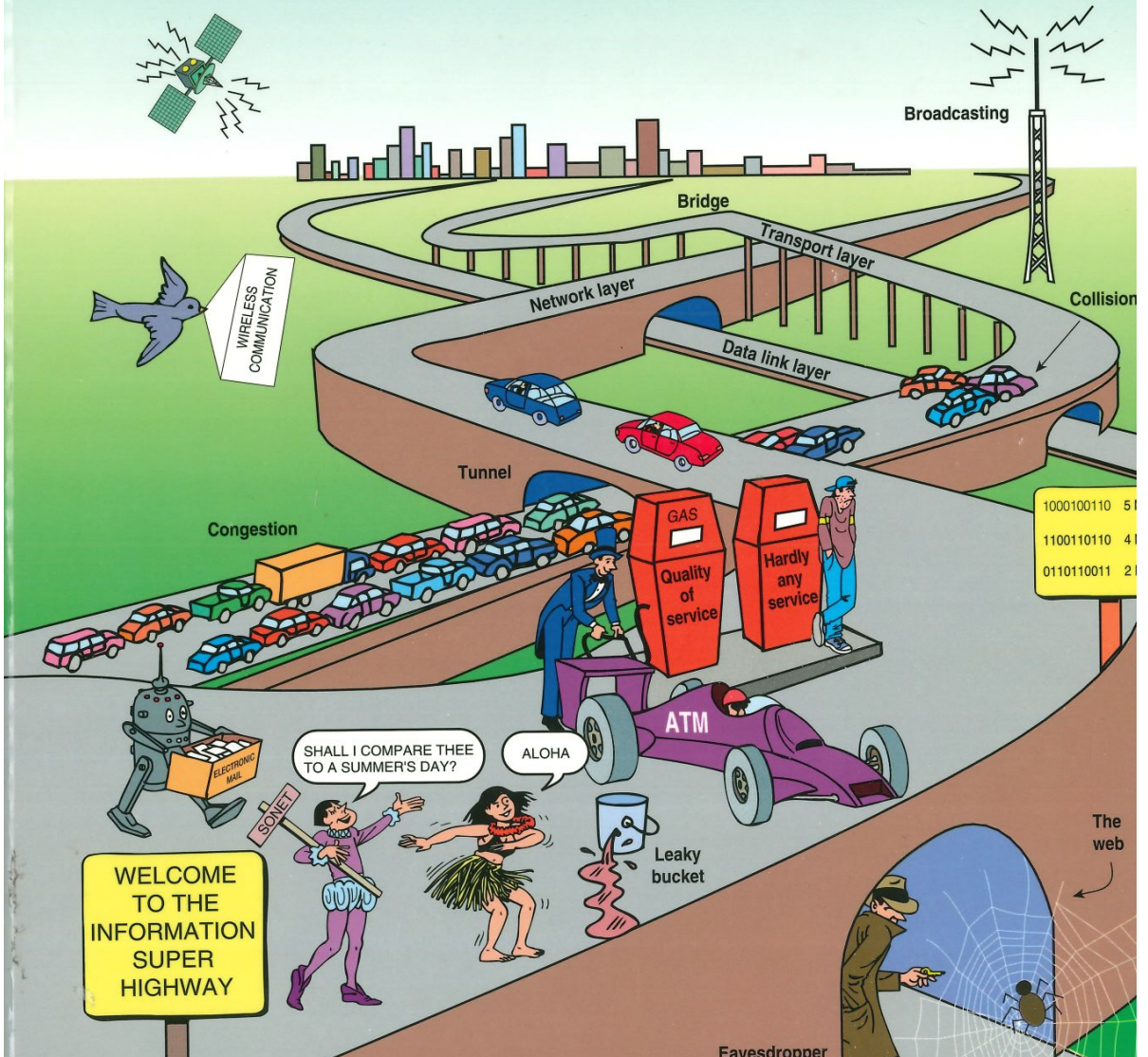


THIRD EDITION
COMPUTER NETWORKS

ANDREW S. TANENBAUM



Computer Networks

Third Edition

Andrew S. Tanenbaum

*Vrije Universiteit
Amsterdam, The Netherlands*

For book and bookstore information



<http://www.prenhall.com>



Prentice Hall PTR
Upper Saddle River, New Jersey 07458

Library of Congress Cataloging in Publication Data

Tanenbaum, Andrew S. 1944-

Computer networks / Andrew S. Tanenbaum. -- 3rd ed.

p. cm.

Includes bibliographical references and index.

ISBN 0-13-349945-6

1. Computer networks. I. Title.

TK5105.5.T36 1996

96-4121

004.6--dc20

CIP

Editorial/production manager: *Camille Trentacoste*

Interior design and composition: *Andrew S. Tanenbaum*

Cover design director: *Jerry Votta*

Cover designer: *Don Martinetti, DM Graphics, Inc.*

Cover concept: *Andrew S. Tanenbaum, from an idea by Marilyn Tremaine*

Interior graphics: *Hadel Studio*

Manufacturing manager: *Alexis R. Heydt*

Acquisitions editor: *Mary Franz*

Editorial Assistant: *Noreen Regina*



© 1996 by Prentice Hall PTR

Prentice-Hall, Inc.

A Simon & Schuster Company

Upper Saddle River, New Jersey 07458

The publisher offers discounts on this book when ordered in bulk quantities. For more information, contact:

Corporate Sales Department, Prentice Hall PTR, One Lake Street, Upper Saddle River, NJ 07458.
Phone: (800) 382-3419; Fax: (201) 236-7141. E-mail: corpsales@prenhall.com

All rights reserved. No part of this book may be reproduced, in any form or by any means, without permission in writing from the publisher.

All product names mentioned herein are the trademarks of their respective owners.

Printed in the United States of America

10 9 8 7 6 5 4

ISBN 0-13-349945-6

Prentice-Hall International (UK) Limited, *London*

Prentice-Hall of Australia Pty. Limited, *Sydney*

Prentice-Hall Canada Inc., *Toronto*

Prentice-Hall Hispanoamericana, S.A., *Mexico*

Prentice-Hall of India Private Limited, *New Delhi*

Prentice-Hall of Japan, Inc., *Tokyo*

Simon & Schuster Asia Pte. Ltd., *Singapore*

Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*

The Network Layer

The **network layer** is concerned with controlling the operation of the subnet. A key design issue is determining how packets are routed from source to destination. Routes can be based on static tables that are “wired into” the network and rarely changed. They can also be determined at the start of each conversation, for example a terminal session. Finally, they can be highly dynamic, being determined anew for each packet, to reflect the current network load.

If too many packets are present in the subnet at the same time, they will get in each other’s way, forming bottlenecks. The control of such congestion also belongs to the network layer.

Since the operators of the subnet may well expect remuneration for their efforts, there is often some accounting function built into the network layer. At the very least, the software must count how many packets or characters or bits are sent by each customer, to produce billing information. When a packet crosses a national border, with different rates on each side, the accounting can become complicated.

When a packet has to travel from one network to another to get to its destination, many problems can arise. The addressing used by the second network may be different from the first one. The second one may not accept the packet at all because it is too large. The protocols may differ, and so on. It is up to the network layer to overcome all these problems to allow heterogeneous networks to be interconnected.

In broadcast networks, the routing problem is simple, so the network layer is often thin or even nonexistent.

The Transport Layer

The basic function of the **transport layer** is to accept data from the session layer, split it up into smaller units if need be, pass these to the network layer, and ensure that the pieces all arrive correctly at the other end. Furthermore, all this must be done efficiently, and in a way that isolates the upper layers from the inevitable changes in the hardware technology.

