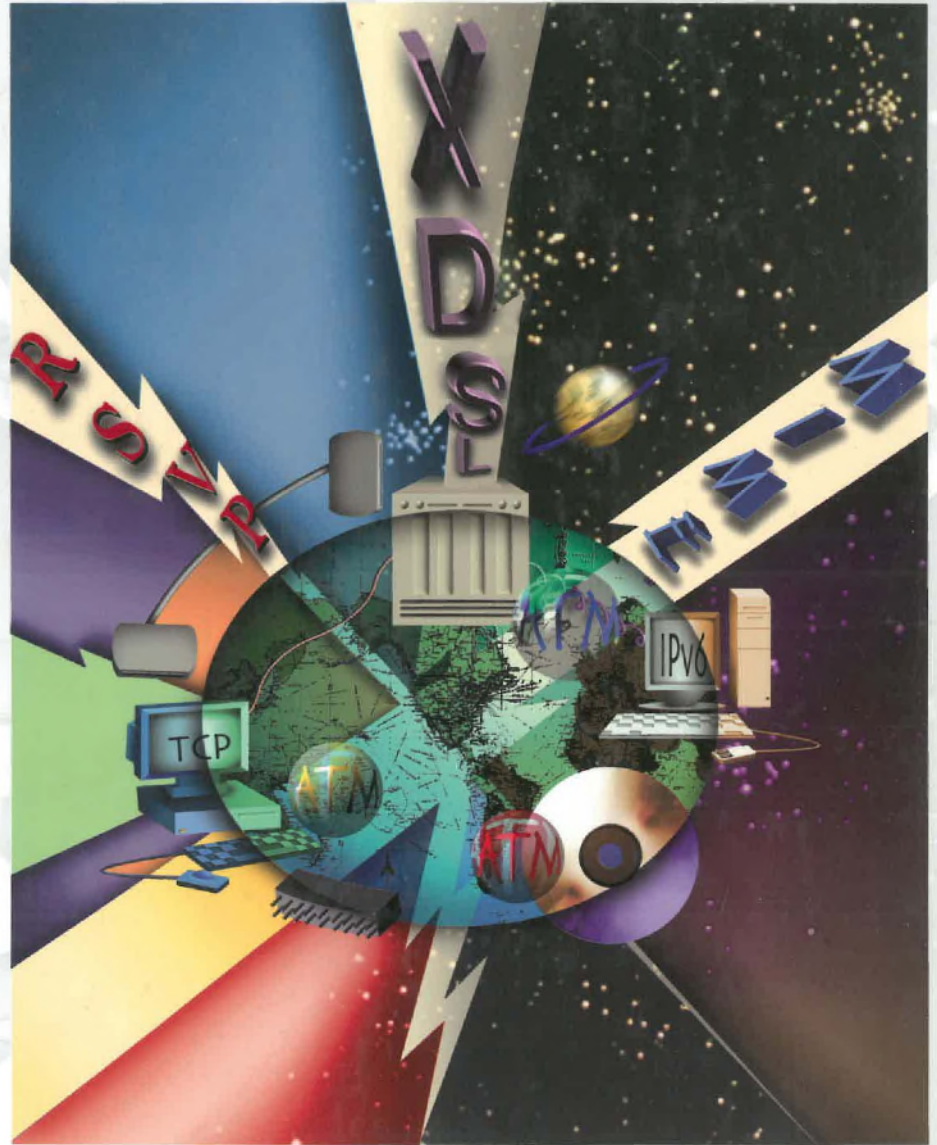


DATA & COMPUTER COMMUNICATIONS

SIXTH EDITION



WILLIAM STALLINGS

DATA AND COMPUTER
COMMUNICATIONS

SIXTH EDITION

William Stallings



Prentice Hall
Upper Saddle River, New Jersey 07458

Library of Congress Cataloging-in-Publication Data
Stallings, William.

Data and computer communications / William Stallings.—6th ed.
p. cm.

