

CHAPTER 2

Design Considerations for Packet Video

2.1 Variable rate coding and transmission

2.1.1 Video quality and channel utilization

2.1.2 Interdependence between coder and network

2.2 Hierarchical source coding

2.3 Error recovery

2.3.1 Error control coding

2.3.2 Error concealment through the use of visual redundancy

2.4 Video resynchronization

2.4.1 Network delays

2.4.2 Unequal clock frequencies

2.5 Summary

The absence of a fixed-capacity channel structure makes packet-switched transmission systems a dramatically different environment for video coding and transmission than the traditional circuit-switched systems. This chapter outlines the general design considerations that this new environment imposes on video coding and transmission. Special attention is given to the signal processing issues of source coding, error recovery and resynchronization. First, the pros and cons of variable rate video coding and transmission are discussed. Second, an attractive approach to source coding by means of separating the signal into a hierarchy of sub-signals is described. Third, remedies to the issue of packet loss are considered, and, last, the synchronization constraints on video are discussed as pertaining to variable delays and lack of common time-reference for source and receiver. (In Chapter 3 we will consider these issues in the context of a network architecture model.)

2.1 Variable rate coding and transmission

A packet network may efficiently accommodate sources with varying output rates since it does not allocate a fixed quota of network resources to each source. This is a most important issue for packet video, and one that distinguishes it from circuit-switched video. In the first section we will discuss the advantages of such transmission regarding received video quality and utilization of channel capacity. The following section brings forth the associated complications pertaining to a strong linkage between signal processing and transmission functions.

2.1.1. Video quality and channel utilization

The amount of information in a video signal depends on the activity of the captured scene. When compressed, the resulting bit rate may be highly varying, with a bursty behavior. The bursts correspond to some activity in the captured scene and they may therefore last several seconds (see Chapter 6). The concept of source coding which produces a variable bit rate will be referred to as variable bit rate coding. It is worth noting that most common coding methods produce a variable output rate. Only when they are aimed at a channel with less capacity than the source's peak rate, are the rate variations minimized by a buffer placed between source and network (Fig. 2.1 (a)). Knowledge of the statistical behavior of variable rate coded video will be necessary for network performance evaluation. However, modeling these signals is difficult, especially if the model should lend itself to tractable queueing analysis. A brief survey of modeling attempts is given in Appendix A.

From a transmission point of view, delivery of packets within a bounded time delay, thus providing real-time service, poses a difficult resource allocation and control problem when the sources provide high and varying rates [LAZ87]. Fixed