Under normal conditions, the transport layer creates a distinct network connection for each transport connection required by the session layer. If the transport connection requires a high throughput, however, the transport layer might create multiple network connections, dividing the data among the network connections to improve throughput. On the other hand, if creating or maintaining a network connection is expensive, the transport layer might multiplex several transport connections onto the same network connection to reduce the cost. In all cases, the transport layer is required to make the multiplexing transparent to the session layer.

The transport layer also determines what type of service to provide the session

layer, and ultimately, the users of the network. The most popular type of transport connection is an error-free point-to-point channel that delivers messages or bytes in the order in which they were sent. However, other possible kinds of transport service are transport of isolated messages with no guarantee about the order of delivery, and broadcasting of messages to multiple destinations. The type of service is determined when the connection is established.

The transport layer is a true end-to-end layer, from source to destination. In other words, a program on the source machine carries on a conversation with a similar program on the destination machine, using the message headers and control messages. In the lower layers, the protocols are between each machine and its immediate neighbors, and not by the ultimate source and destination machines, which may be separated by many routers. The difference between layers 1 through 3, which are chained, and layers 4 through 7, which are end-to-end, is illustrated in Fig. 1-16.

Many hosts are multiprogrammed, which implies that multiple connections will be entering and leaving each host. There needs to be some way to tell which message belongs to which connection. The transport header (H_4 in Fig. 1-11) is one place this information can be put.

In addition to multiplexing several message streams onto one channel, the transport layer must take care of establishing and deleting connections across the network. This requires some kind of naming mechanism, so that a process on one machine has a way of describing with whom it wishes to converse. There must also be a mechanism to regulate the flow of information, so that a fast host cannot overrun a slow one. Such a mechanism is called **flow control** and plays a key role in the transport layer (also in other layers). Flow control between hosts is distinct from flow control between routers, although we will later see that similar principles apply to both.

The Session Layer

The session layer allows users on different machines to establish **sessions** between them. A session allows ordinary data transport, as does the transport layer, but it also provides enhanced services useful in some applications. A session might be used to allow a user to log into a remote timesharing system or to transfer a file between two machines.

One of the services of the session layer is to manage dialogue control. Sessions can allow traffic to go in both directions at the same time, or in only one direction at a time. If traffic can only go one way at a time (analogous to a single railroad track), the session layer can help keep track of whose turn it is.

A related session service is **token management**. For some protocols, it is essential that both sides do not attempt the same operation at the same time. To manage these activities, the session layer provides tokens that can be exchanged. Only the side holding the token may perform the critical operation.

FTTH should be the goal from the beginning is a matter of some debate within the telephone industry.

An alternative design using the existing cable TV infrastructure is shown in Fig. 2-23(b). Here a multidrop cable is used instead of the point-to-point system characteristic of the telephone system. It is likely that both Fig. 2-23(a) and Fig. 2-23(b) will coexist in the future, as telephone companies and cable TV operators become direct competitors for voice, data, and possibly even television service. For more information about this topic, see (Cook and Stern, 1994; Miki, 1994b; and Mochida, 1994).

2.4.4. Trunks and Multiplexing

Economies of scale play an important role in the telephone system. It costs essentially the same amount of money to install and maintain a high-bandwidth trunk as a low-bandwidth trunk between two switching offices (i.e., the costs come from having to dig the trench and not from the copper wire or optical fiber). Consequently, telephone companies have developed elaborate schemes for multiplexing many conversations over a single physical trunk. These multiplexing schemes can be divided into two basic categories: **FDM (Frequency Division Multiplexing)**, and **TDM (Time Division Multiplexing)**. In FDM the frequency spectrum is divided among the logical channels, with each user having exclusive possession of some frequency band. In TDM the users take turns (in a round robin), each one periodically getting the entire bandwidth for a little burst of time.

AM radio broadcasting provides illustrations of both kinds of multiplexing. The allocated spectrum is about 1 MHz, roughly 500 to 1500 kHz. Different frequencies are allocated to different logical channels (stations), each operating in a portion of the spectrum, with the interchannel separation great enough to prevent interference. This system is an example of frequency division multiplexing. In addition (in some countries), the individual stations have two logical subchannels: music and advertising. These two alternate in time on the same frequency, first a burst of music, then a burst of advertising, then more music, and so on. This situation is time division multiplexing.

Below we will examine frequency division multiplexing. After that we will see how FDM can be applied to fiber optics (wavelength division multiplexing). Then we will turn to TDM, and end with an advanced TDM system used for fiber optics (SONET).

Frequency Division Multiplexing

Figure 2-24 shows how three voice-grade telephone channels are multiplexed using FDM. Filters limit the usable bandwidth to about 3000 Hz per voice-grade channel. When many channels are multiplexed together, 4000 Hz is allocated to each channel to keep them well separated. First the voice channels are raised in

frequency, each by a different amount. Then they can be combined, because no two channels now occupy the same portion of the spectrum. Notice that even though there are gaps (guard bands) between the channels, there is some overlap between adjacent channels, because the filters do not have sharp edges. This overlap means that a strong spike at the edge of one channel will be felt in the adjacent one as nonthermal noise.

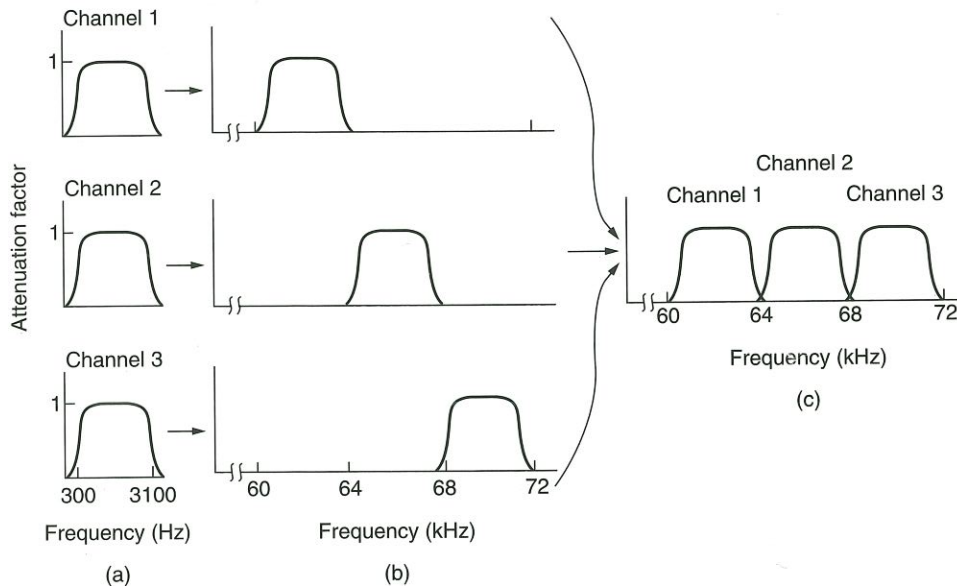


Fig. 2-24. Frequency division multiplexing. (a) The original bandwidths. (b) The bandwidths raised in frequency. (c) The multiplexed channel.

The FDM schemes used around the world are to some degree standardized. A widespread standard is 12 4000-Hz voice channels (3000 Hz for the user, plus two guard bands of 500 Hz each) multiplexed into the 60 to 108 kHz band. This unit is called a **group**. The 12- to 60-kHz band is sometimes used for another group. Many carriers offer a 48- to 56-kbps leased line service to customers, based on the group. Five groups (60 voice channels) can be multiplexed to form a **super-group**. The next unit is the **mastergroup**, which is five supergroups (CCITT standard) or ten supergroups (Bell system). Other standards up to 230,000 voice channels also exist.

