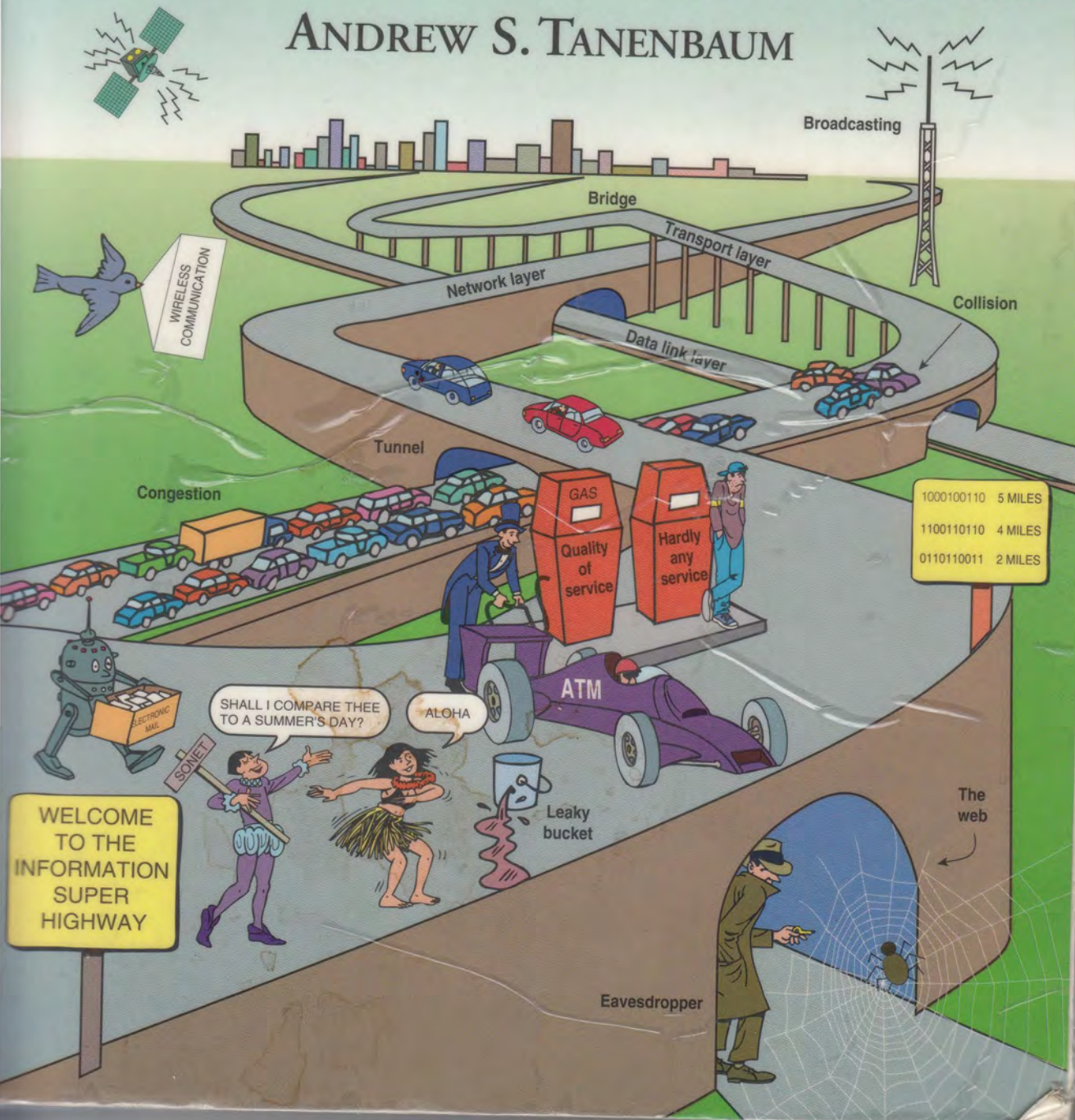


INTERNATIONAL EDITION

THIRD EDITION

# COMPUTER NETWORKS

ANDREW S. TANENBAUM



# Computer Networks

Third Edition

Andrew S. Tanenbaum

*Vrije Universiteit  
Amsterdam, The Netherlands*

*For book and bookstore information*



<http://www.prenhall.com>



*Prentice-Hall International, Inc.*

ISBN 0-13-394248-1

This edition may be sold only in those countries to which it is consigned by Prentice-Hall International. It is not to be re-exported, and it is not for sale in the U.S.A., Mexico, or Canada.