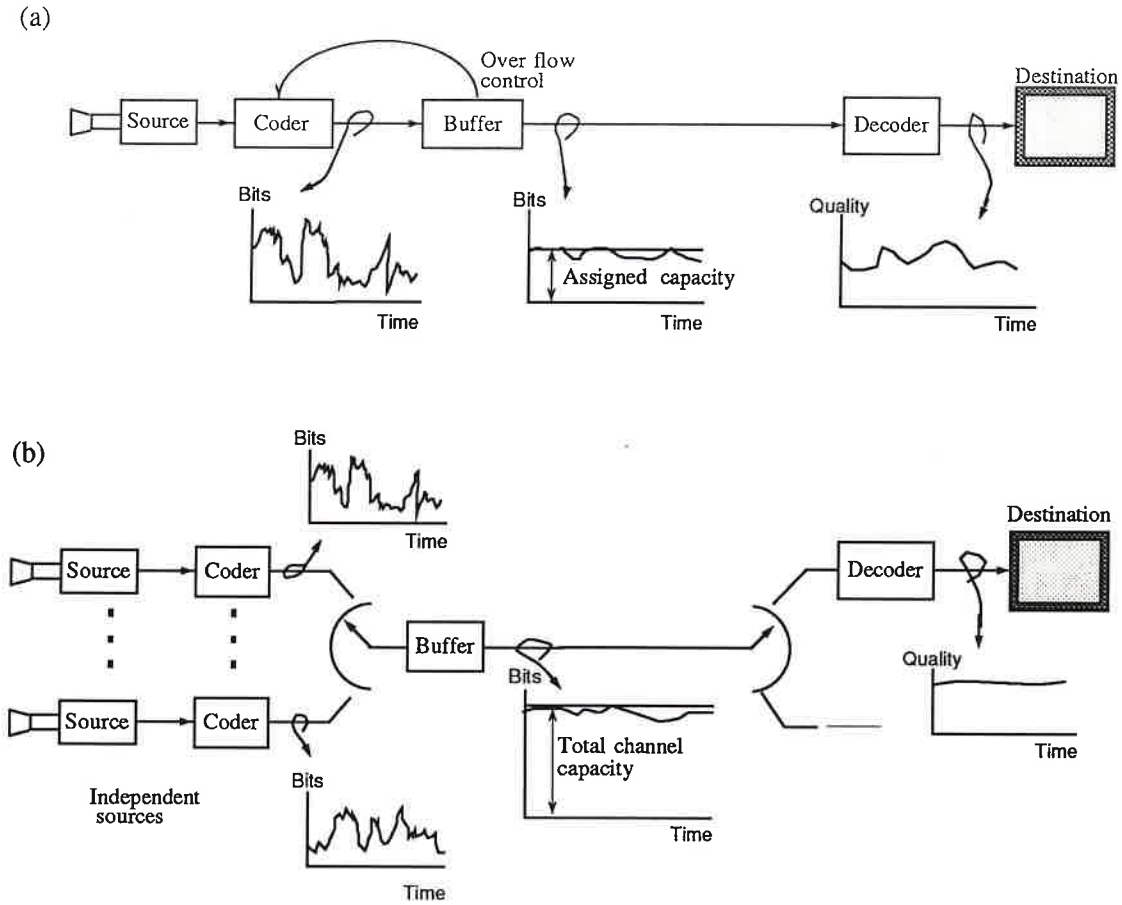


Figure 2.1 Multiplexing in circuit-switching and packet-switching: (a) In a circuit-switched system the multiplexing is done by assigning a fixed amount of capacity for each transmitter. (b) For a packet-switched system, there is no predetermined capacity assignment for the transmitter, and with independent sources the transmitted signals are statistically multiplexed.

allocation of resources, which could eliminate congestion based packet loss, would waste channel capacity since the assignment has to equal the maximum transmission rate. Generally, this peak rate could be reduced by buffering the signal. Due to the burstiness of the signal, that would require a buffer of substantial size which, in turn, would introduce unacceptable delays. The utilization of assigned capacity therefore remains poor since the peaks of the transmission rate cannot be sufficiently smoothed out under a real-time constraint. This is illustrated in Fig. 2.1 (a). For instance, Verbiest *et alii* [VER88c] show that for a single source, large fluctuations

still exists after integration times of up to 500 frames.

Given a buffer with a size restricted to meet a real-time limit, there will be nonnegligible probabilities of buffer overflow. To reduce the occurrence of overflow, a feedback mechanism is put in place, which enforces a higher compression when the amount of data in the buffer exceeds some prescribed threshold. In a video coding context, the overflow control degrades the image quality during times when the coder is forced to use a higher compression mode, as during periods of motion in the video scene. The video quality at the receiver may therefore fluctuate, where the variations are correlated with the activity in the captured scene. We shall refer to this scenario as constant-rate / varying-quality transmission.

By sharing the total channel capacity among the users statistically, packet-switching may yield a higher channel utilization than what is possible with fixed allocation. When the varying rates of all video sessions are added in the channel, as shown in Fig. 2.1 (b), the total rate exhibits less burstiness as a result of the statistical multiplexing that takes place (assuming independent sources) [HAS72, KOG81, MAG88, DOU88, VER88c, VER88d]. As the number of sources increases, the variance of the added rate goes towards zero. Thus, the actual use of the network resources per source tends towards the source's average transmission rate, which should be compared to the near-maximum rate in the fixed-allocation case. However, the absence of packet loss cannot be guaranteed for a finite number of sources since the total rate still exhibits variations. As shown in Fig. 2.1 (b), when the total rate of all sources exceeds the network capacity, packets are lost (assuming full network buffers). Any quality variation would be caused by an overloaded network and, thus, might not be correlated to the signal. For the video quality, it is expected that the receiver will perceive a constant video quality under

normal network loading conditions since a source transmits with a rate reflecting the signal's information contents [VER86, LAZ87, OHT88a, KAR88b, VER88b, NOM89]. Analogous to the fixed allocation case, we shall refer to this as variable-rate / constant-quality transmission.