Wavelength Division Multiplexing

For fiber optic channels, a variation of frequency division multiplexing is used. It is called **WDM (Wavelength Division Multiplexing)**. A simple way of achieving FDM on fibers is depicted in Fig. 2-25. Here two fibers come together

at a prism (or more likely, a diffraction grating), each with its energy in a different band. The two beams are passed through the prism or grating, and combined onto a single shared fiber for transmission to a distant destination, where they are split again.

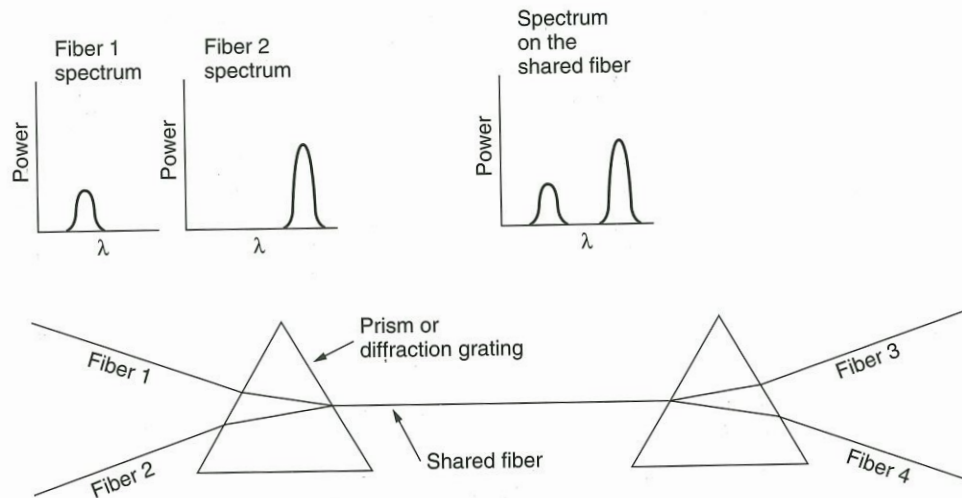


Fig. 2-25. Wavelength division multiplexing.

There is really nothing new here. As long as each channel has its own frequency range, and all the ranges are disjoint, they can be multiplexed together on the long-haul fiber. The only difference with electrical FDM is that an optical system using a diffraction grating is completely passive, and thus highly reliable.

It should be noted that the reason WDM is popular is that the energy on a single fiber is typically only a few gigahertz wide because it is currently impossible to convert between electrical and optical media any faster. Since the bandwidth of a single fiber band is about 25,000 GHz (see Fig. 2-6), there is great potential for multiplexing many channels together over long-haul routes. A necessary condition, however, is that the incoming channels use different frequencies.

A potential application of WDM is in the FTTC systems described earlier. Initially, a telephone company could run a single fiber from an end office to a neighborhood junction box where it met up with twisted pairs from the houses. Years later, when the cost of fiber is lower and the demand for it is higher, the twisted pairs can be replaced by fiber and all the local loops joined onto the fiber running to the end office using WDM.

In the example of Fig. 2-25, we have a fixed wavelength system. Bits from fiber 1 go to fiber 3, and bits from fiber 2 go to fiber 4. It is not possible to have bits go from fiber 1 to fiber 4. However, it is also possible to build WDM systems that are switched. In such a device, there are many input fibers and many output

fibers, and the data from any input fiber can go to any output fiber. Typically, the coupler is a passive star, with the light from every input fiber illuminating the star. Although spreading the energy over n outputs dilutes it by a factor n , such systems are practical for hundreds of channels.

Of course, if the light from one of the incoming fibers is at 1.50206 microns and potentially might have to go to any output fiber, all the output fibers need tunable filters so the selected one can set itself to 1.50206 microns. Such optical tunable filters can be built from Fabry-Perot or Mach-Zehnder interferometers. Alternatively, the input fibers could be tunable and the output ones fixed. Having both be tunable is an unnecessary expense and is rarely worth it.

Time Division Multiplexing

Although FDM is still used over copper wires or microwave channels, it requires analog circuitry and is not amenable to being done by a computer. In contrast, TDM can be handled entirely by digital electronics, so it has become far more widespread in recent years. Unfortunately, it can only be used for digital data. Since the local loops produce analog signals, a conversion is needed from analog to digital in the end office, where all the individual local loops come together to be combined onto outgoing trunks. We will now look at how multiple analog voice signals are digitized and combined onto a single outgoing digital trunk. (Remember that computer data sent over a modem are also analog when they get to the end office.)

The analog signals are digitized in the end office by a device called a **codec** (coder-decoder), producing a 7- or 8-bit number (see Fig. 2-17). The codec makes 8000 samples per second (125 μ sec/sample) because the Nyquist theorem says that this is sufficient to capture all the information from the 4-kHz telephone channel bandwidth. At a lower sampling rate, information would be lost; at a higher one, no extra information would be gained. This technique is called **PCM (Pulse Code Modulation)**. PCM forms the heart of the modern telephone system. As a consequence, virtually all time intervals within the telephone system are multiples of 125 μ sec.

When digital transmission began emerging as a feasible technology, CCITT was unable to reach agreement on an international standard for PCM. Consequently, there are now a variety of incompatible schemes in use in different countries around the world. International hookups between incompatible countries require (often expensive) “black boxes” to convert the originating country’s system to that of the receiving country.

One method that is in widespread use in North America and Japan is the T1 carrier, depicted in Fig. 2-26. (Technically speaking, the format is called DS1 and the carrier is called T1, but we will not make that subtle distinction here.) The T1 carrier consists of 24 voice channels multiplexed together. Usually, the analog signals are sampled on a round-robin basis with the resulting analog stream being

fed to the codec rather than having 24 separate codecs and then merging the digital output. Each of the 24 channels, in turn, gets to insert 8 bits into the output stream. Seven bits are data, and one is for control, yielding $7 \times 8000 = 56,000$ bps of data, and $1 \times 8000 = 8000$ bps of signaling information per channel.

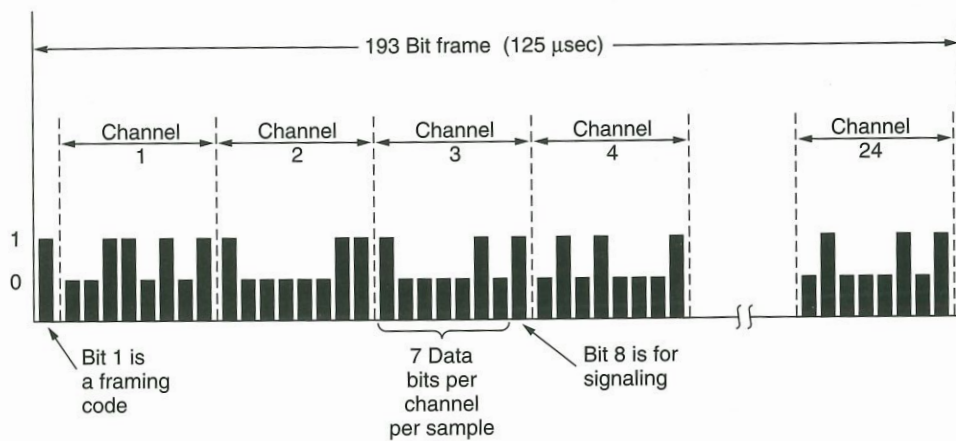


Fig. 2-26. The T1 carrier (1.544 Mbps).