© 1996 by Prentice-Hall, Inc.  
A Simon & Schuster Company  
Upper Saddle River, New Jersey 07458

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

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

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

ISBN 0-13-394248-1

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*  
Prentice-Hall, Inc., *Upper Saddle River, New Jersey*

# 1

## INTRODUCTION

Each of the past three centuries has been dominated by a single technology. The 18th Century was the time of the great mechanical systems accompanying the Industrial Revolution. The 19th Century was the age of the steam engine. During the 20th Century, the key technology has been information gathering, processing, and distribution. Among other developments, we have seen the installation of worldwide telephone networks, the invention of radio and television, the birth and unprecedented growth of the computer industry, and the launching of communication satellites.

Due to rapid technological progress, these areas are rapidly converging, and the differences between collecting, transporting, storing, and processing information are quickly disappearing. Organizations with hundreds of offices spread over a wide geographical area routinely expect to be able to examine the current status of even their most remote outpost at the push of a button. As our ability to gather, process, and distribute information grows, the demand for even more sophisticated information processing grows even faster.

Although the computer industry is young compared to other industries (e.g., automobiles and air transportation), computers have made spectacular progress in a short time. During the first two decades of their existence, computer systems were highly centralized, usually within a single large room. Not infrequently, this room had glass walls, through which visitors could gawk at the great electronic wonder inside. A medium-size company or university might have had one or two

computers, while large institutions had at most a few dozen. The idea that within 20 years equally powerful computers smaller than postage stamps would be mass produced by the millions was pure science fiction.

The merging of computers and communications has had a profound influence on the way computer systems are organized. The concept of the "computer center" as a room with a large computer to which users bring their work for processing is now totally obsolete. The old model of a single computer serving all of the organization's computational needs has been replaced by one in which a large number of separate but interconnected computers do the job. These systems are called **computer networks**. The design and organization of these networks are the subjects of this book.

Throughout the book we will use the term "computer network" to mean an *interconnected* collection of *autonomous* computers. Two computers are said to be interconnected if they are able to exchange information. The connection need not be via a copper wire; fiber optics, microwaves, and communication satellites can also be used. By requiring the computers to be autonomous, we wish to exclude from our definition systems in which there is a clear master/slave relation. If one computer can forcibly start, stop, or control another one, the computers are not autonomous. A system with one control unit and many slaves is not a network; nor is a large computer with remote printers and terminals.

There is considerable confusion in the literature between a computer network and a **distributed system**. The key distinction is that in a distributed system, the existence of multiple autonomous computers is transparent (i.e., not visible) to the user. He<sup>†</sup> can type a command to run a program, and it runs. It is up to the operating system to select the best processor, find and transport all the input files to that processor, and put the results in the appropriate place.

In other words, the user of a distributed system is not aware that there are multiple processors; it looks like a virtual uniprocessor. Allocation of jobs to processors and files to disks, movement of files between where they are stored and where they are needed, and all other system functions must be automatic.

With a network, users must *explicitly* log onto one machine, *explicitly* submit jobs remotely, *explicitly* move files around and generally handle all the network management personally. With a distributed system, nothing has to be done explicitly; it is all automatically done by the system without the users' knowledge.

In effect, a distributed system is a software system built on top of a network. The software gives it a high degree of cohesiveness and transparency. Thus the distinction between a network and a distributed system lies with the software (especially the operating system), rather than with the hardware.

Nevertheless, there is considerable overlap between the two subjects. For example, both distributed systems and computer networks need to move files around. The difference lies in who invokes the movement, the system or the user.

<sup>†</sup> "He" should be read as "he or she" throughout this book.

