Computer Networks

Third Edition

Andrew S. Tanenbaum

Vrije Universiteit
Amsterdam, The Netherlands

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*

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

Printed in the United States of America
10 9 8 7 6 5 4 3 2 1

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*

CONTENTS

2.4. THE TELEPHONE SYSTEM

When two computers owned by the same company or organization and located close to each other need to communicate, it is often easiest just to run a cable between them. LANs work this way. However, when the distances are large, or there are many computers, or the cables would have to pass through a public road or other public right of way, the costs of running private cables are usually prohibitive. Furthermore, in just about every country in the world, stringing private transmission lines across (or underneath) public property is also illegal. Consequently, the network designers must rely upon the existing telecommunication facilities.

These facilities, especially the **PSTN**, (**Public Switched Telephone Network**), were usually designed many years ago, with a completely different goal in mind: transmitting the human voice in a more or less recognizable form. Their suitability for use in computer-computer communication is often marginal at best, but the situation is rapidly changing with the introduction of fiber optics and digital technology. In any event, the telephone system is so tightly intertwined with (wide area) computer networks, that it is worth devoting considerable time studying it.

To see the order of magnitude of the problem, let us make a rough but illustrative comparison of the properties of a typical computer-computer connection via a local cable and via a dial-up telephone line. A cable running between two computers can transfer data at memory speeds, typically 10^7 to 10^8 bps. The error rate is usually so low that it is hard to measure, but one error per day would be considered poor at most installations. One error per day at these speeds is equivalent to one error per 10^{12} or 10^{13} bits sent.

In contrast, a dial-up line has a maximum data rate on the order of 10^4 bps and an error rate of roughly 1 per 10^5 bits sent, varying somewhat with the age of the telephone switching equipment involved. The combined bit rate times error rate performance of a local cable is thus 11 orders of magnitude better than a voice-grade telephone line. To make an analogy in the field of transportation, the ratio of the cost of the entire Apollo project, which landed men on the moon to the cost of a bus ride downtown is about 11 orders of magnitude (in 1965 dollars: 40 billion to 0.40).

The trouble, of course, is that computer systems designers are used to working with computer systems, and when suddenly confronted with another system whose performance (from their point of view) is 11 orders of magnitude worse, it is not surprising that much time and effort have been devoted to trying to figure out how to use it efficiently. On the other hand, the telephone companies have made massive strides in the past decade in upgrading equipment and improving service in certain areas. In the following sections we will describe the telephone system and show what it used to be and where it is going. For additional information about the innards of the telephone system see (Bellamy, 1991).

2.4.1. Structure of the Telephone System

When Alexander Graham Bell patented the telephone in 1876 (just a few hours ahead of his rival, Elisha Gray), there was an enormous demand for his new invention. The initial market was for the sale of telephones, which came in pairs. It was up to the customer to string a single wire between them. The electrons returned through the earth. If a telephone owner wanted to talk to n other telephone owners, separate wires had to be strung to all n houses. Within a year, the cities were covered with wires passing over houses and trees in a wild jumble. It became immediately obvious that the model of connecting every telephone to every other telephone, as shown in Fig. 2-14(a) was not going to work.

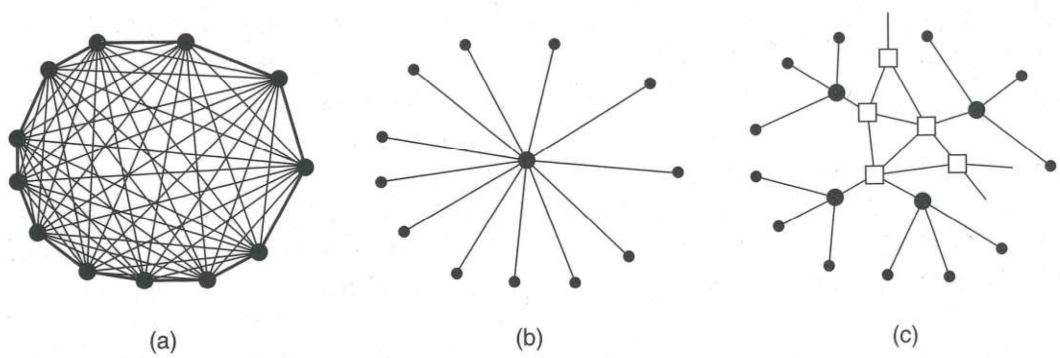

Fig. 2-14. (a) Fully interconnected network. (b) Centralized switch. (c) Two-level hierarchy.

To his credit, Bell saw this and formed the Bell Telephone Company, which opened its first switching office (in New Haven, Connecticut) in 1878. The company ran a wire to each customer's house or office. To make a call, the customer would crank the phone to make a ringing sound in the telephone company office to attract the attention of an operator, who would then manually connect the caller to the callee using a jumper cable. The model of a single switching office is illustrated in Fig. 2-14(b).

Pretty soon, Bell System switching offices were springing up everywhere and people wanted to make long-distance calls between cities, so the Bell system began to connect the switching offices. The original problem soon returned: to connect every switching office to every other switching office by means of a wire between them quickly became unmanageable, so second-level switching offices were invented, After a while, multiple second-level offices were needed, as shown in Fig. 2-14(c). Eventually, the hierarchy grew to five levels.

By 1890, the three major parts of the telephone system were in place: the switching offices, the wires between the customers and the switching offices (by now balanced, insulated, twisted pairs instead of open wires with an earth return), and the long-distance connections between the switching offices. While there have been improvements in all three areas since then, the basic Bell System model has remained essentially intact for over 100 years. For a short technical history of the telephone system, see (Hawley, 1991).

At present, the telephone system is organized as a highly redundant, multilevel hierarchy. The following description is highly simplified but gives the essential flavor nevertheless. Each telephone has two copper wires coming out of it that go directly to the telephone company's nearest **end office** (also called a **local central office**). The distance is typically 1 to 10 km, being smaller in cities than in rural areas.

In the United States alone there are about 19,000 end offices. The concatenation of the area code and the first three digits of the telephone number uniquely specify an end office, which is why the rate structure uses this information. The two-wire connections between each subscriber's telephone and the end office are known in the trade as the **local loop**. If the world's local loops were stretched out end to end, they would extend to the moon and back 1000 times.

At one time, 80 percent of AT&T's capital value was the copper in the local loops. AT&T was then, in effect, the world's largest copper mine. Fortunately, this fact was not widely known in the investment community. Had it been known, some corporate raider might have bought AT&T, terminated all telephone service in the United States, ripped out all the wire, and sold the wire to a copper refiner to get a quick payback.

If a subscriber attached to a given end office calls another subscriber attached to the same end office, the switching mechanism within the office sets up a direct electrical connection between the two local loops. This connection remains intact for the duration of the call.

If the called telephone is attached to another end office, a different procedure has to be used. Each end office has a number of outgoing lines to one or more nearby switching centers, called **toll offices** (or if they are within the same local area, **tandem offices**). These lines are called **toll connecting trunks**. If both the caller's and callee's end offices happen to have a toll connecting trunk to the same toll office (a likely occurrence if they are relatively close by), the connection may be established within the toll office. A telephone network consisting only of telephones (the small dots), end offices (the large dots) and toll offices (the squares) is shown in Fig. 2-14(c).

If the caller and callee do not have a toll office in common, the path will have to be established somewhere higher up in the hierarchy. There are primary, sectional, and regional offices that form a network by which the toll offices are connected. The toll, primary, sectional, and regional exchanges communicate with each other via high bandwidth **intertoll trunks** (also called **interoffice trunks**). The number of different kinds of switching centers and their topology (e.g., may two sectional offices have a direct connection or must they go through a regional office?) varies from country to country depending on its telephone density. Figure 2-15 shows how a medium-distance connection might be routed.

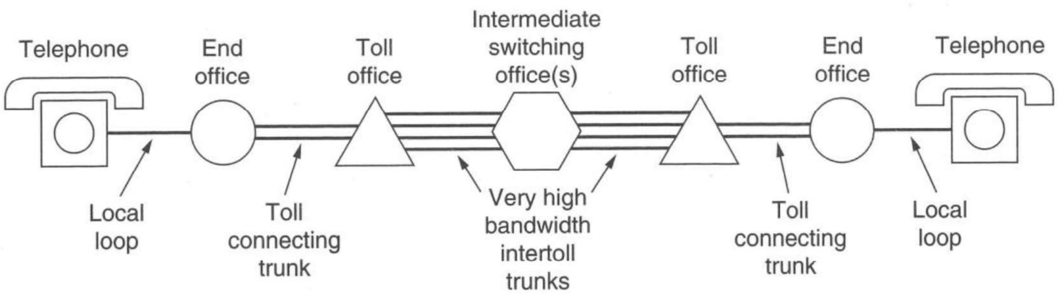

Fig. 2-15. Typical circuit route for a medium-distance call.

A variety of transmission media are used for telecommunication. Local loops consist of twisted pairs nowadays, although in the early days of telephony, uninsulated wires spaced 25 cm apart on telephone poles were common. Between switching offices, coaxial cables, microwaves, and especially fiber optics are widely used.

In the past, signaling throughout the telephone system was analog, with the actual voice signal being transmitted as an electrical voltage from source to destination. With the advent of digital electronics and computers, digital signaling has become possible. In this system, only two voltages are allowed, for example −5 volts and +5 volts.

This scheme has a number of advantages over analog signaling. First is that although the attenuation and distortion are more severe when sending two-level signals than when using modems, it is easy to calculate how far a signal can propagate and still be recognizable. A digital regenerator can be inserted into the line there, to restore the signal to its original value, since there are only two possibilities. A digital signal can pass through an arbitrary number of regenerators with no loss in signal and thus travel long distances with no information loss. In contrast, analog signals always suffer some information loss when amplified, and this loss is cumulative. The net result is that digital transmission can be made to have a low error rate.

A second advantage of digital transmission is that voice, data, music, and images (e.g., television, fax, and video) can be interspersed to make more efficient use of the circuits and equipment. Another advantage is that much higher data rates are possible using existing lines.

A third advantage is that digital transmission is much cheaper than analog transmission, since it is not necessary to accurately reproduce an analog waveform after it has passed through potentially hundreds of amplifiers on a transcontinental call. Being able to correctly distinguish a 0 from a 1 is enough.

Finally, maintenance of a digital system is easier than maintenance of an analog one. A transmitted bit is either received correctly or not, making it simpler to track down problems.

Consequently, all the long-distance trunks within the telephone system are rapidly being converted to digital. The old system used analog transmission over copper wires; the new one uses digital transmission over optical fibers.

In summary, the telephone system consists of three major components:

1. Local loops (twisted pairs, analog signaling).
2. Trunks (fiber optics or microwave, mostly digital).
3. Switching offices.

After a short digression on the politics of telephones, we will come back to each of these three components in some detail. For the local loop, we will be concerned with how to send digital data over it (quick answer: use a modem). For the long-haul trunks, the main issue is how to collect multiple calls together and send them together. This subject is called multiplexing, and we will study three different ways to do it. Finally, there are two fundamentally different ways of doing switching, so we will look at both of these.

2.4.2. The Politics of Telephones

For decades prior to 1984, the Bell System provided both local and long distance service throughout most of the United States. In the 1970s, the U.S. government came to believe that this was an illegal monopoly and sued to break it up. The government won, and on Jan. 1, 1984, AT&T was broken up into AT&T Long Lines, 23 **BOC**s (**Bell Operating Companies**), and a few other pieces. The 23 BOCs were grouped together into seven regional BOCs (RBOCs) to make them economically viable. The entire nature of telecommunication in the United States was changed overnight by court order (*not* by an act of Congress).

The exact details of the divestiture were described in the so-called **MFJ** (**Modified Final Judgment**), an oxymoron if ever there was one (if the judgment could be modified, it clearly was not final). This event led to increased competition, better service, and lower prices to consumers and businesses. Many other countries are now considering introducing competition along similar lines.

To make it clear who could do what, the United States was divided up into about 160 **LATA**s (**Local Access and Transport Areas**). Very roughly, a LATA is about as big as the area covered by one area code. Within a LATA, there is normally one **LEC** (**Local Exchange Carrier**) that has a monopoly on traditional telephone service within the LATA. The most important LECs are the BOCs, although some LATAs contain one or more of the 1500 independent telephone companies operating as LECs. In geographically large LATAs (mostly in the West), the LEC may handle long distance calls within its own LATA but may not handle calls going to a different LATA.

All inter-LATA traffic is handled by a different kind of company, an **IXC** (**IntereXchange Carrier**). Originally, AT&T Long Lines was the only serious IXC, but now MCI and Sprint are well-established competitors in the IXC business. One of the concerns at the breakup was to ensure that all the IXCs would be treated equally in terms of line quality, tariffs, and the number of digits their customers would have to dial to use them. The way this is handled is illustrated in Fig. 2-16. Here we see three example LATAs, each with several end offices. LATAs 2 and 3 also have a small hierarchy with tandem offices (intra-LATA toll offices).

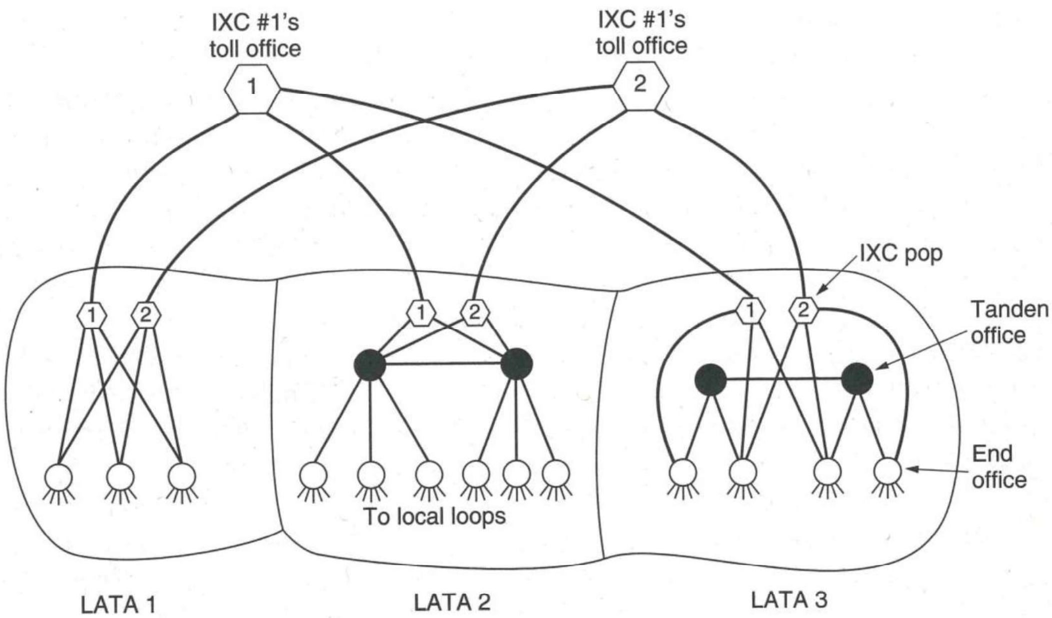

Fig. 2-16. The relationship of LATAs, LECs, and IXCs. All the circles are LEC switching offices. Each hexagon belongs to the IXC whose number is in it.