A frame consists of $24 \times 8 = 192$ bits, plus one extra bit for framing, yielding 193 bits every 125 μ sec. This gives a gross data rate of 1.544 Mbps. The 193rd bit is used for frame synchronization. It takes on the pattern 01010101 Normally, the receiver keeps checking this bit to make sure that it has not lost synchronization. If it does get out of sync, the receiver can scan for this pattern to get resynchronized. Analog customers cannot generate the bit pattern at all, because it corresponds to a sine wave at 4000 Hz, which would be filtered out. Digital customers can, of course, generate this pattern, but the odds are against its being present when the frame slips. When a T1 system is being used entirely for data, only 23 of the channels are used for data. The 24th one is used for a special synchronization pattern, to allow faster recovery in the event that the frame slips.

When CCITT finally did reach agreement, they felt that 8000 bps of signaling information was far too much, so its 1.544-Mbps standard is based upon an 8- rather than a 7-bit data item; that is, the analog signal is quantized into 256 rather than 128 discrete levels. Two (incompatible) variations are provided. In **common-channel signaling**, the extra bit (which is attached onto the rear rather than the front of the 193 bit frame) takes on the values 10101010 . . . in the odd frames and contains signaling information for all the channels in the even frames.

In the other variation, **channel associated signaling**, each channel has its own private signaling subchannel. A private subchannel is arranged by allocating one of the eight user bits in every sixth frame for signaling purposes, so five out of six samples are 8 bits wide, and the other one is only 7 bits wide. CCITT also has a

recommendation for a PCM carrier at 2.048 Mbps called **E1**. This carrier has 32 8-bit data samples packed into the basic 125- μ sec frame. Thirty of the channels are used for information and two are used for signaling. Each group of four frames provides 64 signaling bits, half of which are used for channel associated signaling and half of which are used for frame synchronization or are reserved for each country to use as it wishes. Outside North America and Japan, the 2.048-Mbps carrier is in widespread use.

Once the voice signal has been digitized, it is tempting to try to use statistical techniques to reduce the number of bits needed per channel. These techniques are appropriate not only to encoding speech, but to the digitization of any analog signal. All of the compaction methods are based upon the principle that the signal changes relatively slowly compared to the sampling frequency, so that much of the information in the 7- or 8-bit digital level is redundant.

One method, called **differential pulse code modulation**, consists of outputting not the digitized amplitude, but the difference between the current value and the previous one. Since jumps of ± 16 or more on a scale of 128 are unlikely, 5 bits should suffice instead of 7. If the signal does occasionally jump wildly, the encoding logic may require several sampling periods to "catch up." For speech, the error introduced can be ignored.

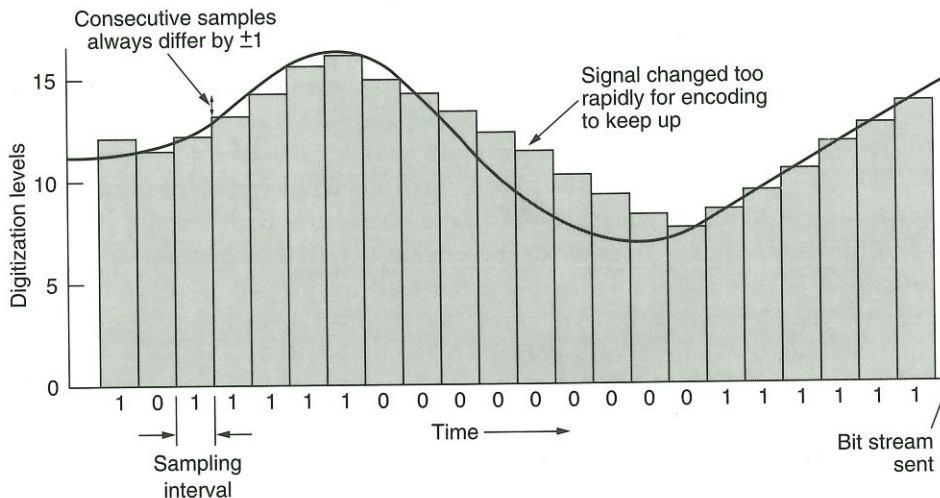


Fig. 2-27. Delta modulation.

A variation of this compaction method requires each sampled value to differ from its predecessor by either +1 or -1. A single bit is transmitted, telling whether the new sample is above or below the previous one. This technique, called **delta modulation**, is illustrated in Fig. 2-27. Like all compaction techniques that assume small level changes between consecutive samples, delta

encoding can get into trouble if the signal changes too fast, as shown in the figure. When this happens, information is lost.

An improvement to differential PCM is to extrapolate the previous few values to predict the next value and then to encode the difference between the actual signal and the predicted one. The transmitter and receiver must use the same prediction algorithm, of course. Such schemes are called **predictive encoding**. They are useful because they reduce the size of the numbers to be encoded, hence the number of bits to be sent.

Although PCM is widely used on interoffice trunks, the computer user gets relatively little benefit from it if all data must be sent to the end office in the form of a modulated analog sine wave at 28.8 kbps. It would be nice if the carrier would attach the local loop directly to the PCM trunk system, so that the computer could output digital data directly onto the local loop at 1.544 or 2.048 Mbps. Unfortunately, the local loops cannot run at these speeds for very far.

Time division multiplexing allows multiple T1 carriers to be multiplexed into higher-order carriers. Figure 2-28 shows how this can be done. At the left we see four T1 channels being multiplexed onto one T2 channel. The multiplexing at T2 and above is done bit for bit, rather than byte for byte with the 24 voice channels that make up a T1 frame. Four T1 streams at 1.544 Mbps should generate 6.176 Mbps, but T2 is actually 6.312 Mbps. The extra bits are used for framing and recovery, in case the carrier slips.

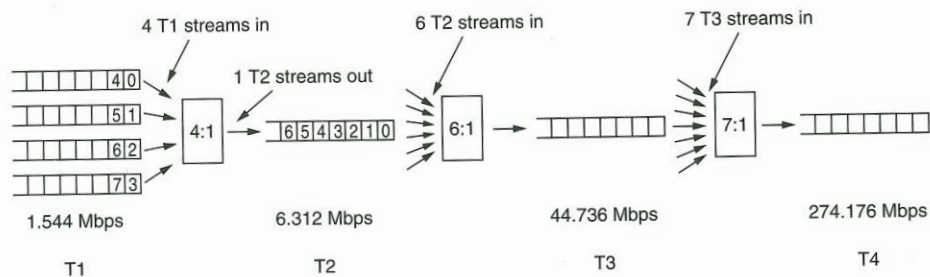


Fig. 2-28. Multiplexing T1 streams onto higher carriers.

At the next level, six T2 streams are combined bitwise to form a T3 stream. Then seven T3 streams are joined to form a T4 stream. At each step a small amount of overhead is added for framing and recovery.

Just as there is little agreement on the basic carrier between the United States and the rest of the world, there is equally little agreement on how it is to be multiplexed into higher bandwidth carriers. The U.S. scheme of stepping up by 4, 6, and 7 did not strike everyone else as the way to go, so the CCITT standard calls for multiplexing four streams onto one stream at each level. Also, the framing and recovery data are different. The CCITT hierarchy for 32, 128, 512, 2048, and 8192 channels runs at speeds of 2.048, 8.848, 34.304, 139.264, and 565.148 Mbps.

4.6.3. FDM

Frequency division multiplexing is the oldest and probably still the most widely used channel allocation scheme. A typical 36-Mbps transponder might be divided statically into 500 or so 64,000-bps PCM channels, each one operating at its own unique frequency to avoid interfering with the others.

Although simple, FDM also has some drawbacks. First, guard bands are needed between the channels to keep the stations separated. This requirement exists because it is not possible to build transmitters that output all their energy in the main band and nothing in the side bands. The amount of bandwidth wasted in the guard bands can be a substantial fraction of the total.