Although  
important  
see (Coul

## 1.1. USE

Before  
some time  
what they

### 1.1.1. Net

Many  
often locat  
computer a  
do the loca  
tion from t  
them to be

Put in  
goal is to  
on the net  
user. In ot  
his data sh  
This goal n  
of geograph

A sec  
supply. Fo  
one of the  
used. In ad  
others may  
military, b  
cations, the  
utmost imp

Anothe  
price/perfor  
roughly a  
times more  
consisting  
shared file  
whole arra

In the  
request me

Although this book primarily focuses on networks, many of the topics are also important in distributed systems. For more information about distributed systems, see (Coulouris et al., 1994; Mullender, 1993; and Tanenbaum, 1995).

## 1.1. USES OF COMPUTER NETWORKS

Before we start to examine the technical issues in detail, it is worth devoting some time to pointing out why people are interested in computer networks and what they can be used for.

### 1.1.1. Networks for Companies

Many organizations have a substantial number of computers in operation, often located far apart. For example, a company with many factories may have a computer at each location to keep track of inventories, monitor productivity, and do the local payroll. Initially, each of these computers may have worked in isolation from the others, but at some point, management may have decided to connect them to be able to extract and correlate information about the entire company.

Put in slightly more general form, the issue here is **resource sharing**, and the goal is to make all programs, equipment, and especially data available to anyone on the network without regard to the physical location of the resource and the user. In other words, the mere fact that a user happens to be 1000 km away from his data should not prevent him from using the data as though they were local. This goal may be summarized by saying that it is an attempt to end the "tyranny of geography."

A second goal is to provide **high reliability** by having alternative sources of supply. For example, all files could be replicated on two or three machines, so if one of them is unavailable (due to a hardware failure), the other copies could be used. In addition, the presence of multiple CPUs means that if one goes down, the others may be able to take over its work, although at reduced performance. For military, banking, air traffic control, nuclear reactor safety, and many other applications, the ability to continue operating in the face of hardware problems is of utmost importance.

Another goal is **saving money**. Small computers have a much better price/performance ratio than large ones. Mainframes (room-size computers) are roughly a factor of ten faster than personal computers, but they cost a thousand times more. This imbalance has caused many systems designers to build systems consisting of personal computers, one per user, with data kept on one or more shared **file server** machines. In this model, the users are called **clients**, and the whole arrangement is called the **client-server model**. It is illustrated in Fig. 1-1.

In the client-server model, communication generally takes the form of a request message from the client to the server asking for some work to be done.

headers are stripped off one by one as the message propagates up the layers until it finally arrives at the receiving process.

The key idea throughout is that although actual data transmission is vertical in Fig. 1-17, each layer is programmed as though it were horizontal. When the sending transport layer, for example, gets a message from the session layer, it attaches a transport header and sends it to the receiving transport layer. From its point of view, the fact that it must actually hand the message to the network layer on its own machine is an unimportant technicality. As an analogy, when a Tagalog-speaking diplomat is addressing the United Nations, he thinks of himself as addressing the other assembled diplomats. That, in fact, he is really only speaking to his translator is seen as a technical detail.

#### 1.4.2. The TCP/IP Reference Model

Let us now turn from the OSI reference model to the reference model used in the grandparent of all computer networks, the ARPANET, and its successor, the worldwide Internet. Although we will give a brief history of the ARPANET later, it is useful to mention a few key aspects of it now. The ARPANET was a research network sponsored by the DoD (U.S. Department of Defense). It eventually connected hundreds of universities and government installations using leased telephone lines. When satellite and radio networks were added later, the existing protocols had trouble interworking with them, so a new reference architecture was needed. Thus the ability to connect multiple networks together in a seamless way was one of the major design goals from the very beginning. This architecture later became known as the **TCP/IP Reference Model**, after its two primary protocols. It was first defined in (Cerf and Kahn, 1974). A later perspective is given in (Leiner et al., 1985). The design philosophy behind the model is discussed in (Clark, 1988).