Any IXC that wishes to handle calls originating in a LATA can build a switching office called a **POP** (**Point of Presence**) there. The LEC is required to connect each IXC to every end office, either directly, as in LATAs 1 and 3, or indirectly, as in LATA 2. Furthermore, the terms of the connection, both technical and financial, must be identical for all IXCs. In this way, a subscriber in, say, LATA 1, can choose which IXC to use for calling subscribers in LATA 3.

As part of the MFJ, the IXCs were forbidden to offer local telephone service and the LECs were forbidden to offer inter-LATA telephone service, although both were free to enter other businesses, such as operating fried chicken restaurants. In 1984, that was a fairly unambiguous statement. Unfortunately, technology has a way of making the law obsolete. Neither cable television nor cellular phones were covered by the agreement. As cable television went from one way to two way, and cellular phones exploded in popularity, both LECs and IXCs began buying up or merging with cable and cellular operators.

By 1995, Congress saw that trying to maintain a distinction between the various kinds of companies was no longer tenable and drafted a bill to allow cable TV companies, local telephone companies, long distance carriers, and cellular operators to enter one another's businesses. The idea was that any company could then offer its customers a single integrated package containing cable TV, telephone, and information services, and that different companies would compete on service and price. The bill was enacted into law in February 1996. As a result, the U.S. telecommunications landscape is currently undergoing a radical restructuring.

2.4.3. The Local Loop

For the past 100 years, analog transmission has dominated all communication. In particular, the telephone system was originally based entirely on analog signaling. While the long-distance trunks are now largely digital in the more advanced countries, the local loops are still analog and are likely to remain so for at least a decade or two, due to the enormous cost of converting them. Consequently, when a computer wishes to send digital data over a dial-up line, the data must first be converted to analog form by a modem for transmission over the local loop, then converted to digital form for transmission over the long-haul trunks, then back to analog over the local loop at the receiving end, and finally back to digital by another modem for storage in the destination computer. This arrangement is shown in Fig. 2-17.

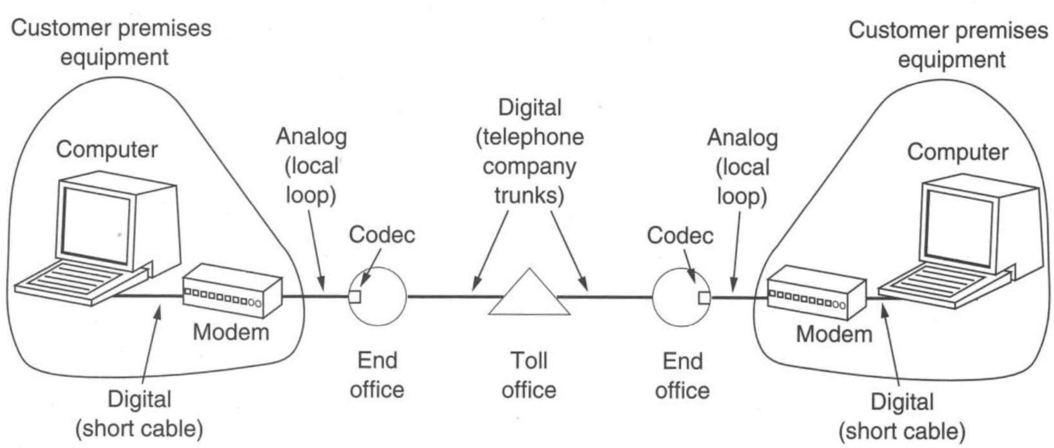

Fig. 2-17. The use of both analog and digital transmission for a computer to computer call. Conversion is done by the modems and codecs.

While this situation is not exactly ideal, such is life for the time being, and students of networking should have some understanding of both analog and digital transmission, as well as how the conversions back and forth work. For leased lines it is possible to go digital from start to finish, but these are expensive and are only useful for building intracompany private networks.

In the following sections we will look briefly at what is wrong with analog transmission and examine how modems make it possible to transmit digital data over analog circuits. We will also look at two common modem interfaces, RS-232-C and RS-449.

Transmission Impairments

Analog signaling consists of varying a voltage with time to represent an information stream. If transmission media were perfect, the receiver would receive exactly the same signal that the transmitter sent. Unfortunately, media are not perfect, so the received signal is not the same as the transmitted signal. For digital data, this difference can lead to errors.

Transmission lines suffer from three major problems: attenuation, delay distortion, and noise. **Attenuation** is the loss of energy as the signal propagates outward. On guided media (e.g., wires and optical fibers), the signal falls off logarithmically with the distance. The loss is expressed in decibels per kilometer. The amount of energy lost depends on the frequency. To see the effect of this frequency dependence, imagine a signal not as a simple waveform, but as a series of Fourier components. Each component is attenuated by a different amount, which results in a different Fourier spectrum at the receiver, and hence a different signal.

If the attenuation is too much, the receiver may not be able to detect the signal at all, or the signal may fall below the noise level. In many cases, the attenuation properties of a medium are known, so amplifiers can be put in to try to compensate for the frequency-dependent attenuation. The approach helps but can never restore the signal exactly back to its original shape.

The second transmission impairment is **delay distortion**. It is caused by the fact that different Fourier components travel at different speeds. For digital data, fast components from one bit may catch up and overtake slow components from the bit ahead, mixing the two bits and increasing the probability of incorrect reception.

The third impairment is **noise**, which is unwanted energy from sources other than the transmitter. Thermal noise is caused by the random motion of the electrons in a wire and is unavoidable. Cross talk is caused by inductive coupling between two wires that are close to each other. Sometimes when talking on the telephone, you can hear another conversation in the background. That is cross talk. Finally, there is impulse noise, caused by spikes on the power line or other causes. For digital data, impulse noise can wipe out one or more bits.

Modems

Due to the problems just discussed, especially the fact that both attenuation and propagation speed are frequency dependent, it is undesirable to have a wide range of frequencies in the signal. Unfortunately, square waves, as in digital data, have a wide spectrum and thus are subject to strong attenuation and delay distortion. These effects make baseband (DC) signaling unsuitable except at slow speeds and over short distances.

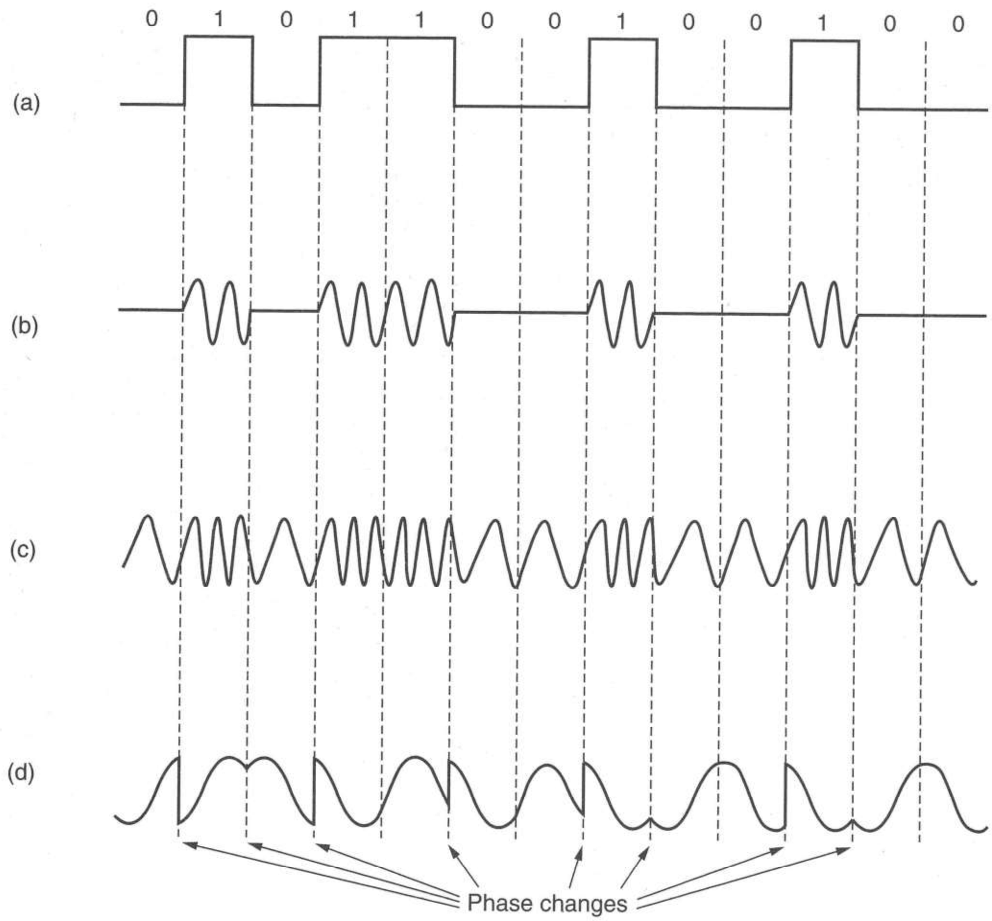

Fig. 2-18. (a) A binary signal. (b) Amplitude modulation. (c) Frequency modulation. (d) Phase modulation.

To get around the problems associated with DC signaling, especially on telephone lines, AC signaling is used. A continuous tone in the 1000- to 2000-Hz range, called a **sine wave carrier** is introduced. Its amplitude, frequency, or phase can be modulated to transmit information. In **amplitude modulation**, two different voltage levels are used to represent 0 and 1, respectively. In **frequency modulation**, also known as **frequency shift keying**, two (or more) different tones are used. In the simplest form of **phase modulation**, the carrier wave is systematically shifted 45, 135, 225, or 315 degrees at uniformly spaced intervals. Each phase shift transmits 2 bits of information. Figure 2-18 illustrates the three forms of modulation. A device that accepts a serial stream of bits as input and produces a modulated carrier as output (or vice versa) is called a **modem** (for modulator-demodulator). The modem is inserted between the (digital) computer and the (analog) telephone system.

To go to higher and higher speeds, it is not possible to just keep increasing the sampling rate. The Nyquist theorem says that even with a perfect 3000-Hz line (which a dial-up telephone is decidedly not), there is no point in sampling faster than 6000 Hz. Thus all research on faster modems is focused on getting more bits per sample (i.e., per baud).

Most advanced modems use a combination of modulation techniques to transmit multiple bits per baud. In Fig. 2-19(a), we see dots at 0, 90, 180, and 270 degrees, with two amplitude levels per phase shift. Amplitude is indicated by the distance from the origin. In Fig. 2-19(b) we see a different modulation scheme, in which 16 different combinations of amplitude and phase shift are used. Thus Fig. 2-19(a) has eight valid combinations and can be used to transmit 3 bits per baud. In contrast, Fig. 2-19(b) has 16 valid combinations and can thus be used to transmit 4 bits per baud. The scheme of Fig. 2-19(b) when used to transmit 9600 bps over a 2400-baud line is called **QAM** (**Quadrature Amplitude Modulation**).

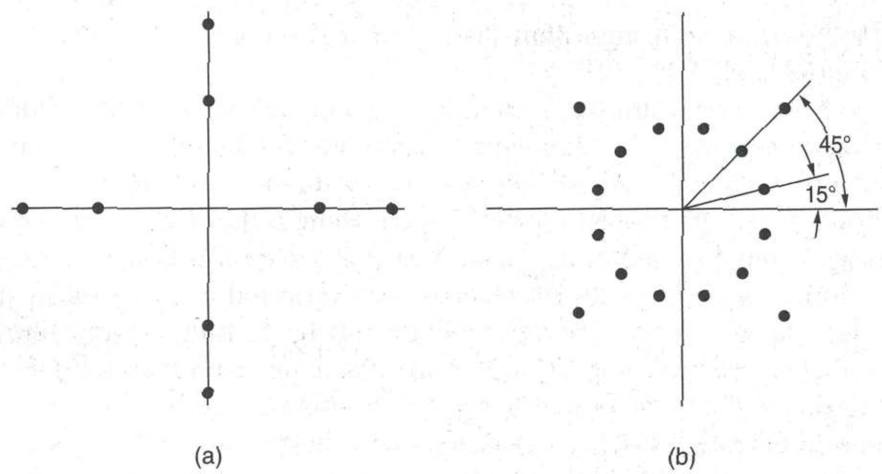

Fig. 2-19. (a) 3 bits/baud modulation. (b) 4 bits/baud modulation.

Diagrams such as those of Fig. 2-19, which show the legal combinations of amplitude and phase, are called **constellation patterns**. Each high-speed modem standard has its own constellation pattern and can talk only to other modems that use the same one (although most modems can emulate all the slower ones). The ITU V.32 9600 bps modem standard uses the constellation pattern of Fig. 2-19(b), for example.

The next step above 9600 bps is 14,400 bps. It is called **V.32 bis**. This speed is achieved by transmitting 6 bits per sample at 2400 baud. Its constellation pattern has 64 points. Fax modems use this speed to transmit pages that have been scanned in as bit maps. After V.32 bis comes **V.34**, which runs at 28,800 bps.

With so many points in the constellation pattern, even a small amout of noise in the detected amplitude or phase can result in an error, and potentially 6 bad bits. To reduce the chance of getting an error, many modems add a parity bit, giving 128 points in the constellation pattern. The coding of the points is carefully done to maximize the chance of detecting errors. The coding that does this is called **trellis coding**.

A completely different approach to high-speed transmission is to divide the available 3000-Hz spectrum into 512 tiny bands and transmit at, say, 20 bps in each one. This scheme requires a substantial processor inside the modem, but has the advantage of being able to disable frequency bands that are too noisy. Modems that use this approach normally have V.32 or V.34 capability as well, so they can talk to standard modems.