Of the two main advantages with variable rate transmission of video, the constant quality aspect is most important for low bit-rate / low-quality services. For low bit-rate simulations, Nomura *et alii* [NOM89] report an average quality gain with variable rate over constant rate of 2-4 dB and an increase of the minimum quality as high as 5-10 dB. Broadcast quality video has to appear unimpaired regardless of transmission mode. Consequently, there is no quality improvement to be gained. However, the channel utilization is improved by statistical multiplexing. For example, Verbiest *et alii* [VER88c] show that 16 sources with peak rates of 14 Mbps, which in a fixed allocation case would require 224 Mbps, only reach that capacity level with a probability of 10^{-48} . Instead the probability of exceeding a capacity-level of 6.8 Mbps per source is only 10^{-8} when the 16 sources are statistically multiplexed. This result agrees with our simulation results for video telephony scenes, as reported in Section 6.3. Finally, it should be noted that a variable-rate coder is less complex than its fixed-rate counter part because it does not need various compression modes nor an output buffer.

Note that variable rate transmission may benefit any information source for which the varying rate cannot be sufficiently smoothed out by buffering. This includes non-stationary sources, which do not have a constant time-average, and stationary sources with a delay constraint, as previously discussed.

2.1.2 Interdependence between coder and network

In the case of variable-rate transmission of real-time information, we can no longer assume the traditional separation of source, channel and receiver as a valid paradigm (the separation is illustrated in Fig. 2.2 (a) [VIT79]). We have seen that the video coder will require various amounts of capacity over time but that the packet network will provide a channel whose capacity changes depending on the total load of the network. The decoder, in turn, has to function together with the channel where packets are lost or delayed depending on the total network load. Also, the decoding rate might differ from that of the encoder since there is no common time reference in the system [COC85, VER86, VER88b, VER88d]. Hence, there are strong dependencies between the entities involved, as depicted in Fig. 2.2 (b), which requires 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 [CHI84a, CHI84b, LAZ87, KAR89a]. Such a model will be considered in Chapter 3.

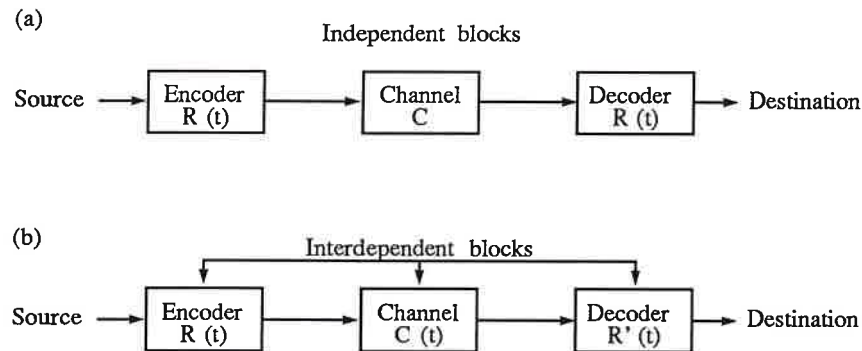


Figure 2.2 Transmission of variable rate signals over (a) a circuit-switched network and (b) a packet-switched network. $R(t)$ is the coder output rate, C is the assigned capacity, while $C(t)$ is the available capacity, and $R'(t)$ is the rate of the decoder. $R(t)$ and $R'(t)$ might differ since there is no common time reference in the system.

Specifically, the interactions may be seen as between signal processing issues and transmission issues (see Fig. 2.3). The behavior of variable-rate coded video will mainly 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.

