sound signals," in *Proc. GLOBECOM '85*, New Orleans, LA, Dec. 1985, pp. 791-794.

[15] B. G. Haskell, "Buffer and channel sharing by several interframe picturephone coders," *Bell Syst. Tech. J.*, vol. 51, pp. 261-289, Jan. 1972.

[16] T. Koga *et al.*, "Statistical performance analysis of an interframe encoder for broadcast television signals," *IEEE Trans. Commun.*, vol. COM-29, pp. 1888-1875, Dec. 1981.

[17] S-S Huang, "Source modelling for packet video," in *Proc. IEEE 1988 Int. Commun. Conf.*, Philadelphia, PA, June 1988, pp. 1262-1267.

[18] B. Maglaris *et al.*, "Performance models of statistical multiplexing in packet video communications," *IEEE Trans. Commun.*, vol. COM-36, pp. 834-844, July 1988.

[19] P. Douglas *et al.*, "Statistical analysis of the output rate of a subband video coder," in *Proc. SPIE Conf. Visual Commun. Image Processing III*, Cambridge, MA, Nov. 1988, pp. 1011-1025.

[20] Abstracts from the Second International Workshop on Packet Video, Torino, Italy, Sept. 7-9, 1988.

[21] L. Chiariglione and L. Corgnier, "System considerations for picture communication," in *Proc. IEEE 1984 Int. Commun. Conf.*, Amsterdam, May 1984, pp. 245-249.

[22] M. Schwartz, *Telecommunication Networks*, Reading, MA: Addison-Wesley, 1987.

[23] T. Bailly *et al.*, "A technique for adaptive voice flow control in integrated packet networks," *IEEE Trans. Commun.*, vol. COM-28, pp. 325-333, Mar. 1980.

[24] M. Vetterli, "Multi-dimensional sub-band coding: Some theory and algorithms," *Signal Processing*, vol. 6, pp. 97-112, Feb. 1984.

[25] H. G. Musmann *et al.*, "Advances in picture coding," *Proc. IEEE*, vol. 73, pp. 523-548, Apr. 1985.

[26] S-Q Li, "Study of packet loss in packet voice systems," *IEEE Trans. Commun.*, to be published.

[27] R. E. Blahut, *Theory and Practice of Error Control Codes.* Reading, MA: Addison-Wesley, 1983.

[28] G. Karlsson, "Sub-band coding for packet video," Ph.D. dissertation, Dep. Elec. Eng., Columbia University, May 1989.

[29] J-Y Cochennec, private communication.

[30] D. J. Le Gall and A. Tabatabai, "Sub-band coding of digital images using symmetric short kernel filters and arithmetic coding techniques," in *Proc. ICASSP 88*, New York, Apr. 1988, pp. 761-764.

[31] D. J. Le Gall and H. Gaggioni, "Transmission of HDTV signals under 140 Mbit/s using subband decomposition and discrete transform coding," in *Proc. Second Int. Workshop Signal Proc. HDTV*, L'Aquila, Italy, Feb. 1988, paper 64.

Gunnar Karlsson (S'89) was born in Jönköping, Sweden. He received the M.S. degree in electrical engineering from Chalmers University of Technology, Gothenburg, Sweden, in 1983 and the Ph.D. in electrical engineering from Columbia University, New York, NY, May 1989.

During the academic year 1982-1983 he was studying at the University of Massachusetts on a Fulbright scholarship. He did the Master's thesis at Telefonaktiebolaget LM Ericsson, Stockholm, Sweden, during the summer 1983. In 1984, he was working as a hardware designer at Saab Training Systems, Huskvarna, Sweden. During the summers in 1985 and 1986 he did research at Bell Communications Research, Morristown, NJ. From 1985 to 1988 he held positions as a Teaching Assistant at the Department of Electrical Engineering and a Research Assistant at the Center for Telecommunications Research at Columbia University. In January 1989, he joined IBM Zürich Research Laboratory as a Research Staff Member.

Martin Vetterli (S'86-M'86) was born in Switzerland in 1957. He received the Dipl. El.-Ing. degree from the Eidgenösssische Technische Hochschule Zürich, Switzerland, in 1981, the Master of Science degree from Stanford University, Stanford, CA, in 1982, and the Doctorat ès Science degree from the Ecole Polytechnique Fédérale de Lausanne, Switzerland, in 1986.

In 1982, he was a Research Assistant with the Computer Science Department of Stanford University, and from 1983 to 1986 he was a Researcher at the Ecole Polytechnique. He has worked for Siemens, Switzerland, and AT&T Bell Laboratories, Holmdel, NJ. Since 1986, he has been at Columbia University in New York, first with the Center for Telecommunications Research and now with the Department of Electrical Engineering where he is currently an Assistant Professor.

Dr. Vetterli is member of the editorial board of *Signal Processing* and served as European Liaison for ICASSP '88 in New York. He was recipient of the Best Paper Award of EURASIP in 1984 and of the Research Prize of the Brown Bovery Corporation (Switzerland) in 1986. His research interests include multirate signal processing, computational complexity, algorithm design for VLSI, and signal processing for telecommunications.