Many modems now have compression and error correction built into the modems. The big advantage of this approach is that these features improve the effective data rate without requiring any changes to existing software. One popular compression scheme is **MNP 5**, which uses run-length encoding to squeeze out runs of identical bytes. Fax modems also use run-length encoding, since runs of 0s (blank paper) are very common. Another scheme is **V.42 bis**, which uses a Ziv-Lempel compression algorithm also used in Compress and other programs (Ziv and Lempel, 1977).

Even when modems are used, another problem can occur on telephone lines: echoes. On a long line, when the signal gets to the final destination, some of the energy may be reflected back, analogous to acoustic echos in the mountains. As an illustration of electromagnetic echoes, try shining a flashlight from a darkened room through a closed window at night. You will see a reflection of the flashlight in the window (i.e., some of the energy has been reflected at the air-glass junction and sent back toward you). The same thing happens on transmission lines, especially at the point where the local loop terminates in the end office.

The effect of the echo is that a person speaking on the telephone hears his own words after a short delay. Psychological studies have shown that this is annoying to many people, often making them stutter or become confused. To eliminate the problem of echoes, echo suppressors are installed on lines longer than 2000 km. (On short lines the echoes come back so fast that people are not bothered by them.) An **echo suppressor** is a device that detects human speech coming from one end of the connection and suppresses all signals going the other way. It is basically an amplifier than can be switched on and off by a control signal produced by a speech detection circuit.

When the first person stops talking and the second begins, the echo suppressor switches directions. A good echo suppressor can reverse in 2 to 5 msec. While it is functioning, however, information can only travel in one direction; echoes cannot get back to the sender. Figure 2-20(a) shows the state of the echo suppressors while *A* is talking to *B*. Figure 2-20(b) shows the state after *B* has started talking.

The echo suppressors have several properties that are undesirable for data

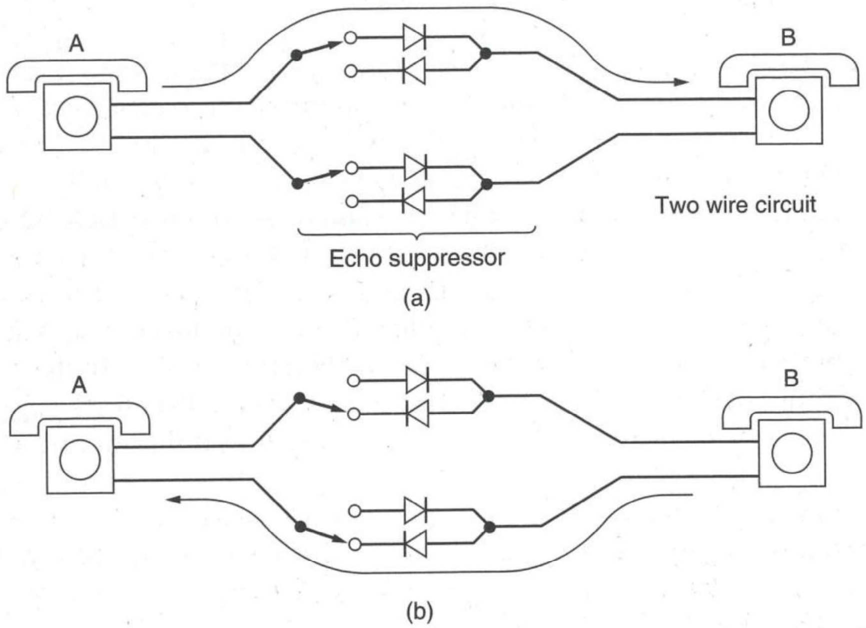

Fig. 2-20. (a) *A* talking to *B*. (b) *B* talking to *A*.

communication. First, it they were not present, it would be possible to transmit in both directions at the same time by using a different frequency band for each direction. This approach is called **full-duplex** transmission. With echo suppressors, full-duplex transmission is impossible. The alternative is **half-duplex** transmission, in which communication can go either way, but only one at a time. A single railroad track is half-duplex. Even if half-duplex transmission is adequate, it is a nuisance because the time required to switch directions can be substantial. Furthermore, the echo suppressors are designed to reverse upon detecting human speech, not digital data.

To alleviate these problems, an escape hatch has been provided on telephone circuits with echo suppressors. When the echo suppressors hear a pure tone at 2100 Hz, they shut down and remain shut down as long as a carrier is present. This arrangement is one of the many examples of **in-band signaling**, so called because the control signals that activate and deactivate internal control functions lie within the band accessible to the user. In general the trend is away from in-band signaling, to prevent users from interfering with the operation of the system itself. In the United States, most of the in-band signaling is gone, but in other countries it still exists.

An alternative to echo suppressors are **echo cancelers.** These are circuits that simulate the echo, estimate how much it is, and subtract it from the signal delivered, without the need for mechanical relays. When echo cancelers are used, full-duplex operation is possible. For this reason, echo cancelers are rapidly replacing echo suppressors in the United States and other large countries.

RS-232-C and RS-449

The interface between the computer or terminal and the modem is an example of a physical layer protocol. It must specify in detail the mechanical, electrical, functional, and procedural interface. We will now look closely at two well-known physical layer standards: RS-232-C and its successor, RS-449.

Let us start with **RS-232-C**, the third revision of the original RS-232 standard. The standard was drawn up by the Electronic Industries Association, a trade organization of electronics manufacturers, and is properly referred to as EIA RS-232-C. The international version is given in CCITT recommendation **V.24**, which is similar but differs slightly on some of the rarely used circuits. In the standards, the terminal or computer is officially called a **DTE** (**Data Terminal Equipment**) and the modem is officially called a **DCE** (**Data Circuit-Terminating Equipment**).

The mechanical specification is for a 25-pin connector 47.04 ± .13 mm wide (screw center to screw center), with all the other dimensions equally well specified. The top row has pins numbered 1 to 13 (left to right); the bottom row has pins numbered 14 to 25 (also left to right).

The electrical specification for RS-232-C is that a voltage more negative than −3 volts is a binary 1 and a voltage more positive than +4 volts is a binary 0. Data rates up to 20 kbps are permitted, as are cables up to 15 meters.

The functional specification tells which circuits are connected to each of the 25 pins, and what they mean. Figure 2-21 shows 9 pins that are nearly always implemented. The remaining ones are frequently omitted. When the terminal or computer is powered up, it asserts (i.e., sets to a logical 1) Data Terminal Ready (pin 20). When the modem is powered up, it asserts Data Set Ready (pin 6). When the modem detects a carrier on the telephone line, it asserts Carrier Detect (pin 8). Request to Send (pin 4) indicates that the terminal wants to send data. Clear to Send (pin 5) means that the modem is prepared to accept data. Data are transmitted on the Transmit circuit (pin 2) and received on the Receive circuit (pin 3).

Other circuits are provided for selecting the data rate, testing the modem, clocking the data, detecting ringing signals, and sending data in the reverse direction on a secondary channel. They are hardly ever used in practice.

The procedural specification is the protocol, that is, the legal sequence of events. The protocol is based on action-reaction pairs. When the terminal asserts Request to Send, for example, the modem replies with Clear to Send, if it is able to accept data. Similar action-reaction pairs exist for other circuits as well.

It commonly occurs that two computers must be connected using RS-232-C. Since neither one is a modem, there is an interface problem. This problem is solved by connecting them with a device called a **null modem**, which connects the transmit line of one machine to the receive line of the other. It also crosses some of the other lines in a similar way. A null modem looks like a short cable.

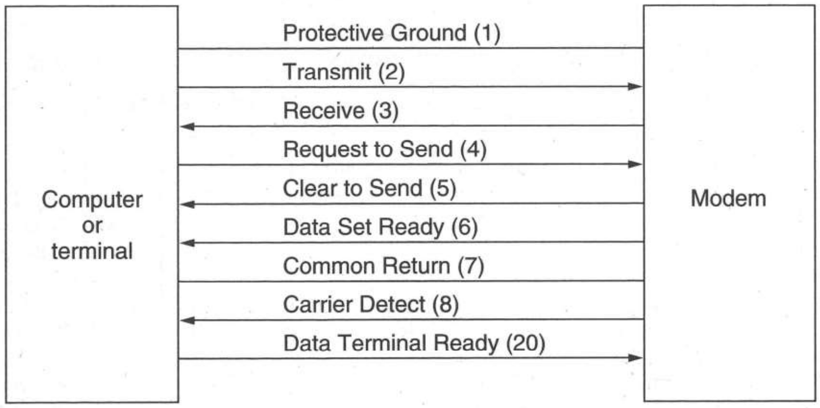

Fig. 2-21. Some of the principal RS-232-C circuits. The pin numbers are given in parentheses.

RS-232-C has been around for years. Gradually, the limitation of the data rate to not more than 20 kbps and the 15-meter maximum cable length have become increasingly annoying. EIA had a long debate about whether to try to have a new standard that was compatible with the old one (but technically not very advanced) or a new and incompatible one that would meet all needs for years to come. They eventually compromised by choosing both.

The new standard, called **RS-449**, is actually three standards in one. The mechanical, functional, and procedural interfaces are given in RS-449, but the electrical interface is given by two different standards. The first of these, **RS-423-A**, is similar to RS-232-C in that all its circuits share a common ground. This technique is called **unbalanced transmission**. The second electrical standard, **RS-422-A**, in contrast, uses **balanced transmission**, in which each of the main circuits requires two wires, with no common ground. As a result, RS-422-A can be used at speeds up to 2 Mbps over 60-meter cables.

The circuits used in RS-449 are shown in Fig. 2-22. Several new circuits not present in RS-232-C have been added. In particular, circuits for testing the modem both locally and remotely were included. Due to the inclusion of a number of two-wire circuits (when RS-422-A is used), more pins are needed in the new standard, so the familiar 25-pin connector was dropped. In its place is a 37-pin connector and a 9-pin connector. The 9-pin connector is required only if the second (reverse) channel is being used.

Fiber in the Local Loop

For advanced future services, such as video on demand, the 3-kHz channel currently used will not do. Discussions about what to do about this tend to focus on two solutions. The straightforward one—running a fiber from the end office

	RS-232-C			CCITT V .24			RS-449		
	Code	Pin	Circuit	Code	Pin	Circuit	Code	Pin	Circuit
	AA	1	Protective ground	101	1	Protective ground	_	1	
	AB	7	Signal ground	102	7	Signal ground	SG	19	Signal ground
							SC	37	Send common
							RC	20	Receive common
	BA	2	Transmitted data	103	2	Transmitted data	SD	4, 22	Send data
	BB	3	Received data	104	3	Received data	RD	6, 24	Receive data
	CA	4	Request to send	105	4	Request to send	RS	7, 25	Request to send
	CB	5	Clear to send	106	5	Ready for sending	CS	9, 27	Clear to send
	CC	6	Data set ready	107	6	Data set ready	DM	11, 29	Data mode
	CD	20	Data terminal ready	108	20	Data terminal ready	TR	12, 30	Terminal ready
	CE	22	Ring indicator	125	22	Calling indicator	IC	15	Incoming call
	CF	8	Line detector	109	8	Line detector	RR	13, 31	Receiver ready
	CG	21	Signal quality	110	21	Signal quality	SQ	33	Signal quality
	CH	23	DTE rate	111	23	DTE rate	SR	16	Signaling rate
	CI	18	DCE rate	112	18	DCE rate	SI	2	Signaling indicators
							IS	28	Terminal in service
				136		New signal	NS	34	New signal
				126	11	Select frequency	SF	16	Select frequency
	DA	24	DTE timing	113	24	DTE timing	TT	17, 25	Terminal timing
	DB	15	DCE timing	114	15	DCE timing	ST	5, 23	Send timing
	DD	17	Receiver timing	115	17	Receiver timing	RT	8, 26	Receive timing
Secondary Channel	SBA	14	Transmitted data	118	14	Transmitted data	SSD	3	Send data
Secondary Channel	SBB	16	Received data	119	16	Received data	SRD	4	Receive data
Secondary Channel	SCA	19	Request to send	120	19	Line signal	SRS	7	Request to send
Secondary Channel	SCB	13	Clear to send	121	13	Channel ready	SCS	8	Clear to send
Secondary Channel	SCF	12	Line detector	122	12	Line detector	SRR	2	Receiver ready
							LL	10	Local loopback
							RL	14	Remote loopback
							TM	18	Test mode
							SS	32	Select standby
							SB	36	Standby indicator

Fig. 2-22. Comparison of RS-232-C, V.24, and RS-449.

into everyone's house is called **FTTH** (**Fiber To The Home**). This solution fits in well with the current system but will not be economically feasible for decades. It is simply too expensive.

An alternative solution that is much cheaper is **FTTC** (**Fiber To The Curb**). In this model, the telephone company runs an optical fiber from each end office into each neighborhood (the curb) that it serves (Paff, 1995). The fiber is terminated in a junction box that all the local loops enter. Since the local loops are now much shorter (perhaps 100 meters instead of 3 km), they can be run at higher speeds, probably around 1 Mbps, which is just enough for compressed video. This design is shown in Fig. 2-23(a).