Second, the stations must be carefully power controlled. If a station puts out too much power in the main band, it will also automatically put out too much power in the side bands, spilling over into adjacent channels and causing interference. Finally, FDM is entirely an analog technique and does not lend itself well to implementation in software.

If the number of stations is small and fixed, the frequency channels can be allocated statically in advance. However, if the number of stations, or the load on each one can fluctuate rapidly, some form of dynamic allocation of the frequency bands is needed. One such mechanism is the **SPADE** system used on some early Intelsat satellites. Each SPADE transponder was divided into 794 simplex (64-kbps) PCM voice channels, along with a 128-kbps common signaling channel. The PCM channels were used in pairs to provide full duplex service. The total transponder bandwidth used was 50 Mbps for the uplink portion and another 50 Mbps for the downlink.

The common signaling channel was divided into units of 50 msec. A unit contained 50 slots of 1 msec (128 bits). Each slot was "owned" by one of (not more than) 50 ground stations. When a ground station had data to send, it picked a currently unused channel at random and wrote the number of that channel in its next 128-bit slot. If the selected channel was still unused when the request was seen on the downlink, the channel was considered allocated and all other stations refrained from trying to acquire it. If two or more stations tried to allocate the same channel in the same frame, a collision occurred and they had to try again later. When a station was finished using its channel, it sent a deallocation message in its slot on the common channel.

4.6.4. TDM

Like FDM, TDM is well understood and widely used in practice. It requires time synchronization for the slots, but this can be provided by a reference station, as described for slotted ALOHA above. Similarly to FDM, for a small and unvarying number of stations, the slot assignment can be set up in advance and

never changed, but for a varying number of stations, or a fixed number of stations with time-varying loads, time slots must be assigned dynamically.

Slot assignment can be done in a centralized or a decentralized way. As an example of centralized slot assignment, let us consider the experimental **ACTS (Advanced Communication Technology Satellite)**, which was designed for a few dozen stations (Palmer and White, 1990). ACTS was launched in 1992 and has four independent 110-Mbps TDM channels, two uplink and two downlink. Each channel is organized as a sequence of 1-msec frames, each frame containing 1728 time slots. Each time slot has a 64-bit payload, allowing each one to hold a 64-kbps voice channel.

The beams can be switched from one geographical area to another, but since moving the beam takes several slot times, channels originating or terminating in the same geographic area are normally assigned to contiguous time slots to increase dwell time and minimize time lost to beam motion. Thus time slot management requires a thorough knowledge of station geography to minimize the number of wasted time slots. For this and other reasons, time slot management is done by one of the ground stations, the **MCS (Master Control Station)**.

The basic operation of ACTS is a continuous three-step process, each step taking 1 msec. In step 1, the satellite receives a frame and stores it in a 1728-entry onboard RAM. In step 2, an onboard computer copies each input entry to the corresponding output entry (possibly for the other antenna). In step 3, the output frame is transmitted on the downlink.

Initially, each station is assigned at least one time slot. To acquire additional channels (for new voice calls), a station sends a short request message to the MCS. Similarly, it can release an existing channel with a message to the MCS. These messages make use of a small number of overhead bits and provide a special control channel to the MCS with a capacity of about 13 messages/sec per station. The channels are dedicated; there is no contention for them.

Dynamic TDM slot allocation is also possible. Below we will discuss three schemes. In each of these, TDM frames are divided into time slots, with each slot having a (temporary) owner. Only the owner may use a time slot.

The first scheme assumes that there are more slots than stations, so each station can be assigned a home slot (Binder, 1975). If there are more slots than stations, the extra slots are not assigned to anyone. If the owner of a slot does not want it during the current group, it goes idle. An empty slot is a signal to everyone else that the owner has no traffic. During the next frame, the slot becomes available to anyone who wants it, on a contention (ALOHA) basis.

If the owner wants to retrieve "his" home slot, he transmits a frame, thus forcing a collision (if there was other traffic). After a collision, everyone except the owner must desist from using the slot in the next frame. Thus the owner can always begin transmitting within two frame times in the worst case. At low channel utilization the system does not perform as well as normal slotted ALOHA, since after each collision, the colliders must abstain for one frame to see if the

owner wants the slot back. Fig. 4-51(a) shows a frame with eight slots, seven of which are owned by *G*, *A*, *F*, *E*, *B*, *C*, and *D*, respectively. The eighth slot is not owned by anyone and can be fought over.

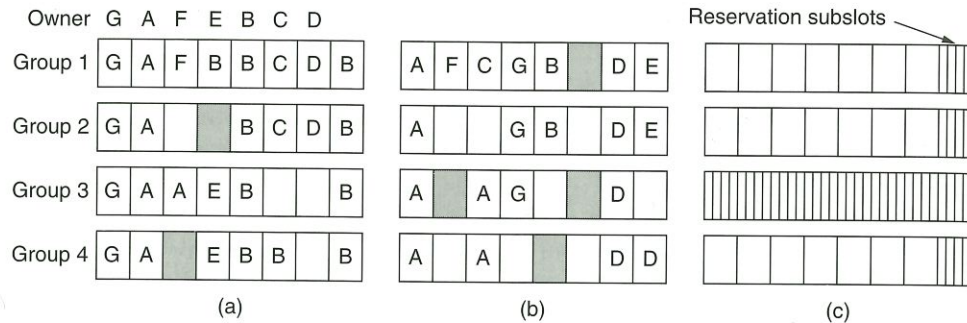


Fig. 4-51. Reservation schemes. (a) Binder. (b) Crowther. (c) Roberts. The shaded boxes indicate collisions. For each of the three schemes, four consecutive groups of slots are shown.

A second scheme is applicable even when the number of stations is unknown and varying (Crowther et al., 1973). In this method, slots do not have permanent owners, as in Binder's. Instead, stations compete for slots using slotted ALOHA. Whenever a transmission is successful, the station making the successful transmission is entitled to that slot in the next frame as well. Thus, as long as a station has data to send, it can continue doing so indefinitely (subject to some "Please-do-not-be-a-pig" rules). In essence the proposal allows a dynamic mix of slotted ALOHA and TDM, with the number of slots devoted to each varying with demand. Fig. 4-51(b) also shows a frame with eight slots. Initially, *E* is using the last slot, but after two frames, it no longer needs it. It lies idle for one frame, and then *D* picks it up and keeps it until it is done.

A third scheme, due to Roberts (1973), requires stations to make advance requests before transmitting. Each frame contains, say, one special slot [the last one in Fig. 4-51(c)] which is divided into V smaller subslots used to make reservations. When a station wants to send data, it broadcasts a short request frame in a randomly-chosen reservation subslot. If the reservation is successful (i.e., no collision), then the next regular slot (or slots) is reserved. At all times everyone must keep track of the queue length (number of slots reserved), so that when any station makes a successful reservation it will know how many data slots to skip before transmitting. Stations need not keep track of *who* is queued up; they merely need to know how long the queue is. When the queue length drops to zero, all slots revert to reservation subslots, to speed up the reservation process.

Although TDM is widely used, both with and without reservation schemes, it, too, has some shortcomings. For one, it requires all stations to synchronize in time, which is not entirely trivial in practice because satellites tend to drift in

orbit, which changes the propagation time to each ground station. It also requires each ground station to be capable of extremely high burst speeds. For example, even though an ACTS station may have only one 64-kbps channel, it must be capable of putting out a 64-bit burst in a 578-nsec time slot. In other words, it must actually operate at 110 Mbps. In contrast, a 64-kbps FDM station really operates at 64 kbps.

4.6.5. CDMA

The final scheme is CDMA. CDMA avoids the time synchronization problem and also the channel allocation problem. It is completely decentralized and fully dynamic.