Given the DoD's worry that some of its precious hosts, routers, and internetwork gateways might get blown to pieces at a moment's notice, another major goal was that the network be able to survive loss of subnet hardware, with existing conversations not being broken off. In other words, DoD wanted connections to remain intact as long as the source and destination machines were functioning, even if some of the machines or transmission lines in between were suddenly put out of operation. Furthermore, a flexible architecture was needed, since applications with divergent requirements were envisioned, ranging from transferring files to real-time speech transmission.

#### The Internet Layer

All these requirements led to the choice of a packet-switching network based on a connectionless internetwork layer. This layer, called the **internet layer**, is the linchpin that holds the whole architecture together. Its job is to permit hosts to

inject packets into any network and have them travel independently to the destination (potentially on a different network). They may even arrive in a different order than they were sent, in which case it is the job of higher layers to rearrange them, if in-order delivery is desired. Note that "internet" is used here in a generic sense, even though this layer is present in the Internet.

The analogy here is with the (snail) mail system. A person can drop a sequence of international letters into a mail box in one country, and with a little luck, most of them will be delivered to the correct address in the destination country. Probably the letters will travel through one or more international mail gateways along the way, but this is transparent to the users. Furthermore, that each country (i.e., each network) has its own stamps, preferred envelope sizes, and delivery rules is hidden from the users.

The internet layer defines an official packet format and protocol called **IP (Internet Protocol)**. The job of the internet layer is to deliver IP packets where they are supposed to go. Packet routing is clearly the major issue here, as is avoiding congestion. For these reasons, it is reasonable to say that the TCP/IP internet layer is very similar in functionality to the OSI network layer. Figure 1-18 shows this correspondence.

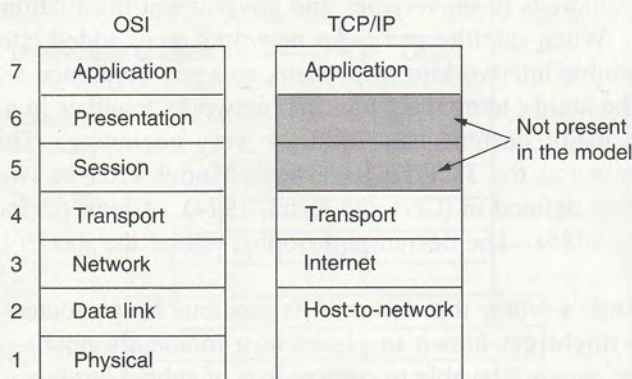


Fig. 1-18. The TCP/IP reference model.

### The Transport Layer

The layer above the internet layer in the TCP/IP model is now usually called the **transport layer**. It is designed to allow peer entities on the source and destination hosts to carry on a conversation, the same as in the OSI transport layer. Two end-to-end protocols have been defined here. The first one, **TCP (Transmission Control Protocol)** is a reliable connection-oriented protocol that allows a byte stream originating on one machine to be delivered without error on

any other machine in the internet. It fragments the incoming byte stream into discrete messages and passes each one onto the internet layer. At the destination, the receiving TCP process reassembles the received messages into the output stream. TCP also handles flow control to make sure a fast sender cannot swamp a slow receiver with more messages than it can handle.

The second protocol in this layer, **UDP (User Datagram Protocol)**, is an unreliable, connectionless protocol for applications that do not want TCP's sequencing or flow control and wish to provide their own. It is also widely used for one-shot, client-server type request-reply queries and applications in which prompt delivery is more important than accurate delivery, such as transmitting speech or video. The relation of IP, TCP, and UDP is shown in Fig. 1-19. Since the model was developed, IP has been implemented on many other networks.