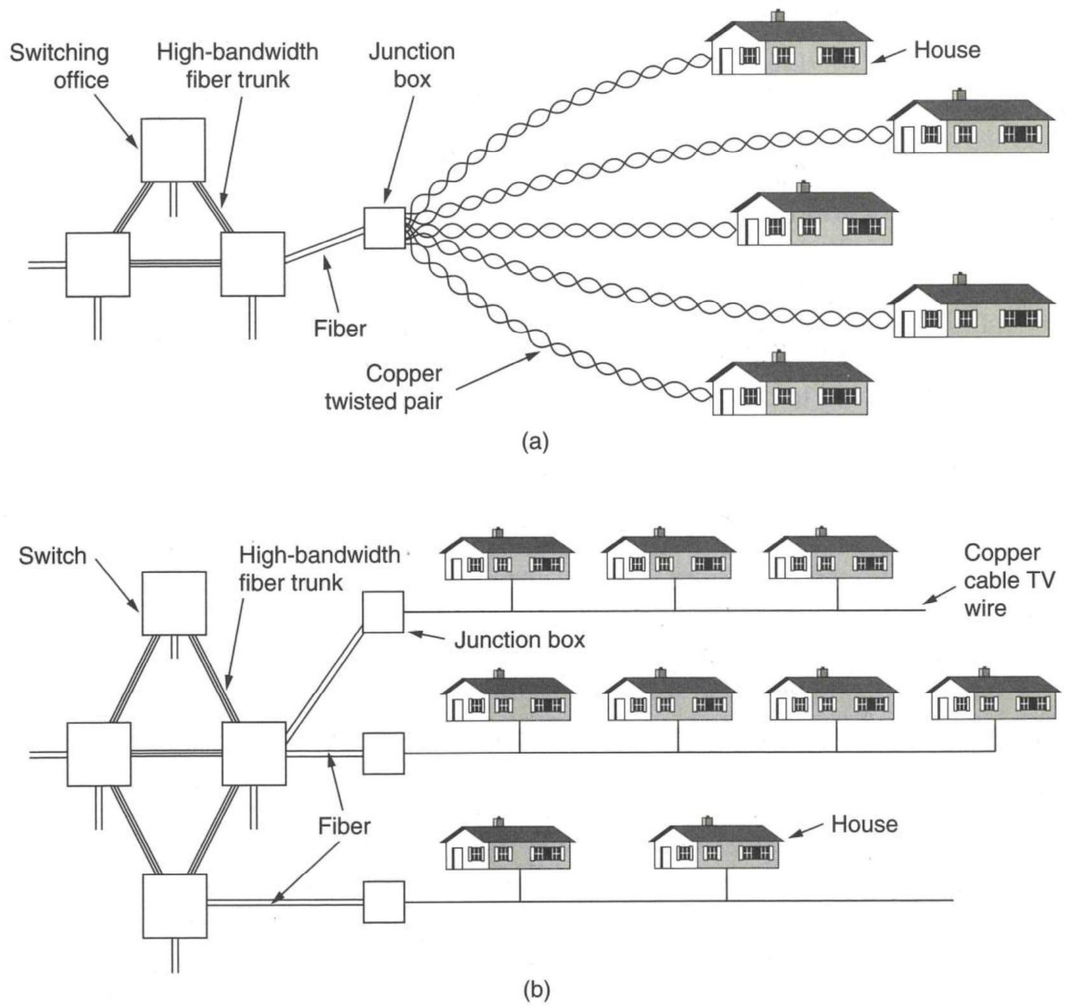

Fig. 2-23. Fiber to the curb. (a) Using the telephone network. (b) Using the cable TV network.

In this manner, multiple videos (or other information channels) can pour down the fiber at high speed and be split over the twisted pairs at the end. By sharing a 1-Gbps fiber over 100 to 1000 customers, the cost per customer can be reduced, and considerably higher bandwidth can be provided than now. Going appreciably above 1 Mbps for long distances with the existing twisted pairs is impossible. Thus in the long term, all the twisted pairs will have to be replaced by fiber. Whether the intermediate solution of FTTC should be used for the time being or

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.

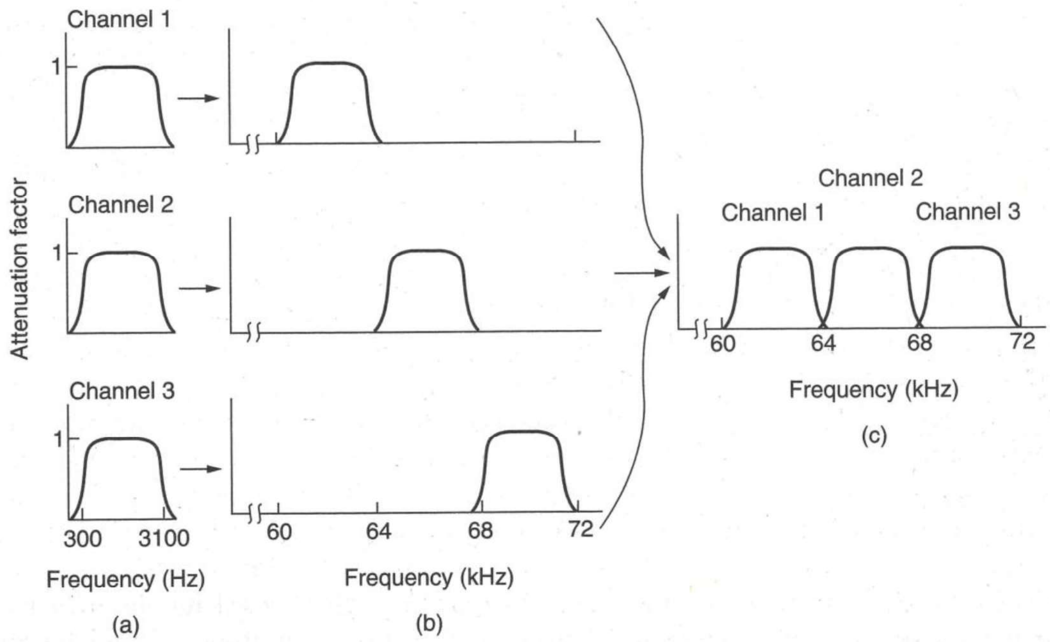

Fig. 2-24. Frequency division multiplexing. (a) The original bandwidths. (b) The bandwidths raised in frequency. (c) The multiplexed channel.

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 **supergroup**. 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.

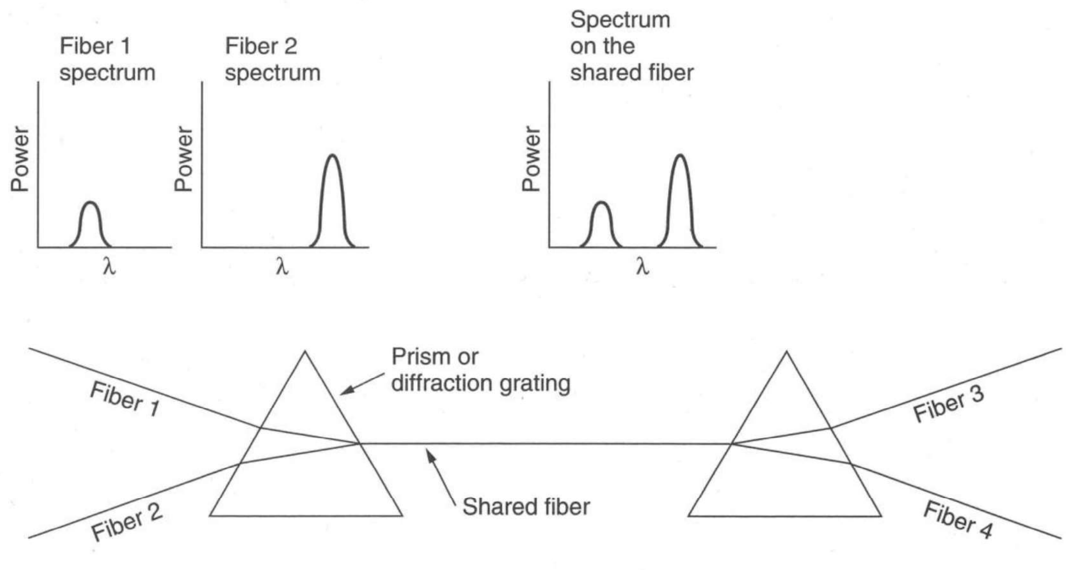

Fig. 2-25. Wavelength division multiplexing.

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 μsec/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 μsec.

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.

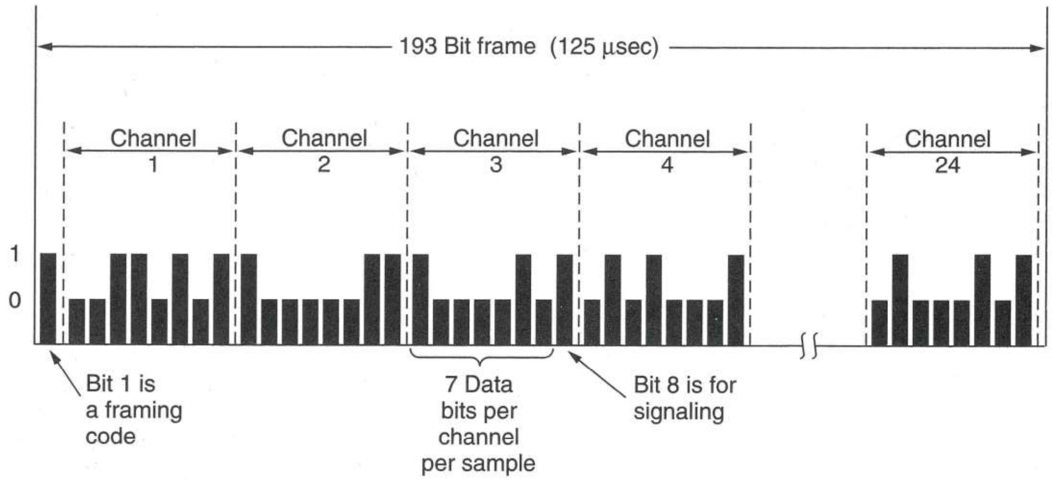

Fig. 2-26. The T1 carrier (1.544 Mbps).

A frame consists of $24 \times 8 = 192$ bits, plus one extra bit for framing, yielding 193 bits every 125 μsec. This gives a gross data rate of 1.544 Mbps. The 193rd bit is used for frame synchronization. It takes on the pattern 0101010101 . . . . 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-μsec 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.

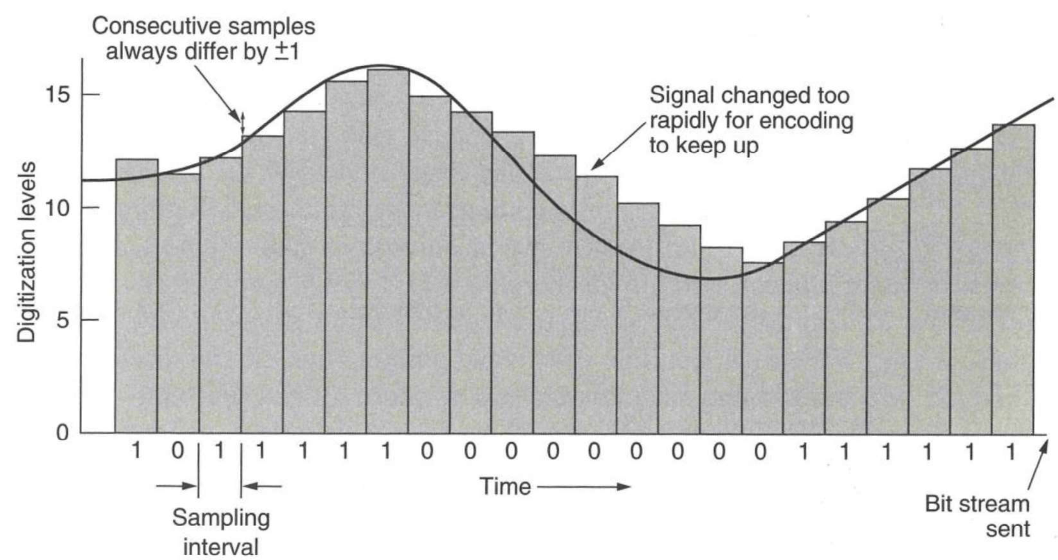

Fig. 2-27. Delta modulation.

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.

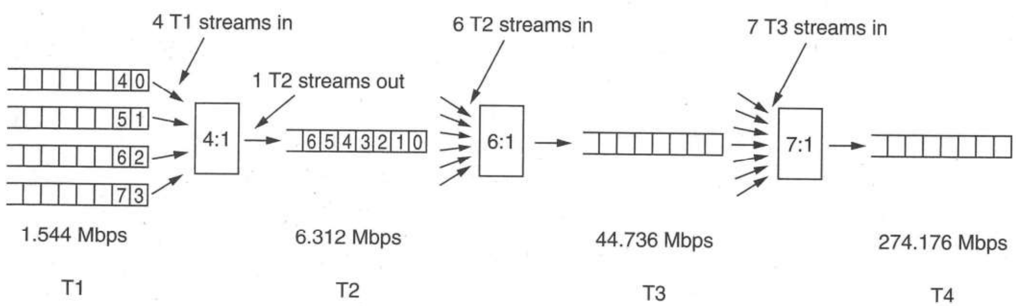

Fig. 2-28. Multiplexing T1 streams onto higher carriers.

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.

SONET/SDH

In the early days of fiber optics, every telephone company had its own proprietary optical TDM system. After AT&T was broken up in 1984, local telephone companies had to connect to multiple long-distance carriers, all with different optical TDM systems, so the need for standardization became obvious. In 1985, Bellcore, the RBOCs research arm, began working on a standard, called **SONET (Synchronous Optical NETwork)**. Later, CCITT joined the effort, which resulted in a SONET standard and a set of parallel CCITT recommendations (G.707, G.708, and G.709) in 1989. The CCITT recommendations are called **SDH (Synchronous Digital Hierarchy)** but differ from SONET only in minor ways. Virtually all the long-distance telephone traffic in the United States, and much of it elsewhere now uses trunks running SONET in the physical layer. As SONET chips become cheaper, SONET interface boards for computers may become more widespread, so it may become easier for companies to plug their computers directly into the heart of the telephone network over specially conditioned leased lines. Below we will discuss the goals and design of SONET briefly. For additional information see (Bellamy, 1991; and Omidyar and Aldridge, 1993).

The SONET design had four major goals. First and foremost, SONET had to make it possible for different carriers to interwork. Achieving this goal required defining a common signaling standard with respect to wavelength, timing, framing structure, and other issues.

Second, some means was needed to unify the U.S., European, and Japanese digital systems, all of which were based on 64-kbps PCM channels, but all of which combined them in different (and incompatible) ways.

Third, SONET had to provide a way to multiplex multiple digital channels together. At the time SONET was devised, the highest speed digital carrier actually used widely in the United States was T3, at 44.736 Mbps. T4 was defined, but not used much, and nothing was even defined above T4 speed. Part of SONET's mission was to continue the hierarchy to gigabits/sec and beyond. A standard way to multiplex slower channels into one SONET channel was also needed.

Fourth, SONET had to provide support for operations, administration, and maintenance (OAM). Previous systems did not do this very well.

An early decision was to make SONET a traditional TDM system, with the entire bandwidth of the fiber devoted to one channel containing time slots for the various subchannels. As such, SONET is a synchronous system. It is controlled by a master clock with an accuracy of about 1 part in 10^9. Bits on a SONET line are sent out at extremely precise intervals, controlled by the master clock.

When cell switching was later proposed to be the basis of broadband ISDN, the fact that it permitted irregular cell arrivals got it labeled as *asynchronous* transfer mode (i.e., ATM) to contrast it to the synchronous operation of SONET.