However, it has three main disadvantages. First, the capacity of a CDMA channel in the presence of noise and uncoordinated stations is typically lower than what TDM can achieve. Second, with 128 chips/bit (a common value), although the bit rate may not be high, the chip rate will be, necessitating a fast (read: expensive) transmitter. Third, few practicing engineers actually understand CDMA, which generally does not increase the chances of their using it, even if it is the best method for a particular application. Nevertheless, CDMA has been used by the military for decades and is now becoming more common in commercial applications as well.

4.7. SUMMARY

Some networks have a single channel that is used for all communication. In these networks, the key design issue is the allocation of this channel among the competing stations wishing to use it. Numerous channel allocation algorithms have been devised. A summary of some of the more important channel allocation methods is given in Fig. 4-52.

The simplest allocation schemes are FDM and TDM. These are efficient when the number of stations is small and the traffic is continuous. Both are widely used under these circumstances, for example, for dividing up the bandwidth in satellite links used as telephone trunks.

When the number of stations is large and variable or the traffic bursty, FDM and TDM are poor choices. The ALOHA protocol, with and without slotting and control, has been proposed as an alternative. ALOHA and its many variants and derivatives have been widely discussed, analyzed, and used in real systems.

When the state of the channel can be sensed, stations can avoid starting a transmission while another station is transmitting. This technique, carrier sensing, has led to a variety of protocols that can be used on LANs and MANs.

A class of protocols that eliminate contention altogether, or at least reduce it considerably, is known. Binary countdown completely eliminates contention.

Method	Description
FDM	Dedicate a frequency band to each station
TDM	Dedicate a time slot to each station
Pure ALOHA	Unsynchronized transmission at any instant
Slotted ALOHA	Random transmission in well-defined time slots
1-persistent CSMA	Standard carrier sense multiple access
Nonpersistent CSMA	Random delay when channel is sensed busy
P-persistent CSMA	CSMA, but with a probability of p of persisting
CSMA/CD	CSMA, but abort on detecting a collision
Bit map	Round robin scheduling using a bit map
Binary countdown	Highest numbered ready station goes next
Tree walk	Reduced contention by selective enabling
Wavelength division	A dynamic FDM scheme for fiber
MACA, MACAW	Wireless LAN protocols
GSM	FDM plus TDM for cellular radio
CDPD	Packet radio within an AMPS channel
CDMA	Everybody speak at once but in a different language
Ethernet	CSMA/CD with binary exponential backoff
Token bus	Logical ring on a physical bus
Token ring	Capture the token to send a frame
DQDB	Distributed queuing on a two-bus MAN
FDDI	Fiber-optic token ring
HIPPI	Crossbar using 50-100 twisted pairs
Fibre channel	Crossbar using fiber optics
SPADE	FDM with dynamic channel allocation
ACTS	TDM with centralized slot allocation
Binder	TDM with ALOHA when slot owner is not interested
Crowther	ALOHA with slot owner getting to keep it
Roberts	Channel time reserved in advance by ALOHA

Fig. 4-52. Channel allocation methods and systems for a common channel.

The tree walk protocol reduces it by dynamically dividing the stations into two disjoint groups, one of which is permitted to transmit and one of which is not. It tries to make the division in such a way that only one station that is ready to send is permitted to do so.

Wireless LANs have their own problems and solutions. The biggest problem is caused by hidden stations, so CSMA does not work. One class of solutions, typified by MACA, attempts to stimulate transmissions around the destination, to make CSMA work better.

For mobile computers and telephones, cellular radio is the up-and-coming technology. GSM, CDPD, and CDMA are widely used.

The IEEE 802 LANs are: CSMA/CD, token bus, and token ring. Each of these has its own unique advantages and disadvantages, and each has found its own user community and will probably continue to serve that community for years to come. Convergence to a single LAN standard is an unlikely event. A new addition to this family is DQDB, being sold as a MAN in many cities.

An organization with multiple LANs often connects them with bridges. When a bridge connects two or more different kinds of LANs, new problems arise, some of them insoluble.

While the 802 LANs are the work horses of the day, the race horses are FDDI, fast Ethernet, HIPPI, and fibre channel. All of these offer bandwidth in the 100 Mbps range and up.

Finally, satellite networks also use multiple access channels (for the uplink). Various channel allocation methods are used here, including ALOHA, FDM, TDM, and CDMA.

PROBLEMS

1. A group of N stations share a 56-kbps pure ALOHA channel. Each station outputs a 1000-bit frame on an average of once every 100 sec, even if the previous one has not yet been sent (e.g., the stations are buffered). What is the maximum value of N ?
2. Consider the delay of pure ALOHA versus slotted ALOHA at low load. Which one is less? Explain your answer.
3. Ten thousand airline reservation stations are competing for the use of a single slotted ALOHA channel. The average station makes 18 requests/hour. A slot is 125 μ sec. What is the approximate total channel load?
4. A large population of ALOHA users manages to generate 50 requests/sec, including both originals and retransmissions. Time is slotted in units of 40 msec.
 - (a) What is the chance of success on the first attempt?
 - (b) What is the probability of exactly k collisions and then a success?
 - (c) What is the expected number of transmission attempts needed?

JPEG may seem complicated, but that is because it *is* complicated. Still, since it often produces a 20:1 compression or better, it is widely used. Decoding a JPEG image requires running the algorithm backward. Unlike some of the other compression algorithms we have seen, JPEG is roughly symmetric: decoding takes as long as encoding.

Interestingly enough, due to the mathematical properties of the DCT, it is possible to perform certain geometric transformations (e.g. image rotation) directly on the transformed matrix, without regenerating the original image. These transformations are discussed in (Shen and Sethi, 1995). Similar properties also apply to MPEG compressed audio (Broadhead and Owen, 1995).

The MPEG Standard

Finally, we come to the heart of the matter: the **MPEG (Motion Picture Experts Group)** standards. These are the main algorithms used to compress videos and have been international standards since 1993. Because movies contain both images and sound, MPEG can compress both audio and video, but since video takes up more bandwidth and also contains more redundancy than audio, we will primarily focus on MPEG video compression below.

The first standard to be finalized was MPEG-1 (International Standard 11172). Its goal was to produce video recorder-quality output (352×240 for NTSC) using a bit rate of 1.2 Mbps. Since we saw earlier that uncompressed video alone can run to 472 Mbps, getting it down to 1.2 Mbps is not entirely trivial, even at this lower resolution. MPEG-1 can be transmitted over twisted pair transmission lines for modest distances. MPEG-1 is also used for storing movies on CD-ROM in CD-I and CD-Video format.

The next standard in the MPEG family was MPEG-2 (International Standard 13818), which was originally designed for compressing broadcast quality video into 4 to 6 Mbps, so it could fit in a NTSC or PAL broadcast channel. Later, MPEG-2 was expanded to support higher resolutions, including HDTV. MPEG-4 is for medium-resolution videoconferencing with low frame rates (10 frames/sec) and at low bandwidths (64 kbps). This will permit videoconferences to be held over a single N-ISDN B channel. Given this numbering, one might think that the next standard will be MPEG-8. Actually, ISO is numbering them linearly, not exponentially. Originally MPEG-3 existed. It was intended for HDTV, but that project was later canceled, and HDTV was added to MPEG-2.

The basic principles of MPEG-1 and MPEG-2 are similar, but the details are different. To a first approximation, MPEG-2 is a superset of MPEG-1, with additional features, frame formats and encoding options. It is likely that in the long run MPEG-1 will dominate for CD-ROM movies and MPEG-2 will dominate for long-haul video transmission. We will discuss MPEG-1 first and then MPEG-2.