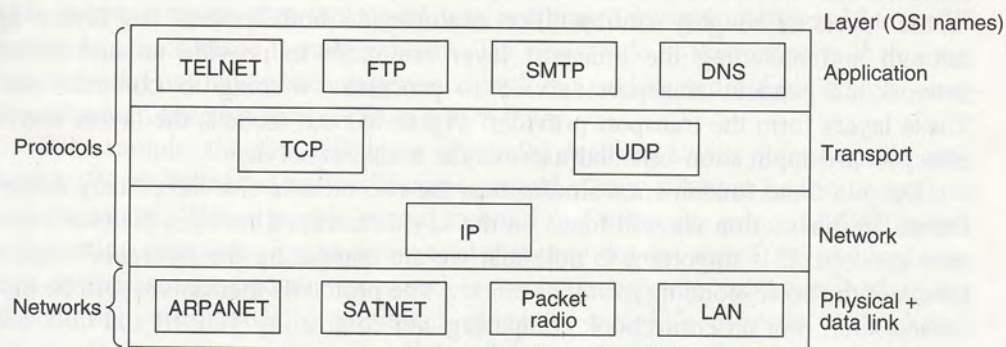


Fig. 1-19. Protocols and networks in the TCP/IP model initially.

### The Application Layer

The TCP/IP model does not have session or presentation layers. No need for them was perceived, so they were not included. Experience with the OSI model has proven this view correct: they are of little use to most applications.

On top of the transport layer is the **application layer**. It contains all the higher-level protocols. The early ones included virtual terminal (TELNET), file transfer (FTP), and electronic mail (SMTP), as shown in Fig. 1-19. The virtual terminal protocol allows a user on one machine to log into a distant machine and work there. The file transfer protocol provides a way to move data efficiently from one machine to another. Electronic mail was originally just a kind of file transfer, but later a specialized protocol was developed for it. Many other protocols have been added to these over the years, such as the Domain Name Service (DNS) for mapping host names onto their network addresses, NNTP, the protocol used for moving news articles around, and HTTP, the protocol used for fetching pages on the World Wide Wide, and many others.

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.

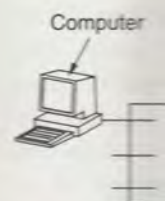
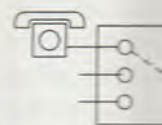
#### Circuit Switching

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  
that either he  
ment or he w  
100 years, t  
Strowger gea  
board operato  
What is the

The mod  
of the “copp  
links onto wh  
is valid: once  
will continue



An impor  
path *before* an  
the start of ri  
calls. During  
as shown in F  
call request

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.