A SONET system consists of switches, multiplexers, and repeaters, all connected by fiber. A path from a source to destination with one intermediate multiplexer and one intermediate repeater is shown in Fig. 2-29. In SONET terminology, a fiber going directly from any device to any other device, with nothing in between, is called a **section**. A run between two multiplexers (possibly with one or more repeaters in the middle) is called a **line**. Finally, the connection between the source and destination (possibly with one or more multiplexers and repeaters) is called a **path**. The SONET topology can be a mesh, but is often a dual ring.

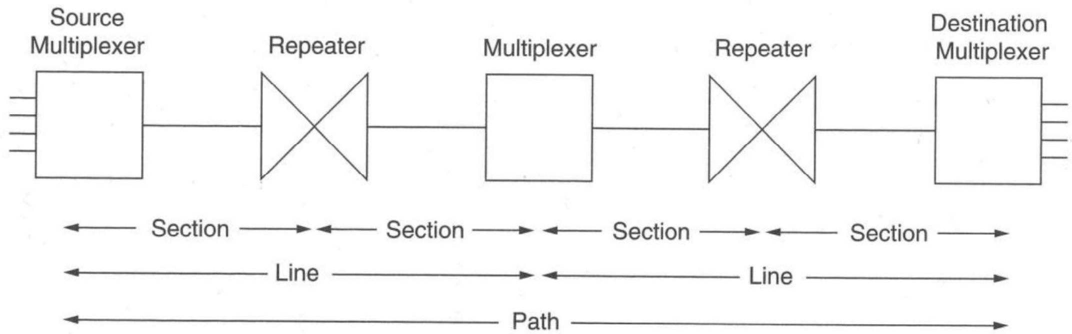

Fig. 2-29. A SONET path.

The basic SONET frame is a block of 810 bytes put out every 125 μsec. Since SONET is synchronous, frames are emitted whether or not there are any useful data to send. Having 8000 frames/sec exactly matches the sampling rate of the PCM channels used in all digital telephony systems.

The 810-byte SONET frames are best described as a rectangle of bytes, 90 columns wide by 9 rows high. Thus $8 \times 810 = 6480$ bits are transmitted 8000 times per second, for a gross data rate of 51.84 Mbps. This is the basic SONET channel and is called **STS-1** (**Synchronous Transport Signal-1**). All SONET trunks are a multiple of STS-1.

The first three columns of each frame are reserved for system management information, as illustrated in Fig. 2-30. The first three rows contain the section overhead; the next six contain the line overhead. The section overhead is generated and checked at the start and end of each section, whereas the line overhead is generated and checked at the start and end of each line.

The remaining 87 columns hold $87 \times 9 \times 8 \times 8000 = 50.112$ Mbps of user data. However, the user data, called the **SPE** (**Synchronous Payload Envelope**) do not always begin in row 1, column 4. The SPE can begin anywhere within the frame. A pointer to the first byte is contained in the first row of the line overhead. The first column of the SPE is the path overhead (i.e., header for the end-to-end path sublayer protocol).

The ability to allow the SPE to begin anywhere within the SONET frame, and even to span two frames, as shown in Fig. 2-30, gives added flexibility to the

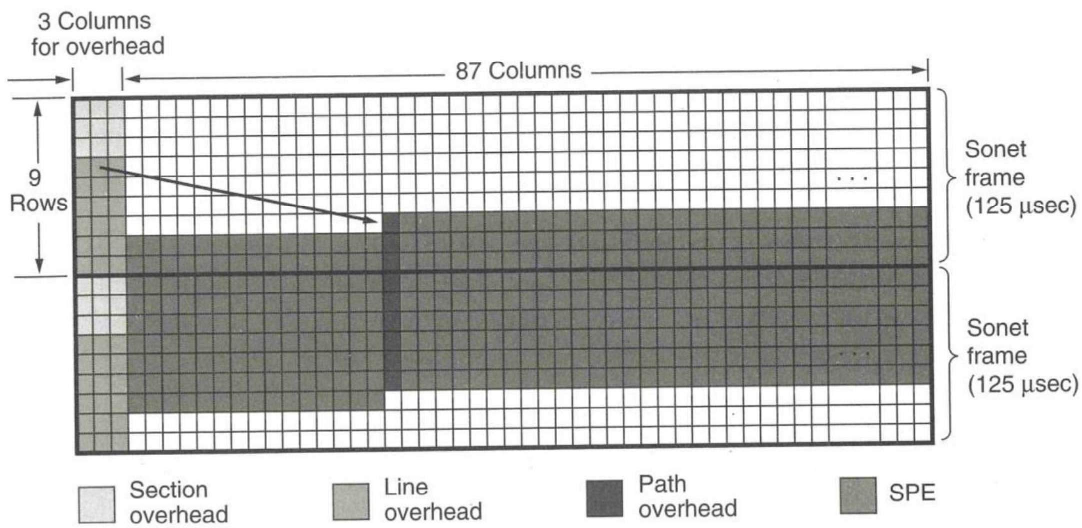

Fig. 2-30. Two back-to-back SONET frames.

system. For example, if a payload arrives at the source while a dummy SONET frame is being constructed, it can be inserted into the current frame, instead of being held until the start of the next one. This feature is also useful when the payload does not fit exactly in one frame, as in the case of a sequence of 53-byte ATM cells. The first row of the line overhead can then point to the start of the first full cell, to provide synchronization.

The section, line, and path overheads contain a profusion of bytes used for operations, administration, and maintenance. Since each byte occurs 8000 times per second, it represents a PCM channel. Three of these are, in fact, used to provide voice channels for section, line, and path maintenance personnel. Other bytes are used for framing, parity, error monitoring, IDs, clocking, synchronization, and other functions. Bellamy (1991) describes all the fields in detail.

The multiplexing of multiple data streams, called **tributaries**, plays an important role in SONET. Multiplexing is illustrated in Fig. 2-31. On the left, we start with various low-speed input streams, which are converted to the basic STS-1 SONET rate, in most cases by adding filler to round up to 51.84 Mbps. Next, three STS-1 tributaries are multiplexed onto one 155.52-Mbps STS-3 output stream. This stream, in turn, is multiplexed with three others onto a final output stream having 12 times the capacity of the STS-1 stream. At this point the signal is scrambled, to prevent long runs of 0s or 1s from interfering with the clocking, and converted from an electrical to an optical signal.

Multiplexing is done byte for byte. For example, when three STS-1 tributaries at 51.84 Mbps are merged into one STS-3 stream at 155.52 Mbps, the multiplexer first outputs 1 byte from tributary 1, then 1 from tributary 2, and finally 1 from tributary 3, before going back to 1. The STS-3 figure analogous to Fig. 2-30

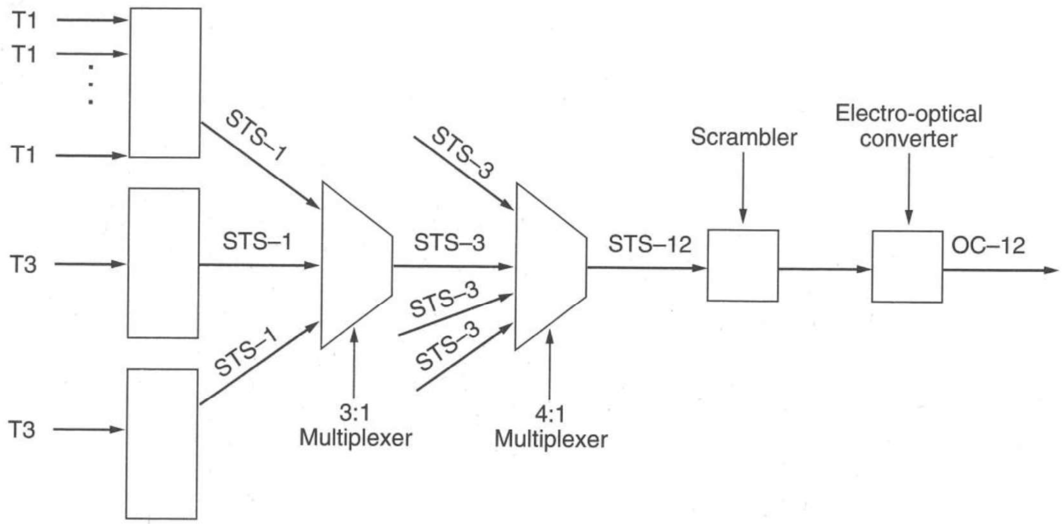

Fig. 2-31. Multiplexing in SONET.

shows (from left to right) columns from tributaries 1, 2, and 3, in that order, then another triple, and so on, out to column 270. One of these 270×9 byte frames is sent every 125 μsec, giving the 155.52-Mbps data rate.

The SONET multiplexing hierarchy is shown in Fig. 2-32. Rates from STS-1 to STS-48 have been defined. The optical carrier corresponding to STS-*n* is called OC-*n* but is bit-for-bit the same except for the scrambling shown in Fig. 2-31. The SDH names are different, and they start at OC-3 because CCITT-based systems do not have a rate near 51.84 Mbps. The OC-9 carrier is present because it closely matches the speed of a major high-speed trunk used in Japan. OC-18 and OC-36 will be used in Japan in the future. The gross data rate includes all the overhead. The SPE data rate excludes the line and section overhead. The user data rate excludes all overhead and only counts the 86 columns available for the payload.

As an aside, when a carrier, such as OC-3, is not multiplexed, but carries the data from only a single source, the letter *c* (for concatenated) is appended to the designation, so OC-3 indicates a 155.52-Mbps carrier consisting of three separate OC-1 carriers, but OC-3c indicates a data stream from a single source at 155.52 Mbps. The three OC-1 streams within an OC-3c stream are interleaved by column, first column 1 from stream 1, then column 1 from stream 2, then column 1 from stream 3, followed by column 2 from stream 1, and so on, leading to a frame 270 columns wide and 9 rows deep.

The amount of actual user data in an OC-3c stream is slightly higher than in an OC-3 stream (149.760 Mbps versus 148.608 Mbps) because the path overhead column is included inside the SPE only once, instead of the three times it would be with three independent OC-1 streams. In other words, 260 of the 270 columns

SONET		SDH	Data rate (Mbps)		
Electrical	Optical	Optical	Gross	SPE	User
STS-1	OC-1		51.84	50.112	49.536
STS-3	OC-3	STM-1	155.52	150.336	148.608
STS-9	OC-9	STM-3	466.56	451.008	445.824
STS-12	OC-12	STM-4	622.08	601.344	594.432
STS-18	OC-18	STM-6	933.12	902.016	891.648
STS-24	OC-24	STM-8	1244.16	1202.688	1188.864
STS-36	OC-36	STM-12	1866.24	1804.032	1783.296
STS-48	OC-48	STM-16	2488.32	2405.376	2377.728

Fig. 2-32. SONET and SDH multiplex rates.

are available for user data in OC-3c, whereas only 258 columns are available for user data in OC-3. Higher-order concatenated frames (e.g., OC-12c) also exist.

By now it should be clear why ATM runs at 155 Mbps: the intention is to carry ATM cells over SONET OC-3c trunks. It should also be clear that the widely quoted 155-Mbps figure is the gross rate, including the SONET overhead. Furthermore, somewhere along the way somebody incorrectly rounded 155.52 Mbps to 155 Mbps instead of 156 Mbps, and now everyone else does it wrong, too.

The SONET physical layer is divided up into four sublayers, as shown in Fig. 2-33. The lowest sublayer is the **photonic sublayer**. It is concerned with specifying the physical properties of the light and fiber to be used.

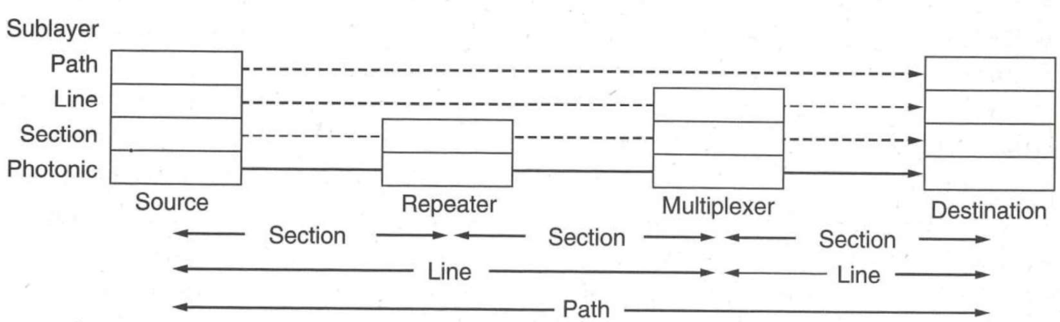

Fig. 2-33. The SONET architecture.

The three remaining sublayers correspond to the sections, lines, and paths. The section sublayer handles a single point-to-point fiber run, generating a standard frame at one end and processing it at the other. Sections can start and end at repeaters, which just amplify and regenerate the bits, but do not change or process them in any way.

The line sublayer is concerned with multiplexing multiple tributaries onto a single line and demultiplexing them at the other end. To the line sublayer, the repeaters are transparent. When a multiplexer puts out bits on a fiber, it expects them to arrive at the next multiplexer unchanged, no matter how many repeaters are used in between. The protocol in the line sublayer is thus between two multiplexers and deals with issues such as how many inputs are being multiplexed together and how. In contrast, the path sublayer and protocol deal with end-to-end issues.

2.4.5. Switching

From the point of view of the average telephone engineer, the phone system is divided into two parts: outside plant (the local loops and trunks, since they are outside the switching offices), and inside plant (the switches). We have just looked at outside plant. Now it is time to examine inside plant.

Two different switching techniques are used inside the telephone system: circuit switching and packet switching. We will give a brief introduction to each of them below. Then we will go into circuit switching in detail, because that is how the current telephone system works. Later in the chapter we will go into packet switching in detail in the context of the next generation telephone system, broadband ISDN.

Circuit Switching

When you or your computer places a telephone call, the switching equipment within the telephone system seeks out a physical "copper" (including fiber and radio) path all the way from your telephone to the receiver's telephone. This technique is called **circuit switching** and is shown schematically in Fig. 2-34(a). Each of the six rectangles represents a carrier switching office (end office, toll office, etc.). In this example, each office has three incoming lines and three outgoing lines. When a call passes through a switching office, a physical connection is (conceptually) established between the line on which the call came in and one of the output lines, as shown by the dotted lines.