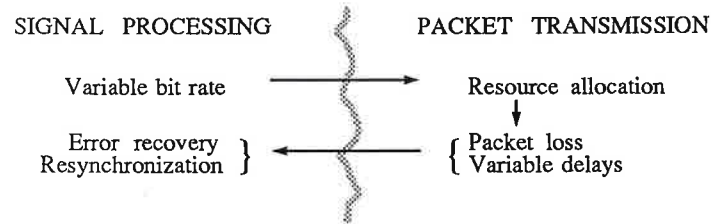


Figure 2.3 The interactions between signal processing issues and transmission issues.

2.2 Hierarchical source coding

Hierarchical coding, known also as layered coding or embedded coding, is a technique first developed for packet voice transmission [BAI80], wherein a signal is separated into sub-signals of various importance to be coded and transmitted separately. This general technique, illustrated in Fig. 2.4, may be advantageously extended to video coding and transmission as well [KAR87, KAR88b, VER88b, VER88d, KAR89a]. In this way, the coding of a sub-signal can be tailored to the information it carries and the sub-signal can be transmitted with priority or capacity guarantees which reflects its importance. Network congestion, where buffer overflow leads to packet discarding, will mostly affect the sub-signals 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 [KAR87, KAR88b, OHT88a, NOM89]. This refers to the fact that high video quality can be traded for lower transmission cost. Note,

however, that separate encoding of each sub-signal may be sub-optimal if the sub-signals are cross-correlated. In our simulations, we have found a potential problem with hierarchical coding: if some of the compressed sub-signals yield low output rates, the packetization delay may become intolerable for fixed-length packets (see Chapter 6). The possible solutions would be to multiplex low bit-rate sub-signals, or the sender could avoid transmission of a packet that required more than the allowed time for its packetization.

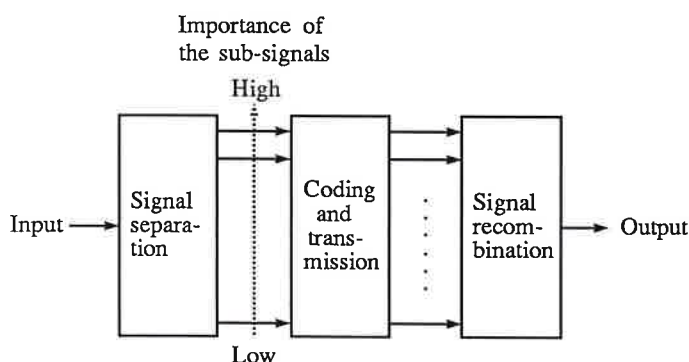


Figure 2.4 A hierarchical coding system. The signal is separated into sub-signals of various importance which may be coded and transmitted independently of one another. When received and decoded, the sub-signals are recombined to form the output signal.

In our definition of functionality, the process of separating (and recombining) the input signal does not include the compression of the resulting sub-signals. The signal separation ought to be such that the total data rate of the sub-signals should equal the rate of 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 sub-signals. The better methods are those which also lead to improved overall compression. Alternatives include bit-plane separation [BAI80], subband analysis/synthesis [VET84]; and unitary transforms [MUS85].

Bit-plane separation means that the most important information consists of the most-significant bits of the image, and, progressively down through the bit-planes, the least important sub-signal is composed of the least significant bits. This separation has the peculiarity that the most important sub-signal 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 [KAR87, KAR88a, KAR88b] since the subbands have a natural hierarchy. The use of subband coding as hierarchical coding is presented in this thesis. For unitary block-transforms, common block sizes lead to 64 and 256 channels respectively, which are prohibitively many for independent transmission and would thus require some multiplexing. Since a transform is applied on sub-blocks of the image, all transform coefficients with the same index would be gathered from the sub-blocks to form a sub-signal (commonly known as Mandala reordering). Zonal encoding may thereby eliminate entire sub-signals by not allocating them any bits.

Protocol functions, which are often implemented in software, may not be executable at the speed necessary to handle encoded video as a single bit stream. Thus, if the hierarchical coder is implementable in parallel it will enable encoding, decoding and protocol functions to run at a lower rate. That, in turn, simplifies their implementation.

2.3 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 uncorrelated resulting from noise in the transmission channel [SCH87]. Packet loss due to network congestion is not so straight forward; 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 [LI88].