MPEG-1 has three parts: audio, video, and system, which integrates the other two, as shown in Fig. 7-85. The audio and video encoders work independently,

which raises the issue of how the two streams get synchronized at the receiver. This problem is solved by having a 90-kHz system clock that outputs the current time value to both encoders. These values are 33 bits, to allow films to run for 24 hours without wrapping around. These timestamps are included in the encoded output and propagated all the way to the receiver, which can use them to synchronize the audio and video streams.

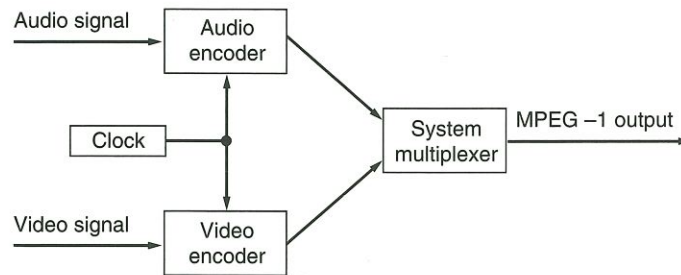


Fig. 7-85. Synchronization of the audio and video streams in MPEG-1.

MPEG audio compression is done by sampling the waveform at 32 kHz, 44.1 kHz, or 48 kHz. It can handle monaural, disjoint stereo (each channel compressed separately), or joint stereo (interchannel redundancy exploited). It is organized as three layers, each one applying additional optimizations to get more compression (and at greater cost). Layer 1 is the basic scheme. This layer is used, for example, in the DCC digital tape system. Layer 2 adds advanced bit allocation to the basic scheme. It is used for CD-ROM audio and movie soundtracks. Layer 3 adds hybrid filters, nonuniform quantization, Huffman coding, and other advanced techniques.

MPEG audio can compress a rock 'n roll CD down to 96 kbps with no perceptible loss in quality, even for rock 'n roll fans with no hearing loss. For a piano concert, at least 128 kbps are needed. These differ because the signal-to-noise ratio for rock 'n roll is much higher than for a piano concert (in an engineering sense, anyway).

Audio compression is carried out by performing a fast Fourier transformation on the audio signal to transform it from the time domain to the frequency domain. The resulting spectrum is then divided up into 32 frequency bands, each of which is processed separately. When two stereo channels are present, the redundancy inherent in having two highly overlapping audio sources is also exploited. The resulting MPEG-1 audio stream is adjustable from 32 kbps to 448 kbps. An introduction to the process is given in (Pan, 1995).

Now let us consider MPEG-1 video compression. Two kinds of redundancies exist in movies: spatial and temporal. MPEG-1 uses both. Spatial redundancy can be utilized by simply coding each frame separately with JPEG. This approach is sometimes used, especially when random access to each frame is needed, as in

editing video productions. In this mode, a compressed bandwidth in the 8- to 10-Mbps range is achievable.

Additional compression can be achieved by taking advantage of the fact that consecutive frames are often almost identical. This effect is smaller than it might first appear since many movie makers cut between scenes every 3 or 4 seconds (time a movie and count the scenes). Nevertheless, even a run of 75 highly similar frames offers the potential of a major reduction over simply encoding each frame separately with JPEG.

For scenes where the camera and background are stationary and one or two actors are moving around slowly, nearly all the pixels will be identical from frame to frame. Here, just subtracting each frame from the previous one and running JPEG on the difference would do fine. However, for scenes where the camera is panning or zooming, this technique fails badly. What is needed is some way to compensate for this motion. This is precisely what MPEG does; it is the main difference between MPEG and JPEG.

MPEG-1 output consists of four kinds of frames:

1. I (Intracoded) frames: Self-contained JPEG-encoded still pictures.
2. P (Predictive) frames: Block-by-block difference with the last frame.
3. B (Bidirectional) frames: Differences with the last and next frame.
4. D (DC-coded) frames: Block averages used for fast forward.

I-frames are just still pictures coded using JPEG, also using full-resolution luminance and half-resolution chrominance along each axis. It is necessary to have I-frames appear in the output stream periodically for three reasons. First, MPEG-1 can be used for a multicast transmission, with viewers tuning it at will. If all frames depended on their predecessors going back to the first frame, anybody who missed the first frame could never decode any subsequent frames. Second, if any frame were received in error, no further decoding would be possible. Third, without I-frames, while doing a fast forward or rewind, the decoder would have to calculate every frame passed over so it would know the full value of the one it stopped on. For these reasons, I-frames are inserted into the output once or twice per second.

P-frames, in contrast, code interframe differences. They are based on the idea of **macroblocks**, which cover 16×16 pixels in luminance space and 8×8 pixels in chrominance space. A macroblock is encoded by searching the previous frame for it or something only slightly different from it.

An example of where P-frames would be useful is given in Fig. 7-86. Here we see three consecutive frames that have the same background, but differ in the position of one person. The macroblocks containing the background scene will match exactly, but the macroblocks containing the person will be offset in position by some unknown amount and will have to be tracked down.

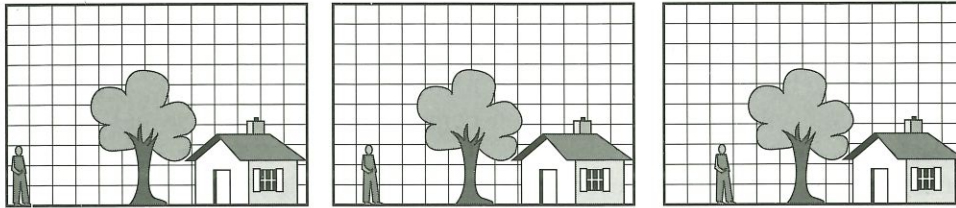


Fig. 7-86. Three consecutive frames.

The MPEG-1 standard does not specify how to search, how far to search, or how good a match has to be to count. This is up to each implementation. For example, an implementation might search for a macroblock at the current position in the previous frame, and all other positions offset $\pm\Delta x$ in the x direction and $\pm\Delta y$ in the y direction. For each position, the number of matches in the luminance matrix could be computed. The position with the highest score would be declared the winner, provided it was above some predefined threshold. Otherwise, the macroblock would be said to be missing. Much more sophisticated algorithms are also possible, of course.

If a macroblock is found, it is encoded by taking the difference with its value in the previous frame (for luminance and both chrominances). These difference matrices are then subject to the discrete cosine transformation, quantization, run-length encoding, and Huffman encoding, just as with JPEG. The value for the macroblock in the output stream is then the motion vector (how far the macroblock moved from its previous position in each direction), followed by the Huffman encoded list of numbers. If the macroblock is not located in the previous frame, the current value is encoded with JPEG, just as in an I-frame.

Clearly, this algorithm is highly asymmetric. An implementation is free to try every plausible position in the previous frame if it wants to, in a desperate attempt to locate every last macroblock. This approach will minimize the encoded MPEG-1 stream at the expense of very slow encoding. This approach might be fine for a one-time encoding of a film library but would be terrible for real-time videoconferencing.

Similarly, each implementation is free to decide what constitutes a “found” macroblock. This freedom allows implementers to compete on the quality and speed of their algorithms, but always produce compliant MPEG-1. No matter what search algorithm is used, the final output is either the JPEG encoding of the current macroblock, or the JPEG encoding of the difference between the current macroblock and one in the previous frame at a specified offset from the current one.

So far, decoding MPEG-1 is straightforward. Decoding I-frames is the same as decoding JPEG images. Decoding P-frames requires the decoder to buffer the previous frame and then build up the new one in a second buffer based on fully

encoded macroblocks and macroblocks containing differences with the previous frame. The new frame is assembled macroblock by macroblock.