In the early days of the telephone, the connection was made by having the operator plug a jumper cable into the input and output sockets. In fact, there is a surprising little story associated with the invention of automatic circuit switching equipment. It was invented by a 19th Century undertaker named Almon B. Strowger. Shortly after the telephone was invented, when someone died, one of the survivors would call the town operator and say: "Please connect me to an undertaker." Unfortunately for Mr. Strowger, there were two undertakers in his town, and the other one's wife was the town telephone operator. He quickly saw that either he was going to have to invent automatic telephone switching equipment or he was going to go out of business. He chose the first option. For nearly 100 years, the circuit switching equipment used worldwide was known as Strowger gear. (History does not record whether the now-unemployed switchboard operator got a job as an information operator, answering questions such as: What is the phone number of an undertaker?

The model shown in Fig. 2-34(a) is highly simplified of course, because parts of the "copper" path between the two telephones may, in fact, be microwave links onto which thousands of calls are multiplexed. Nevertheless, the basic idea is valid: once a call has been set up, a dedicated path between both ends exists and will continue to exist until the call is finished.

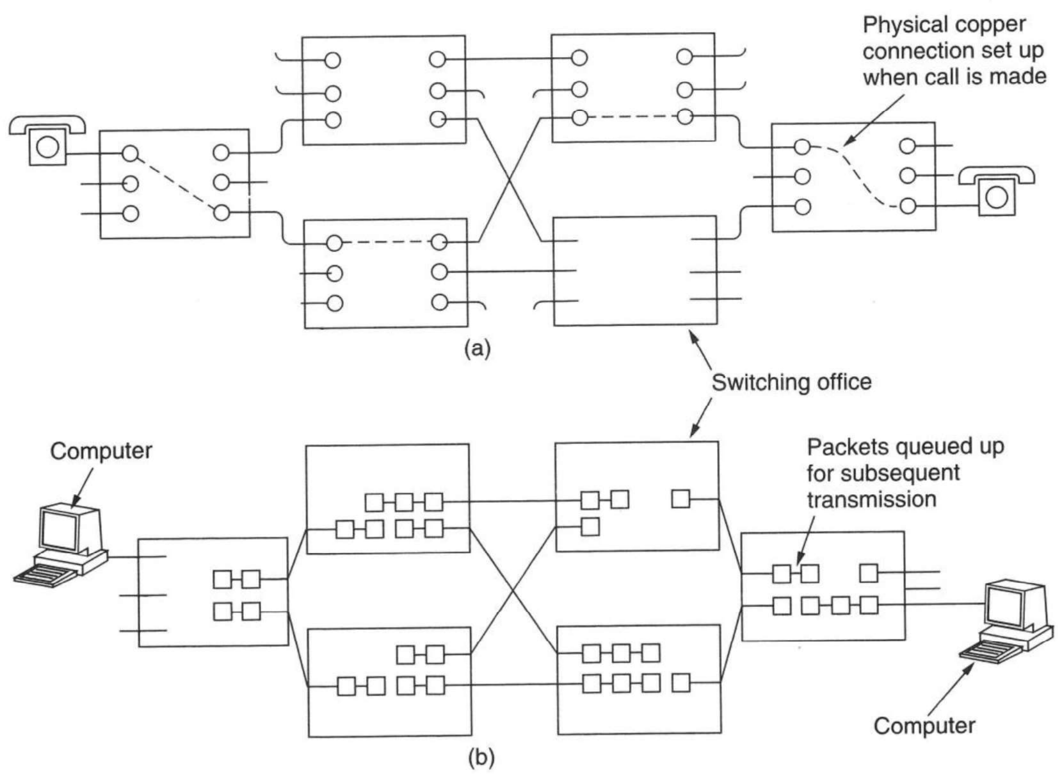

Fig. 2-34. (a) Circuit switching. (b) Packet switching.

An important property of circuit switching is the need to set up an end-to-end path *before* any data can be sent. The elapsed time between the end of dialing and the start of ringing can easily be 10 sec, more on long-distance or international calls. During this time interval, the telephone system is hunting for a copper path, as shown in Fig. 2-35(a). Note that before data transmission can even begin, the call request signal must propagate all the way to the destination, and be acknowledged. For many computer applications (e.g., point-of-sale credit verification), long setup times are undesirable.

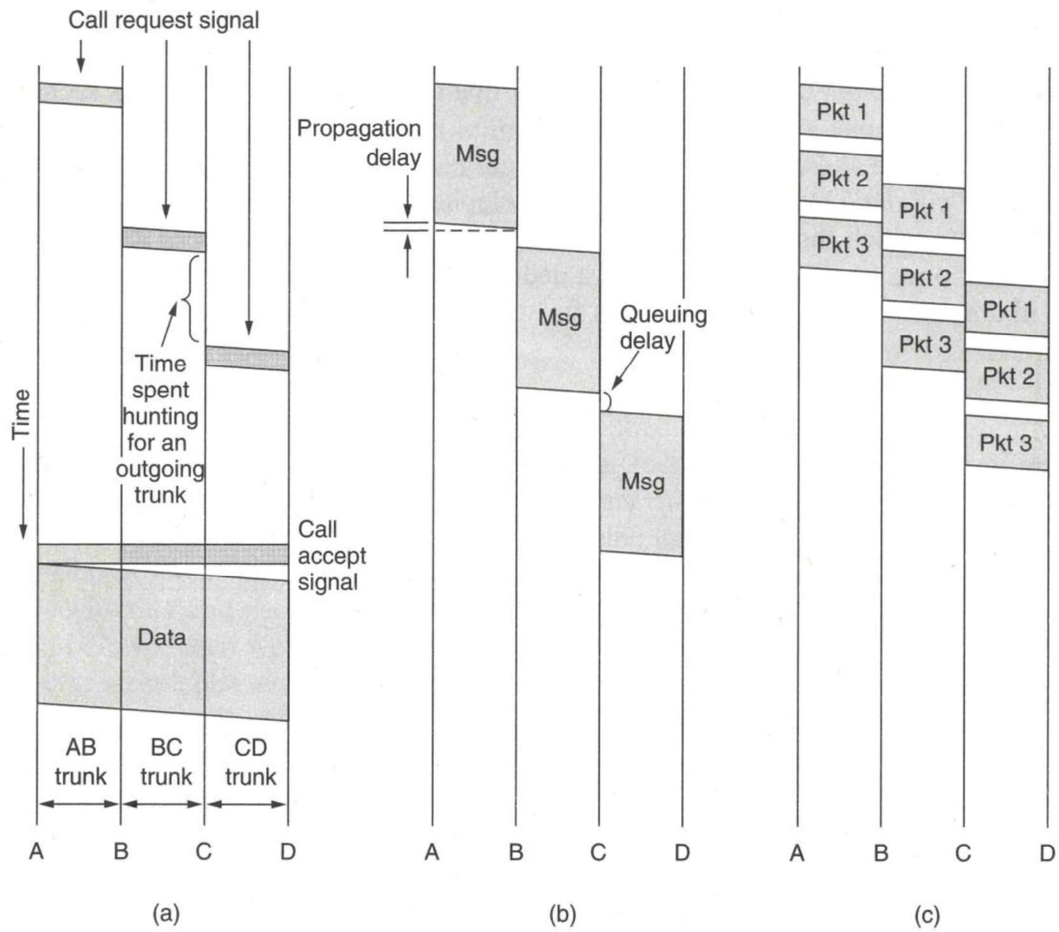

Fig. 2-35. Timing of events in (a) circuit switching, (b) message switching, (c) packet switching.

As a consequence of the copper path between the calling parties, once the setup has been completed, the only delay for data is the propagation time for the electromagnetic signal, about 5 msec per 1000 km. Also as a consequence of the established path, there is no danger of congestion—that is, once the call has been put through, you never get busy signals, although you might get one before the connection has been established due to lack of switching or trunk capacity.

An alternative switching strategy is **message switching**, shown in Fig. 2-35(b). When this form of switching is used, no physical copper path is established in advance between sender and receiver. Instead, when the sender has a block of data to be sent, it is stored in the first switching office (i.e., router) and then forwarded later, one hop at a time. Each block is received in its entirety, inspected for errors, and then retransmitted. A network using this technique is called a **store-and-forward** network, as mentioned in Chap. 1.

The first electromechanical telecommunication systems used message switching, namely for telegrams. The message was punched on paper tape off-line at the sending office, and then read in and transmitted over a communication line to the next office along the way, where it was punched out on paper tape. An operator there tore the tape off and read it in on one of the many tape readers, one per outgoing trunk. Such a switching office was called a **torn tape office**.

With message switching, there is no limit on block size, which means that routers (in a modern system) must have disks to buffer long blocks. It also means that a single block may tie up a router-router line for minutes, rendering message switching useless for interactive traffic. To get around these problems, **packet switching** was invented. Packet-switching networks place a tight upper limit on block size, allowing packets to be buffered in router main memory instead of on disk. By making sure that no user can monopolize any transmission line very long (milliseconds), packet-switching networks are well suited to handling interactive traffic. A further advantage of packet switching over message switching is shown in Fig. 2-35(b) and (c): the first packet of a multipacket message can be forwarded before the second one has fully arrived, reducing delay and improving throughput. For these reasons, computer networks are usually packet switched, occasionally circuit switched, but never message switched.

Circuit switching and packet switching differ in many respects. The key difference is that circuit switching statically reserves the required bandwidth in advance, whereas packet switching acquires and releases it as it is needed. With circuit switching, any unused bandwidth on an allocated circuit is just wasted. With packet switching it may be utilized by other packets from unrelated sources going to unrelated destinations, because circuits are never dedicated. However, just because no circuits are dedicated, a sudden surge of input traffic may overwhelm a router, exceeding its storage capacity and causing it to lose packets.

In contrast, with circuit switching, when packet switching is used, it is straightforward for the routers to provide speed and code conversion. Also, they can provide error correction to some extent. In some packet-switched networks, however, packets may be delivered in the wrong order to the destination. Reordering of packets can never happen with circuit switching.

Another difference is that circuit switching is completely transparent. The sender and receiver can use any bit rate, format, or framing method they want to. The carrier does not know or care. With packet switching, the carrier determines the basic parameters. A rough analogy is a road versus a railroad. In the former, the user determines the size, speed, and nature of the vehicle; in the latter, the carrier does. It is this transparency that allows voice, data, and fax to coexist within the phone system.

A final difference between circuit and packet switching is the charging algorithm. Packet carriers usually base their charge on both the number of bytes (or packets) carried and the connect time. Furthermore, transmission distance usually does not matter, except perhaps internationally. With circuit switching, the charge is based on the distance and time only, not the traffic. The differences are summarized in Fig. 2-36.

Item	Circuit-switched	Packet-switched
Dedicated "copper" path	Yes	No
Bandwidth available	Fixed	Dynamic
Potentially wasted bandwidth	Yes	No
Store-and-forward transmission	No	Yes
Each packet follows the same route	Yes	No
Call setup	Required	Not needed
When can congestion occur	At setup time	On every packet
Charging	Per minute	Per packet

Fig. 2-36. A comparison of circuit-switched and packet-switched networks.

Both circuit switching and packet switching are so important, we will come back to them shortly and describe the various technologies used in detail.

The Switch Hierarchy

It is worth saying a few words about how the routing between switches is done within the current circuit-switched telephone system. We will describe the AT&T system here, but other companies and countries use the same general principles. The telephone system has five classes of switching offices, as illustrated in Fig. 2-37. There are 10 regional switching offices, and these are fully interconnected by 45 high-bandwidth fiber optic trunks. Below the regional offices are 67 sectional offices, 230 primary offices, 1300 toll offices, and 19,000 end offices. The lower four levels were originally connected as a tree.

Calls are generally connected at the lowest possible level. Thus if a subscriber connected to end office 1 calls another subscriber connected to end office 1, the call will be completed in that office. However, a call from a customer attached to end office 1 in Fig. 2-37 to a customer attached to end office 2 will have to go toll office 1. However, a call from end office 1 to end office 4 will have to go up to primary office 1, and so on. With a pure tree, there is only one minimal route, and that would normally be taken.

During years of operation, the telephone companies noticed that some routes were busier than others. For example, there were many calls from New York to Los Angeles. Rather than go all the way up the hierarchy, they simply installed **direct trunks** for the busy routes. A few of these are shown in Fig. 2-37 as

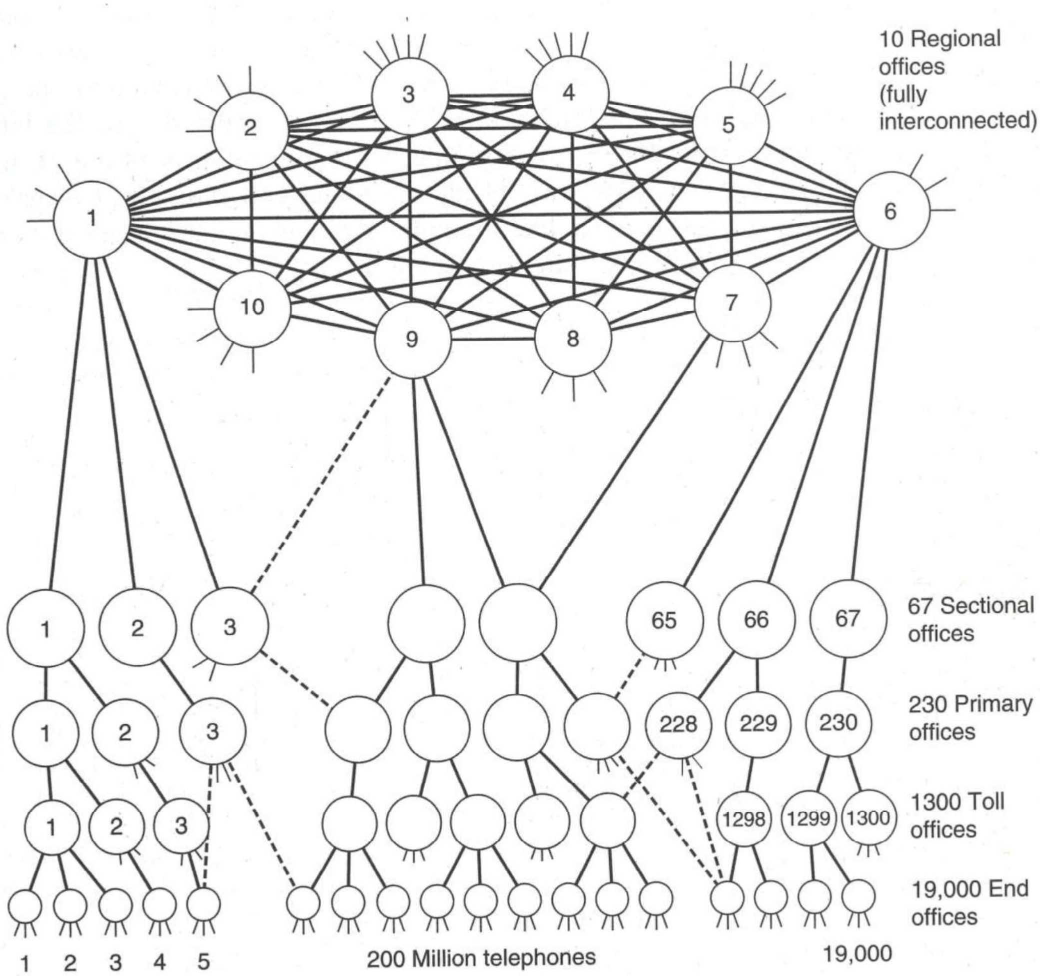

Fig. 2-37. The AT&T telephone hierarchy. The dashed lines are direct trunks.

dashed lines. As a consequence, many calls can now be routed along many paths. The actual route chosen is generally the most direct one, but if the necessary trunks along it are full, an alternative is chosen. This complex routing is now possible because a switching machine, like the AT&T 5 ESS, is in fact just a general purpose computer with a large amount of very specialized I/O equipment.