Error recovery has to achieve robustness so that a video session is never seriously disrupted when packets are lost, and the recovery method ought to minimize the quality degradation of such loss. There are mainly two approaches to error recovery: firstly, the use of error control codes, and secondly, error concealment through the 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).

2.3.1 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. 2.5 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) [LEE87]. The error control

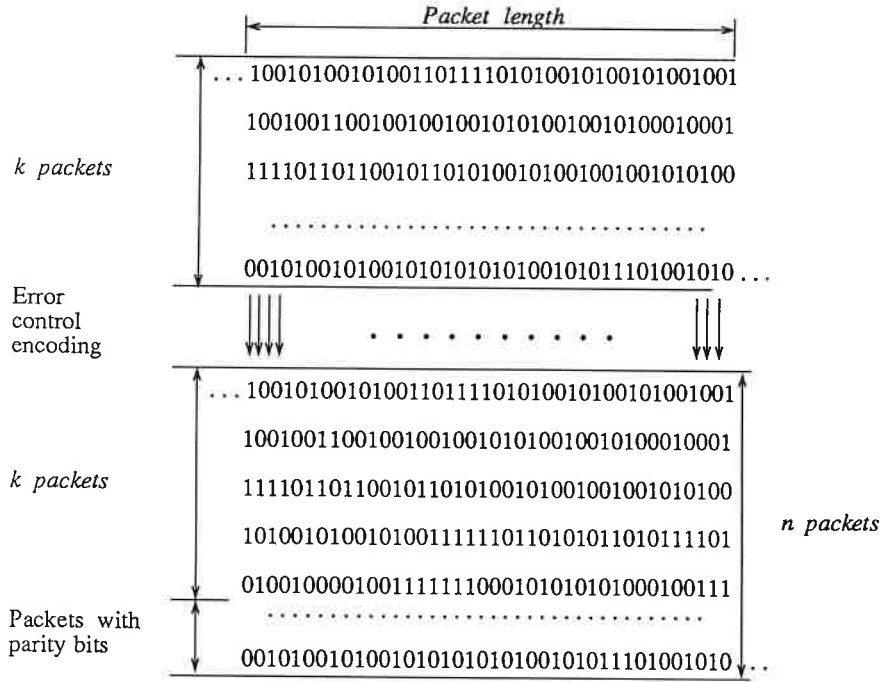


Figure 2.5 Error correction encoding should be applied perpendicular to the segmented bit stream to achieve bit-interleaving. The figure illustrates the case when a systematic block-code is used.

codes used can be block codes or convolutional codes [BLA83]. 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. Also, the received packets can still be used even if the missing packets are too many for the code to reconstruct.

For a (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. For a constant rate data stream with rate R_c , the introduced delay can be written as

$$t_i = \frac{P}{R_c} (i \bmod k) + t_{enc},$$

where t_i denotes the delay of the i th packet, P is the size of the packet's information field, k the number of information symbols of the code, and t_{enc} is the time needed

to perform the encoding on a block of k packets. This delay can however 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, thus the number of lost packets can exceed 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.

Finally, we wish to point out that lost packets have to be detected at the receiver. Otherwise, a lost packet would place the channel encoder and decoder out of synchronization. However, this means that the locations of errors are known so that a code with a minimum distance, d^* , can correct up to $d^* - 1$ errors as opposed to $\lfloor \frac{d^* - 1}{2} \rfloor$ in the general case when the locations are unknown ($\lfloor x \rfloor$ gives the integer part of x).