B-frames are similar to P-frames, except that they allow the reference macroblock to be in either a previous frame or in a succeeding frame. This additional freedom allows improved motion compensation, and is also useful when objects pass in front of, or behind, other objects. To do B-frame encoding, the encoder needs to hold three decoded frames in memory at once: the past one, the current one, and the future one. Although B-frames give the best compression, not all implementations support them.

D-frames are only used to make it possible to display a low-resolution image when doing a rewind or fast forward. Doing the normal MPEG-1 decoding in real time is difficult enough. Expecting the decoder to do it when slewing through the video at ten times normal speed is asking a bit much. Instead, the D-frames are used to produce low-resolution images. Each D-frame entry is just the average value of one block, with no further encoding, making it easy to display in real time. This facility is important to allow people to scan through a video at high speed in search of a particular scene.

Having finished our treatment of MPEG-1, let us move on to MPEG-2. MPEG-2 encoding is fundamentally similar to MPEG-1 encoding, with I-frames, P-frames, and B-frames. D-frames are not supported, however. Also, the discrete cosine transformation is 10×10 instead of 8×8 , to give 50 percent more coefficients, hence better quality. Since MPEG-2 is targeted at broadcast television as well as CD-ROM applications, it supports both progressive and interlaced images, whereas MPEG-1 supports only progressive images. Other minor details also differ between the two standards.

Instead of supporting only one resolution level, MPEG-2 supports four: low (352×240), main (720×480), high-1440 (1440×1152), and high (1920×1080). Low resolution is for VCRs and backward compatibility with MPEG-1. Main is the normal one for NTSC broadcasting. The other two are for HDTV.

In addition to having four resolution levels, MPEG-2 also supports five **profiles**. Each profile targets some application area. The main profile is for general-purpose use, and probably most chips will be optimized for the main profile and the main resolution level. The simple profile is similar to the main one, except that it excludes the use of B-frames, to make software encoding and decoding easier. The other profiles deal with scalability and HDTV. The profiles differ in terms of the presence or absence of B-frames, chrominance resolution, and scalability of the encoded bit stream to other formats.

The compressed data rate for each combination of resolution and profile is different. These range from about 3 Mbps up to 100 Mbps for HDTV. The normal case is about 3 to 4 Mbps. Some performance data for MPEG are given in (Pancha and El Zarki, 1994).

MPEG-2 has a more general way of multiplexing audio and video than the MPEG-1 model of Fig. 7-85. It defines an unlimited number of elementary

streams, including video and audio, but also including data streams that must be synchronized with the audio and video, for example, subtitles in multiple languages. Each of the streams is first packetized with timestamps. A simple two-stream example is shown in Fig. 7-87.

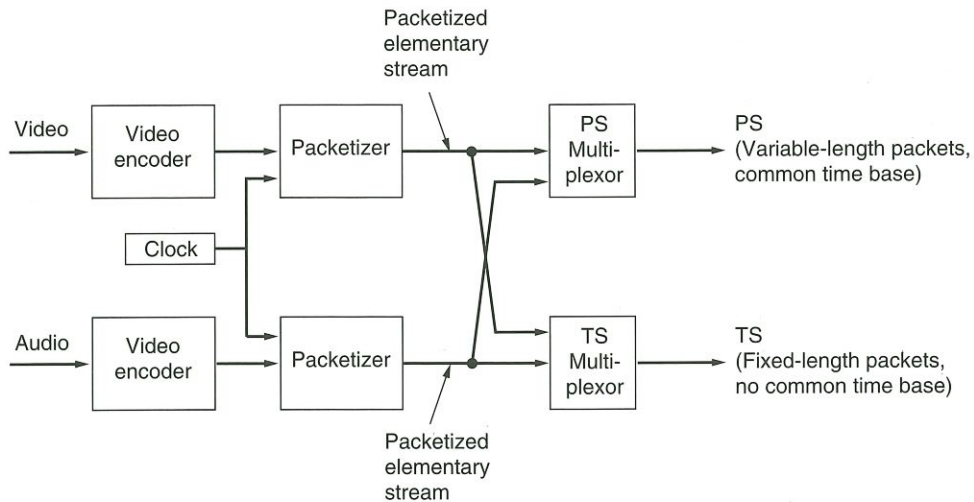


Fig. 7-87. Multiplexing of two streams in MPEG-2.

The output of each packetizer is a **PES (Packetized Elementary Stream)**. Each PES packet has about 30 header fields and flags, including lengths, stream identifiers, encryption control, copyright status, timestamps, and a CRC.

The PES streams for audio, video, and possibly data are then multiplexed together on a single output stream for transmission. Two types of streams are defined. The MPEG-2 **program stream** is similar to the MPEG-1 systems stream of Fig. 7-85. It is used for multiplexing together elementary streams that have a common time base and have to be displayed in a synchronized way. The program stream uses long variable-length packets.

The other MPEG-2 stream is the **transport stream**. It is used for multiplexing together streams (including program streams) that do not use a common time base. The transport stream packets are fixed length (188 bytes), to make it easier to limit the effect of packets damaged or lost during transmission.

It is worth noting that all the encoding schemes we have discussed are based on the model of lossy encoding followed by lossless transmission. Neither JPEG nor MPEG, for example, can recover from lost or damaged packets in a graceful way. A different approach to image transmission is to transform the images in a way that separates the important information from the less important information (as the DCT does, for example). Then add a considerable amount of redundancy (even duplicate packets) to the important information and none to the less

important information. If some packets are lost or garbled, it may still be possible to display reasonable images without retransmission. These ideas are described further in (Danskin et al., 1995). They are especially applicable to multicast transmissions, where feedback from each receiver is impossible anyway.

7.7.4. Video on Demand

Video on demand is sometimes compared to an electronic video rental store. The user (customer) selects any one of a large number of available videos and takes it home to view. Only with video on demand, the selection is made at home using the television set's remote control, and the video starts immediately. No trip to the store is needed. Needless to say, implementing video on demand is a wee bit more complicated than describing it. In this section, we will give an overview of the basic ideas and their implementation. A description of one real implementation can be found in (Nelson and Linton, 1995). A more general treatment of interactive television is in (Hodge, 1995). Other relevant references are (Chang et al., 1994; Hodge et al., 1993; and Little and Venkatesh, 1994).

Is video on demand really like renting a video, or is it more like picking a movie to watch from a 500- or 5000-channel cable system? The answer has important technical implications. In particular, video rental users are used to the idea of being able to stop a video, make a quick trip to the kitchen or bathroom, and then resume from where the video stopped. Television viewers do not expect to put programs on pause.

If video on demand is going to compete successfully with video rental stores, it may be necessary to allow users to stop, start, and rewind videos at will. Giving users this ability virtually forces the video provider to transmit a separate copy to each one.

On the other hand, if video on demand is seen more as advanced television, then it may be sufficient to have the video provider start each popular video, say, every 10 minutes, and run these nonstop. A user wanting to see a popular video may have to wait up to 10 minutes for it to start. Although pause/resume is not possible here, a viewer returning to the living room after a short break can switch to another channel showing the same video but 10 minutes behind. Some material will be repeated, but nothing will be missed. This scheme is called **near video on demand**. It offers the potential for much lower cost, because the same feed from the video server can go to many users at once. The difference between video on demand and near video on demand is similar to the difference between driving your own car and taking the bus.

Watching movies on (near) demand is but one of a vast array of potential new services possible once wideband networking is available. The general model that many people use is illustrated in Fig. 7-88. Here we see a high-bandwidth, (national or international) wide area backbone network at the center of the system. Connected to it are thousands of local distribution networks, such as cable TV or