Crossbar Switches

Let us now turn from how calls are routed among switches to how individual switches actually work inside. Several kinds of switches are (or were) common within the telephone system. The simplest kind is the **crossbar switch** (also called a **crosspoint switch),** shown in Fig. 2-38. In a switch with n input lines and n output lines (i.e., n full duplex lines), the crossbar switch has n^2

intersections, called **crosspoints**, where an input and an output line may be connected by a semiconductor switch, as shown in Fig. 2-38(a). In Fig. 2-38(b) we see an example in which line 0 is connected to line 4, line 1 is connected to line 7, and line 2 is connected to line 6. Lines 3 and 5 are not connected. All the bits that arrive at the switch from line 4, for example, are immediately sent out of the switch on line 0. Thus the crossbar switch implements circuit switching by making a direct electrical connection, just like the jumper cables in the first-generation switches, only automatically and within microseconds.

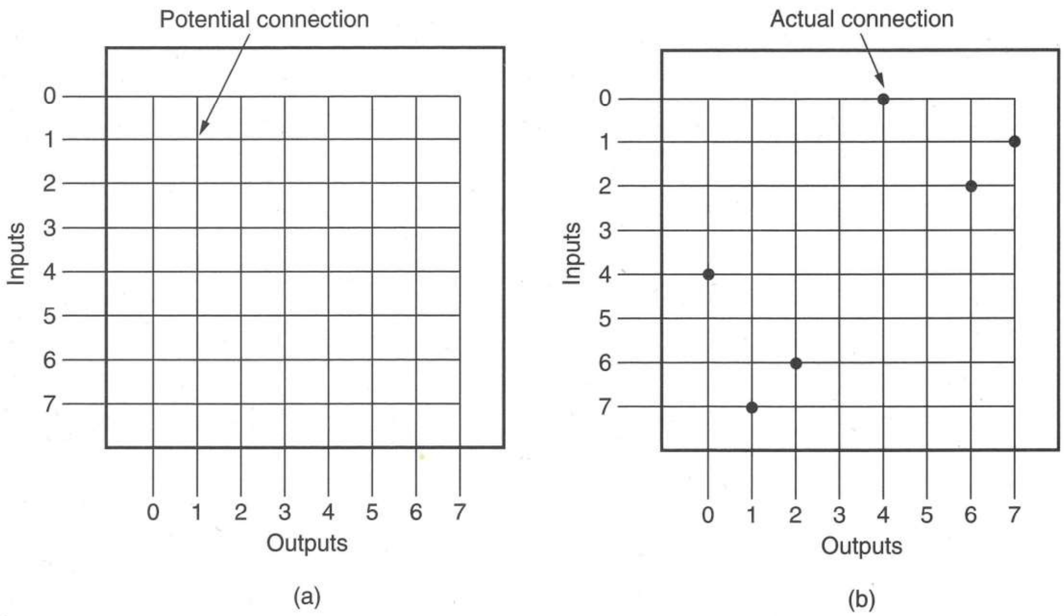

Fig. 2-38. (a) A crossbar switch with no connections. (b) A crossbar switch with three connections set up: 0 with 4, 1 with 7, and 2 with 6.

The problem with a crossbar switch is that the number of crossbars grows as the square of the number of lines into the switch. If we assume that all lines are full duplex and that there are no self-connections, only the crosspoints above the diagonal are needed. Still, $n(n-1)/2$ crosspoints are needed. For $n = 1000$, we need 499,500 crosspoints. While building a VLSI chip with this number of transistor switches is possible, having 1000 pins on the chip is not. Thus a single crossbar switch is only useful for relatively small end offices.

Space Division Switches

By splitting the crossbar switch into small chunks and interconnecting them, it is possible to build multistage switches with many fewer crosspoints. These are called **space division switches**. Two configurations are illustrated in Fig. 2-39.

To keep our example simple, we will consider only three-stage switches, but

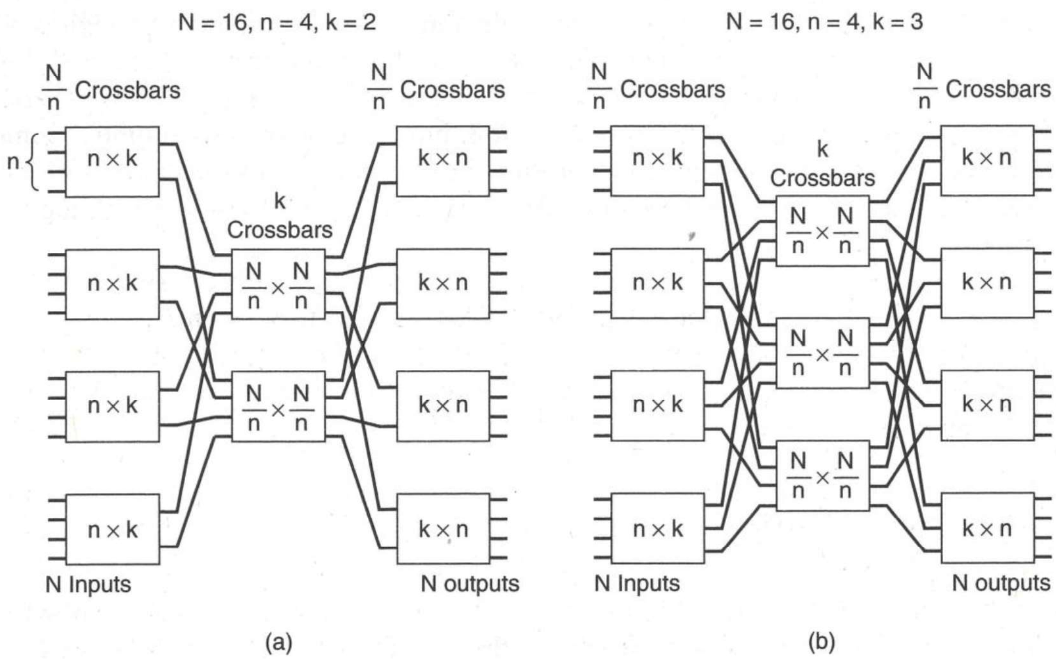

Fig. 2-39. Two space division switches with different parameters.

switches with more stages are also possible. In these examples, we have a total of N inputs and N outputs. Instead of building a single $N \times N$ crossbar, we build the switch out of smaller rectangular crossbars. In the first stage, each crossbar has n inputs, so we need N/n of them to handle all N incoming lines.

The second stage has k crossbars, each with N/n inputs and N/n outputs. The third stage is a repeat of the first stage, but reversed left to right. Each intermediate crossbar is connected to each input crossbar and each output crossbar. Consequently, it is possible to connect every input to every output using either the first intermediate crossbar in Fig. 2-39(a) or using the second one. In fact, there are two disjoint paths from each input to each output, depending which intermediate crossbar is chosen. In Fig. 2-39(b) there are three paths for each input/output pair. With k intermediate stages (k is a design parameter), there are k disjoint paths.

Let us now compute the number of crosspoints needed for a three-stage switch. In the first stage, there are N/n crossbars, each with nk crosspoints, for a total of Nk. In the second stage, there are k crossbars, each with $(N/n)^2$ crosspoints. The third stage is the same as the first. Adding up the three stages, we get

$$\text{Number of crosspoints} = 2kN + k(N/n)^2$$

For $N = 1000$, $n = 50$ and $k = 10$, we need only 24,000 crosspoints instead of the 499,500 required by a 1000×1000 single-stage crossbar.

Unfortunately, as usual, there is no free lunch. The switch can block. Consider Fig. 2-39(a) again. Stage 2 has eight inputs, so a maximum of eight calls can be connected at once. When call nine comes by, it will have to get a busy signal, even though the destination is available. The switch of Fig. 2-39(b) is better, handling a maximum of 12 calls instead of 8, but it uses more crosspoints. Sometimes when making a phone call you may have gotten a busy signal before you finished dialing. This was probably caused by blocking part way through the network.

It should be obvious that the larger k is, the more expensive the switch and the lower the blocking probability. In 1953, Clos showed that when $k = 2n - 1$, the switch will never block (Clos, 1953). Other researchers have analyzed calling patterns in great detail to construct switches that theoretically can block but do so only rarely in practice.

Time Division Switches

A completely different kind of switch is the **time division switch**, shown in Fig. 2-40. With time division switching, the n input lines are scanned in sequence to build up an input frame with n slots. Each slot has k bits. For T1 switches, the slots are 8 bits, with 8000 frames processed per second.

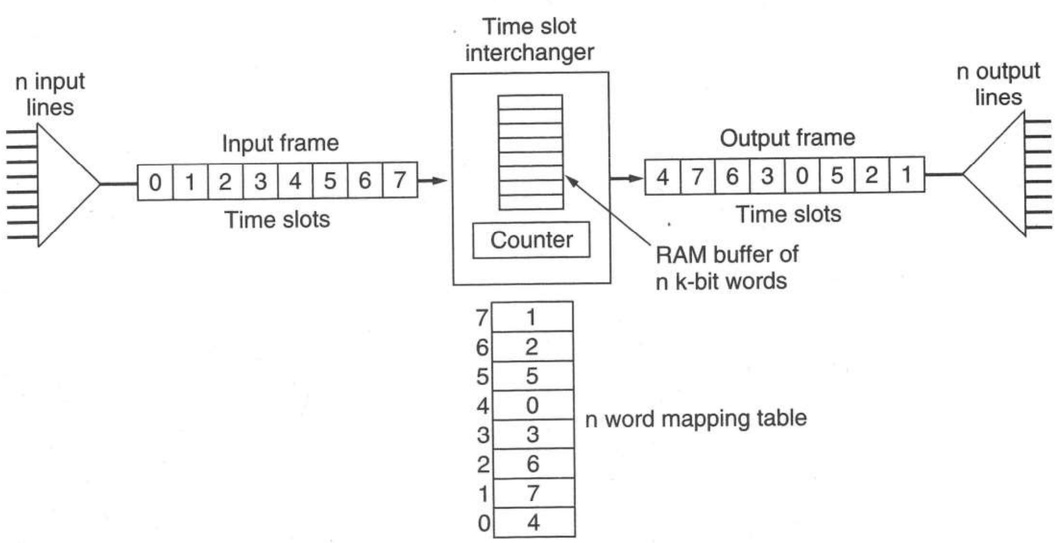

Fig. 2-40. A time division switch.

The heart of the time division switch is the **time slot interchanger**, which accepts input frames and produces output frames in which the time slots have been reordered. In Fig. 2-40, input slot 4 is output first, then slot 7, and so on. Finally, the output frame is demultiplexed, with output slot 0 (input slot 4) going to line 0, and so on. In essence, the switch has moved a byte from input line 4 to output line 0, another byte from input line 7 to output line 1, and so on. Viewed from the outside, the whole arrangement is a circuit switch, even though there are no physical connections.

The time slot interchanger works as follows: When an input frame is ready to be processed, each slot (i.e., each byte in the input frame) is written into a RAM buffer inside the interchanger. The slots are written in order, so buffer word i contains slot i.

After all the slots of the input frame have been stored in the buffer, the output frame is constructed by reading out the words again, but in a different order. A counter goes from 0 to $n - 1$. At step j, the contents of word j of a mapping table is read out and used to address the RAM table. Thus if word 0 of the mapping table contains a 4, word 4 of the RAM buffer will be read out first, and the first slot of the output frame will be slot 4 of the input frame. Thus the contents of the mapping table determine which permutation of the input frame will be generated as the output frame, and thus which input line is connected to which output line.

Time division switches use tables that are linear in the number of lines, rather than quadratic, but they have another limitation. It is necessary to store n slots in the buffer RAM and then read them out again within one frame period of 125 μsec. If each of these memory accesses takes T microsec, the time needed to process a frame is $2nT$ microsec, so we have $2nT = 125$ or $n = 125/2T$. For a memory with 100-nsec cycle time, we can support at most 625 lines. We can also turn this relation around and use it to determine the required memory cycle to support a given number of lines. As with a crossbar switch, it is possible to devise multistage switches that split the work up into several parts and then combine the results in order to handle larger numbers of lines.

2.5. NARROWBAND ISDN

For more than a century, the primary international telecommunication infrastructure has been the public circuit-switched telephone system. This system was designed for analog voice transmission and is inadequate for modern communication needs. Anticipating considerable user demand for an end-to-end digital service (i.e., not like Fig. 2-17 which is part digital and part analog), the world's telephone companies and PTTs got together in 1984 under the auspices of CCITT and agreed to build a new, fully digital, circuit-switched telephone system by the early part of the 21st Century. This new system, called **ISDN** (**Integrated Services Digital Network**), has as its primary goal the integration of voice and nonvoice services. It is already available in many locations and its use is growing slowly. In the following sections we will describe what it does and how it works. For further information, see (Dagdeviren et al., 1994; and Kessler, 1993).

What is needed is a mechanism based on the subnet's carrying capacity rather than on the receiver's buffering capacity. Clearly, the flow control mechanism must be applied at the sender to prevent it from having too many unacknowledged TPDUs outstanding at once. Belsnes (1975) proposed using a sliding window flow control scheme in which the sender dynamically adjusts the window size to match the network's carrying capacity. If the network can handle c TPDUs/sec and the cycle time (including transmission, propagation, queueing, processing at the receiver, and return of the acknowledgement) is r, then the sender's window should be cr. With a window of this size the sender normally operates with the pipeline full. Any small decrease in network performance will cause it to block.