2.3.2 Error concealment through the use of visual redundancy

Error concealment through the use of visual redundancy means that lost data are detected at the receiver and the missing area is concealed by data computed by signal processing means from the received information. This type of error recovery takes place after the source decoding but before the signal recombination of the hierarchical source coding system (see Fig. 2.4). The general requirement for the method is that lost values can be treated as erasures *i.e.*, the number of lost values and their locations are known [KAR87, KAR88b, KAR89a]. This means, first, that there are points in the transmitted bit-stream where the decoder can resume the decoding after a loss has occurred and, second, the spatial locations (within

a video-image) of these points are known. In general, it is easier to obtain the erasure-property with FIR-based methods, such as transform and subband coding, than with recursive coding methods (*e.g.*, DPCM). For DPCM, consider that the decoding can only resume after a loss if the prediction loops is set to the value of the restart-point. This is feasible only for the simplest first-order intraframe predictor, which only has a one-point history. On the other hand, it has poor coding performance. Alternatively for DPCM, the prediction loop could be made “leaky” so that the error decays with time. Again, this reduces the compression. Section 3.2 presents how the erasure property may be obtained.

An erased area of an image may be approximately concealed through spatial and temporal interpolation [CRO83], or statistical image reconstruction methods [VEL88]. 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. In connection with the hierarchical source coding, it is advantageous not to scan out, and later packetize, all sub-signals along the same direction (usually row-order). For instance, if every other sub-signal is scanned out vertically and the others horizontally, lost data may be partly masked by the data in the received sub-signals.

Regardless of the success of the concealment, in a perceptual sense, the limited error propagation guarantees that the session never need be terminated for any amount of lost packets. The performance of error concealment through the use of visual redundancy as indicated by simulation results in Chapter 6 is encouraging.

2.4 Video resynchronization

The timing problems of packet video are twofold. Firstly, there are variations in transmission delays, referred to as packet delay jitter. Analogous to packet loss, these variations are an inherent part of packet transmission. In fact, if the network queues were infinitely long, only packet delay jitter would occur. Secondly, there is no common time reference for the transmitter and the receiver. The asynchronous nature of packet networks implies that clock frequencies at the sender and the receiver may be different.

2.4.1 Network delays

Depending on the number of buffers traversed by a virtual circuit and the occupancy of those buffers, transmission delays can vary drastically over time and from one session to another. The difference in delays complicates the reconstruction of a synchronous video signal, so the signal must go through a resynchronization process. 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. 2.6.

Transmission delays are irrelevant for one-way sessions, such as broadcast television. Consequently, the maximum amount of delay-jitter that can be experienced in the network may be eliminated by a large buffer. However, for high bit-rates and substantial jitter this might require unreasonably large buffer size. A buffer, limited to an economically reasonable size, will not be able to absorb the higher amounts of delay variations and, therefore, discarding of packets must be accepted when they arrive too late for their play-out time.

For video telephony (video and voice), in contrast, short end-to-end delays are of

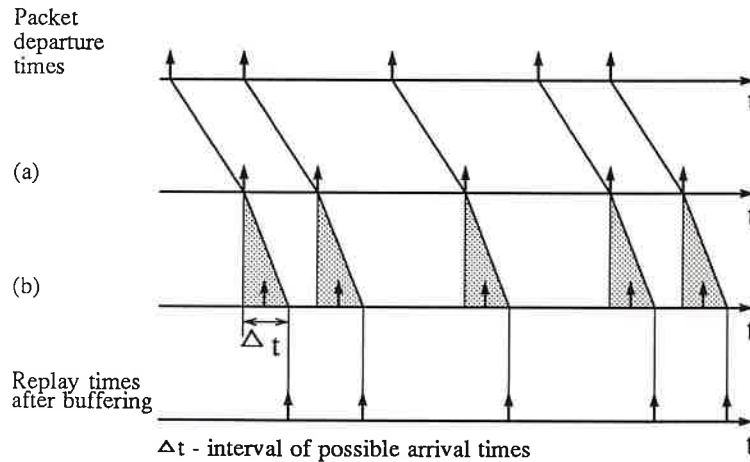


Figure 2.6 The delays in a packet-switched network are composed of (a) a fixed part due to the propagation delay, and (b) a variable (jitter) part which is due to waiting time in buffers.

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 for this application, 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 [COC88]. Priorities used in connection with hierarchical coding will result in lower average delay for high priority sub-signals at the cost of extended delay for low priority ones. Thus, when a delay limit is enforced at the decoder, a high priority class will yield fewer outstanding packets to be disregarded than a low priority class. Thereby the desired behavior of the hierarchical coding is met, in that the least important sub-signals are degraded first [YIN88].