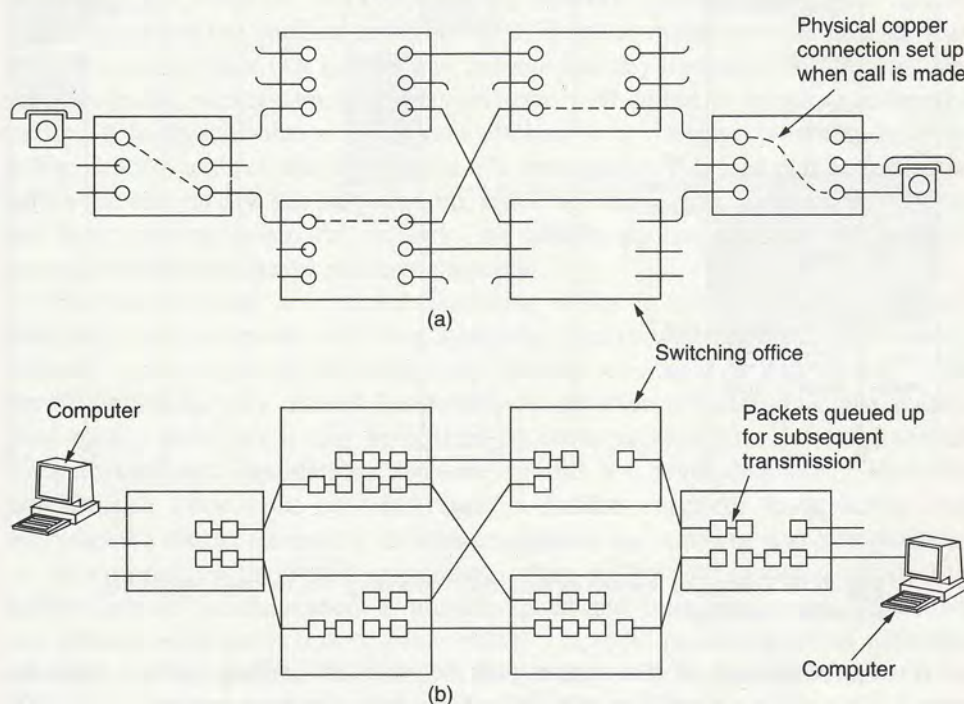
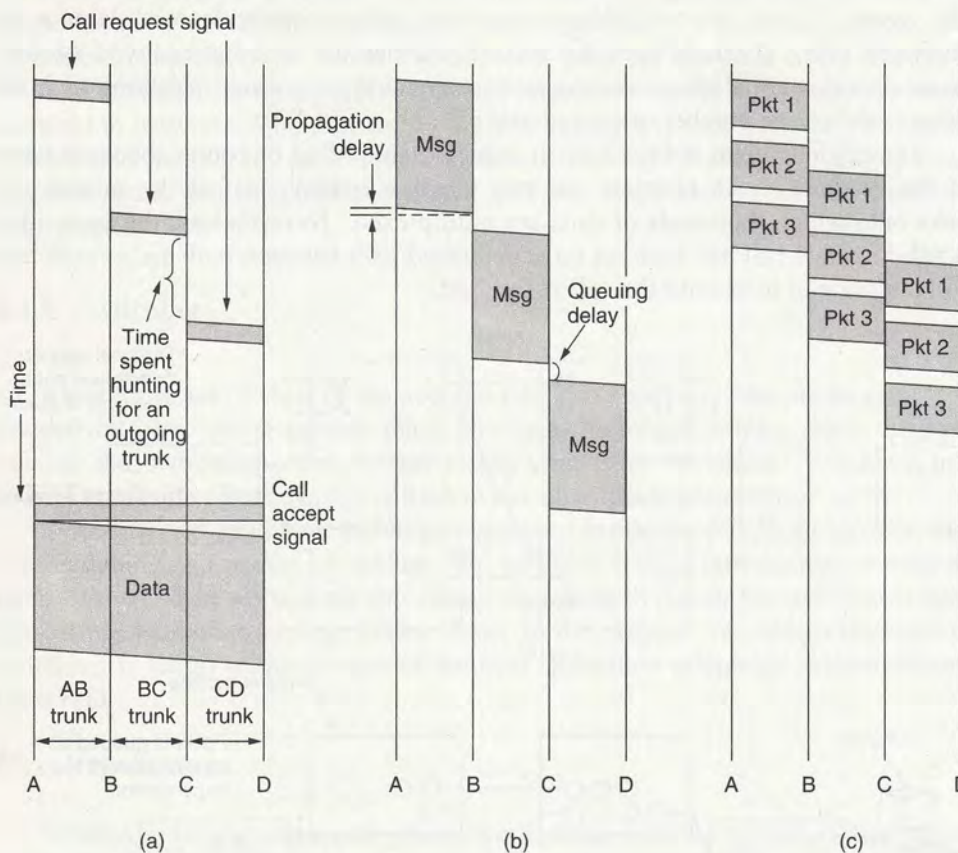


Fig. 2-34. (a) Circuit switching. (b) Packet switching.

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 to 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

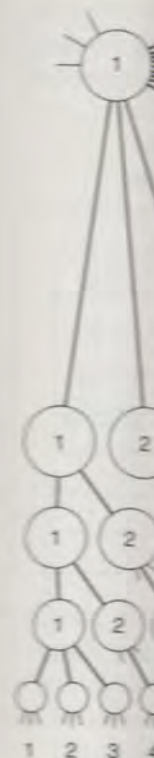


Fig.

dashed lines. The actual trunks along possible because purpose com

### Crossbar Sw

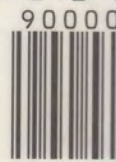
Let us ne switches acti within the t called a croe and  $n$  outpo

These are unabridged paperback reprints of established titles widely used by universities and colleges throughout the world.

Prentice Hall International publishes these lower-priced editions for the benefit of students.

This edition may be sold only in those countries to which it is consigned by Prentice Hall International. It is not to be re-exported, and is not for sale in the U.S.A., Mexico, or Canada.

ISBN 0-13-394248-1



9 780133 942484

Prentice Hall International Editions