In order to adjust the window size periodically, the sender could monitor both parameters and then compute the desired window size. The carrying capacity can be determined by simply counting the number of TPDUs acknowledged during some time period and then dividing by the time period. During the measurement, the sender should send as fast as it can, to make sure that the network's carrying capacity, and not the low input rate, is the factor limiting the acknowledgement rate. The time required for a transmitted TPDU to be acknowledged can be measured exactly and a running mean maintained. Since the capacity of the network depends on the amount of traffic in it, the window size should be adjusted frequently, to track changes in the carrying capacity. As we will see later, the Internet uses a similar scheme.

6.2.5. Multiplexing

Multiplexing several conversations onto connections, virtual circuits, and physical links plays a role in several layers of the network architecture. In the transport layer the need for multiplexing can arise in a number of ways. For example, in networks that use virtual circuits within the subnet, each open connection consumes some table space in the routers for the entire duration of the connection. If buffers are dedicated to the virtual circuit in each router as well, a user who left a terminal logged into a remote machine during a coffee break is nevertheless consuming expensive resources. Although this implementation of packet switching defeats one of the main reasons for having packet switching in the first place—to bill the user based on the amount of data sent, not the connect time—many carriers have chosen this approach because it so closely resembles the circuit switching model to which they have grown accustomed over the decades.

The consequence of a price structure that heavily penalizes installations for having many virtual circuits open for long periods of time is to make multiplexing of different transport connections onto the same network connection attractive. This form of multiplexing, called **upward multiplexing**, is shown in Fig. 6-17(a). In this figure, four distinct transport connections all use the same network connection (e.g., ATM virtual circuit) to the remote host. When connect time forms the major component of the carrier's bill, it is up to the transport layer to group transport connections according to their destination and map each group onto the minimum number of network connections. If too many transport connections are mapped onto one network connection, the performance will be poor, because the window will usually be full, and users will have to wait their turn to send one message. If too few transport connections are mapped onto one network connection, the service will be expensive. When upward multiplexing is used with ATM, we have the ironic (tragic?) situation of having to identify the connection using a field in the transport header, even though ATM provides more than 4000 virtual circuit numbers per virtual path expressly for that purpose.

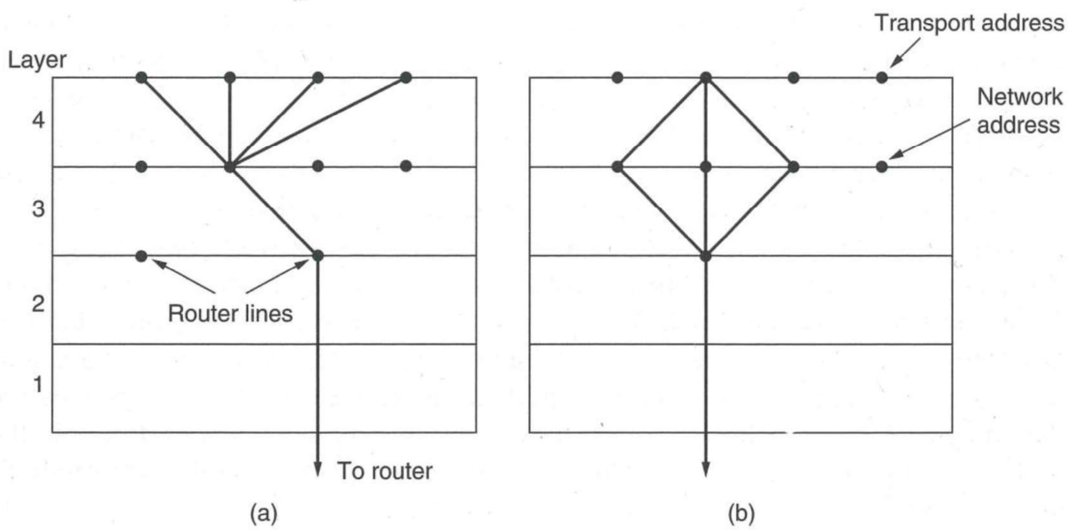

Fig. 6-17. (a) Upward multiplexing. (b) Downward multiplexing.

Multiplexing can also be useful in the transport layer for another reason, related to carrier technical decisions rather than carrier pricing decisions. Suppose, for example, that a certain important user needs a high-bandwidth connection from time to time. If the subnet enforces a sliding window flow control with an n-bit sequence number, the user must stop sending as soon as $2^n - 1$ packets are outstanding and must wait for the packets to propagate to the remote host and be acknowledged. If the physical connection is via a satellite, the user is effectively limited to $2^n - 1$ packets every 540 msec. With, for example, $n = 8$ and 128-byte packets, the usable bandwidth is about 484 kbps, even though the physical channel bandwidth is more than 100 times higher.

One possible solution is to have the transport layer open multiple network connections and distribute the traffic among them on a round-robin basis, as indicated in Fig. 6-17(b). This modus operandi is called **downward multiplexing**. With k network connections open, the effective bandwidth is increased by a factor of k. With 4095 virtual circuits, 128-byte packets, and an 8-bit sequence number, it is theoretically possible to achieve data rates in excess of 1.6 Gbps. Of course, this performance can be achieved only if the output line can support 1.6 Gbps, because all 4095 virtual circuits are still being sent out over one physical line, at least in Fig. 6-17(b). If multiple output lines are available, downward multiplexing can also be used to increase the performance even more.

6.2.6. Crash Recovery

If hosts and routers are subject to crashes, recovery from these crashes becomes an issue. If the transport entity is entirely within the hosts, recovery from network and router crashes is straightforward. If the network layer provides datagram service, the transport entities expect lost TPDUs all the time and know how to cope with them. If the network layer provides connection-oriented service, then loss of a virtual circuit is handled by establishing a new one and then probing the remote transport entity to ask it which TPDUs it has received and which ones it has not received. The latter ones can be retransmitted.

A more troublesome problem is how to recover from host crashes. In particular, it may be desirable for clients to be able to continue working when servers crash and then quickly reboot. To illustrate the difficulty, let us assume that one host, the client, is sending a long file to another host, the file server, using a simple stop-and-wait protocol. The transport layer on the server simply passes the incoming TPDUs to the transport user, one by one. Part way through the transmission, the server crashes. When it comes back up, its tables are reinitialized, so it no longer knows precisely where it was.

In an attempt to recover its previous status, the server might send a broadcast TPDU to all other hosts, announcing that it had just crashed and requesting that its clients inform it of the status of all open connections. Each client can be in one of two states: one TPDU outstanding, *S1*, or no TPDUs outstanding, *S0*. Based on only this state information, the client must decide whether or not to retransmit the most recent TPDU.

At first glance it would seem obvious: the client should retransmit only if it has an unacknowledged TPDU outstanding (i.e., is in state *S1*) when it learns of the crash. However, a closer inspection reveals difficulties with this naive approach. Consider, for example, the situation when the server's transport entity first sends an acknowledgement, and then, when the acknowledgement has been sent, performs the write up to the application process. Writing a TPDU onto the output stream and sending an acknowledgement are two distinct indivisible events that cannot be done simultaneously. If a crash occurs after the acknowledgement has been sent but before the write has been done, the client will receive the acknowledgement and thus be in state *S0* when the crash recovery announcement arrives. The client will therefore not retransmit, (incorrectly) thinking that the TPDU has arrived. This decision by the client leads to a missing TPDU.

6.4.9. Wireless TCP and UDP

In theory, transport protocols should be independent of the technology of the underlying network layer. In particular, TCP should not care whether IP is running over fiber or over radio. In practice, it does matter because most TCP implementations have been carefully optimized based on assumptions that are true for wired networks but which fail for wireless networks. Ignoring the properties of wireless transmission can lead to a TCP implementation that is logically correct but has horrendous performance.

The principal problem is the congestion control algorithm. Nearly all TCP implementations nowadays assume that timeouts are caused by congestion, not by lost packets. Consequently, when a timer goes off, TCP slows down and sends less vigorously (e.g., Jacobson's slow start algorithm). The idea behind this approach is to reduce the network load and thus alleviate the congestion.

Unfortunately, wireless transmission links are highly unreliable. They lose packets all the time. The proper approach to dealing with lost packets is to send them again, and as quickly as possible. Slowing down just makes matters worse. If, say, 20 percent of all packets are lost, then when the sender transmits 100 packets/sec, the throughput is 80 packets/sec. If the sender slows down to 50 packets/sec, the throughput drops to 40 packets/sec.

In effect, when a packet is lost on a wired network, the sender should slow down. When one is lost on a wireless network, the sender should try harder. When the sender does not know what the network is, it is difficult to make the correct decision.

Frequently, the path from sender to receiver is inhomogeneous. The first 1000 km might be over a wired network, but the last 1 km might be wireless. Now making the correct decision on a timeout is even harder, since it matters where the problem occurred. A solution proposed by Bakne and Badrinath (1995), **indirect TCP**, is to split the TCP connection into two separate connections, as shown in Fig. 6-35. The first connection goes from the sender to the base station. The second one goes from the base station to the receiver. The base station simply copies packets between the connections in both directions.

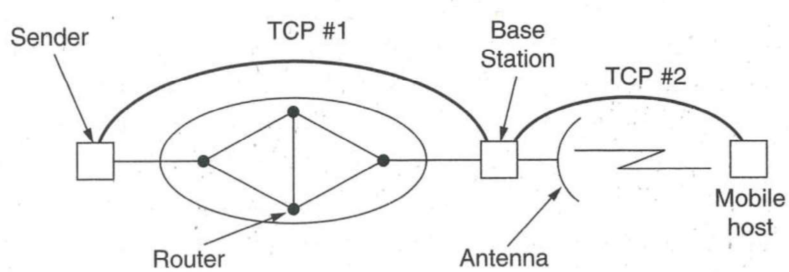

Fig. 6-35. Splitting a TCP connection into two connections.

The advantage of this scheme is that both connections are now homogeneous. Timeouts on the first connection can slow the sender down, whereas timeouts on the second one can speed it up. Other parameters can also be tuned separately for the two connections. The disadvantage is that it violates the semantics of TCP. Since each part of the connection is a full TCP connection, the base station acknowledges each TCP segment in the usual way. Only now, receipt of an acknowledgement by the sender does not mean that the receiver got the segment, only that the base station got it.

A different solution, due to Balakrishnan et al. (1995), does not break the semantics of TCP. It works by making several small modifications to the network layer code in the base station. One of the changes is the addition of a snooping agent that observes and caches TCP segments going out to the mobile host, and acknowledgements coming back from it. When the snooping agent sees a TCP segment going out to the mobile host but does not see an acknowledgement coming back before its (relatively short) timer goes off, it just retransmits that segment, without telling the source that it is doing so. It also generates a retransmission when it sees duplicate acknowledgements from the mobile host go by, invariably meaning that the mobile host has missed something. Duplicate acknowledgements are discarded on the spot, to avoid having the source misinterpret them as a sign of congestion.

One disadvantage of this transparency, however, is that if the wireless link is very lossy, the source may time out waiting for an acknowledgement and invoke the congestion control algorithm. With indirect TCP, the congestion control algorithm will never be started unless there really is congestion in the wired part of the network.

The Balakrishnan et al. paper also has a solution to the problem of lost segments originating at the mobile host. When the base station notices a gap in the inbound sequence numbers, it generates a request for a selective repeat of the missing bytes using a TCP option. Using these two fixes, the wireless link is made more reliable in both directions, without the source knowing about it, and without changing the semantics of TCP.

While UDP does not suffer from the same problems as TCP, wireless communication also introduces difficulties for it. The main trouble is that programs use UDP expecting it to be highly reliable. They know that no guarantees are given, but they still expect it to be near perfect. In a wireless environment, it will be far from perfect. For programs that are able to recover from lost UDP messages, but only at considerable cost, suddenly going from an environment where messages theoretically can be lost but rarely are, to one in which they are constantly being lost can result in a performance disaster.

Wireless communication also affects areas other than just performance. For example, how does a mobile host find a local printer to connect to, rather than use its home printer? Somewhat related to this is how to get the WWW page for the local cell, even if its name is not known. Also, WWW page designers tend to assume lots of bandwidth is available. Putting a large logo on every page becomes counterproductive if it is going to take 30 sec to transmit at 9600 bps every time the page is referenced, irritating the users no end.

6.5. THE ATM AAL LAYER PROTOCOLS

It is not really clear whether or not ATM has a transport layer. On the one hand, the ATM layer has the functionality of a network layer, and there is another layer on top of it (AAL), which sort of makes AAL a transport layer. Some experts agree with this view (e.g., De Prycker, 1993, page 112). One of the protocols used here (AAL 5) is functionally similar to UDP, which is unquestionably a transport protocol.

On the other hand, none of the AAL protocols provide a reliable end-to-end connection, as TCP does (although with only very minor changes they could). Also, in most applications another transport layer is used on top of AAL. Rather than split hairs, we will discuss the AAL layer and its protocols in this chapter without making a claim that it is a true transport layer.

The AAL layer in ATM networks is radically different than TCP, largely because the designers were primarily interested in transmitting voice and video streams, in which rapid delivery is more important than accurate delivery. Remember that the ATM layer just outputs 53-byte cells one after another. It has no error control, no flow control, and no other control. Consequently, it is not well matched to the requirements that most applications need.

To bridge this gap, in Recommendation I.363, ITU has defined an end-to-end layer on top of the ATM layer. This layer, called **AAL** (**ATM Adaptation Layer**) has a tortuous history, full of mistakes, revisions, and unfinished business. In the following sections we will look at it and its design.

The goal of AAL is to provide useful services to application programs and to shield them from the mechanics of chopping data up into cells at the source and reassembling them at the destination. When ITU began defining AAL, it realized that different applications had different requirements, so it organized the service space along three axes:

1. Real-time service versus nonreal-time service.
2. Constant bit rate service versus variable bit rate service.
3. Connection-oriented service versus connectionless service.

In principle, with three axes and two values on each axis, eight distinct services can be defined, as shown in Fig. 6-36. ITU felt that only four of these were of any use, and named them classes A, B, C, and D, as noted. The others were not supported. Starting with ATM 4.0, Fig. 6-36 is somewhat obsolete, so it has been presented here mostly as background information to help understand why the