2.4.2 Unequal clock frequencies

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. 2.7 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 [COC85, VER86, VER88b]. This can be done by monitoring the level of the input buffer [COC85], or by synchronizing the receiver clock to time information in the packets (by use of a phase-locked loop) [VER86, VER88b]. Both methods are however complicated by packet delay jitter and, if used, variable rate coding.

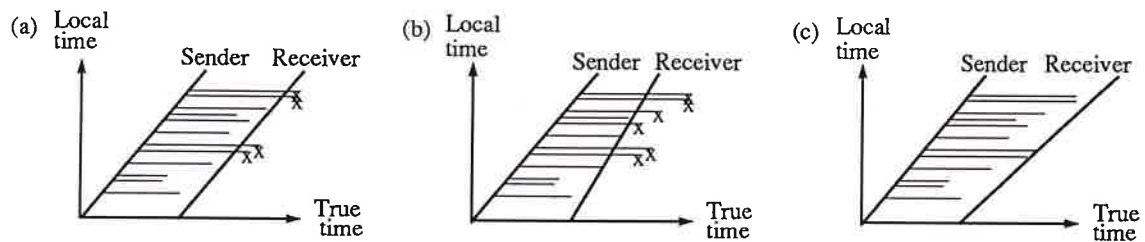


Figure 2.7 In (a) the clocks have equal frequencies, and only the packets which are unduly delayed by the network are considered lost (marked by x). In (b) the receiver clock is fast compared to the sender, thus more packets are considered lost. If the receiver clock is slow, as in (c), too much time is allowed for arriving packets.

There are cases when the receiver clock cannot be adapted to the received data rate. For example, when video is received from more than one sender (as with multi-drop in Section 3.3.1), 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 play-out rate. In the simplest case, this is performed by occasionally repeating or skipping frames. For larger clock discrepancies, the frame rate conversion would require temporal filtering of the image sequence, preferably motion compensated (*e.g.*, see [REU89]), to eliminate temporal discontinuities. For the sake of illustration, assume that only one frame repeat (or frame skip) can be allowed for any hour of a session. This means that the drift of each of the two clocks has to stay within ± 2.5 minutes over a year, which can easily be met. So, combined with clock accuracy of this

magnitude, frame rate conversion (*i.e.*, temporal up- and subsampling) may be a viable resynchronization method which could be employed for all types of session.

2.5 Summary

In the previous parts we have discussed the issues which are highly important to consider in the design of a packet video system. We have made several points which will be important for the material in the chapters to follow. These are summarized below.

Variable rate transmission and statistical multiplexing:

- It enables constant quality sessions of highly compressed video.
- It can yield increased channel utilization.
- The encoder complexity is reduced compared to the multi-mode coders used with circuit-switched systems.
- Dependencies are introduced between source, channel and receiver.

Hierarchical source coding:

- A signal is split into a hierarchy of sub-signals of various importance.
- Error control is simplified since loss of important information can be prevented.
- Hierarchical source coding enables cost / quality tuning.
- The signal separation and recombination should be lossless and preserve the bit rate of the input.
- The signal separation should lead to improved compression and perceptual quality.
- The coder should be implementable in parallel.

- Some sub-signals may have unduly long packetization delays.

Error recovery:

- The best solution is a combination of error control codes and error concealment.
- Error control coding should be carried out by bit interleaving and systematic codes.
- Error concealment through the visual redundancy relies on limited error propagation and known locations of missing values.
- Error concealment is computed from received information by interpolation.
- With error concealment, a session is never interrupted for any amount of packet loss.

Resynchronization:

- Delay jitter can be removed by buffering.
- End-to-end delay is not of importance for one-way sessions.
- For two-way sessions, video delay must be bound to provide lip-synchronization.
- Unequal clock frequencies can be accommodated by repeating and skipping video frames in combination with temporal filtering.
- Alternatively, the transmitter clock can be deduced from the arriving packets.

In the next chapter we will consider these issues in the context of a network architecture model. Thereby their implementation can be considered with the aim of minimizing dependence on the video-format.