1. Bibliography—References and index.

ISBN 0-13-084370-9

1. Title; 2. Computer networks.

TK5105.S73 1999

004.6—dc20

99-24419

CIP

Publisher: Alan Apt
Editor-in-chief: Marcia Horton
Project manager: Ann Ayris Terry
Editorial/production supervision: Rose Kernan
Executive managing editor: Vance O'Brien
Copy editing: Patricia Daly
Art director and cover designer: Heather Scott
Assistant Vice president of production: David W. Riccardi
Manufacturing buyer: Pat Brown
Editorial assistant: Toni Holm

© 2000, 1998 by Prentice-Hall, Inc.
Upper Saddle River, New Jersey 07458

The author and publisher of this book have used their best efforts in preparing this book. These efforts include the development, research, and testing of the theories and programs to determine their effectiveness. The author and publisher make no warranty of any kind, expressed or implied, with regard to these programs or the documentation contained in this book. The author and publisher shall not be liable in any event for incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of these programs.

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.

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-084370-9

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
Prentice-Hall (Singapore) Pte. Ltd., Singapore
Editora Prentice-Hall do Brasil, Ltda., Rio de Janeiro

For my great-hearted wife
ATS
and her constant companions Geoffroi
and Princesse Kate Lan Knecht
and Duchess of Albany, Paroon Schuyler,
Les Enfants du Paradis

CHAPTER 1

INTRODUCTION

1.1 A Communications Model

1.2 Data Communications

1.3 Data Communication Networking

Wide Area Networks

Local Area Networks

1.4 Protocols and Protocol Architecture

A Three-Layer Model

The TCP/IP Protocol Architecture

The OSI Model

1.5 Standards

Appendix 1A Standards Organizations

Internet Standards and the Internet Society

The International Organization for Standardization

ITU Telecommunication Standardization Sector

ATM Forum

Appendix 1B Internet Resources

Web Sites for This Book

Other Web Sites

USENET Newsgroups

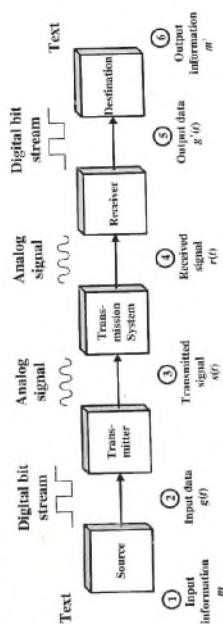


Figure 1.2 Simplified Data Communications Model

To get some flavor for the focus of Part Two, Figure 1.2 provides a new perspective on the communications model of Figure 1.1a. We trace the details of this figure using electronic mail as an example.

Suppose that the input device and transmitter are components of a personal computer. The user of the PC wishes to send a message m to another user. The user activates the electronic mail package on the PC and enters the message via the keyboard (input device). The character string is briefly buffered in main memory. We can view it as a sequence of bits (g) in memory. The personal computer is connected to some transmission medium, such as a local network or a telephone line, by an I/O device (transmitter), such as a local network transceiver or a modem. The input data are transferred to the transmitter as a sequence of voltage shifts $g(t)$ representing bits on some communications bus or cable. The transmitter is connected directly to the medium and converts the incoming stream $g(t)$ into a signal $s(t)$ suitable for transmission; specific alternatives will be described in Chapter 5.

The transmitted signal $s(t)$ presented to the medium is subject to a number of impairments, discussed in Chapter 3, before it reaches the receiver. Thus, the received signal $r(t)$ may differ from $s(t)$. The receiver will attempt to estimate the original $s(t)$, based on $r(t)$ and its knowledge of the medium, producing a sequence of bits $g'(t)$. These bits are sent to the output personal computer, where they are briefly buffered in memory as a block of bits (g'). In many cases, the destination system will attempt to determine if an error has occurred and, if so, cooperate with the source system to eventually obtain a complete, error-free block of data. These data are then presented to the user via an output device, such as a printer or screen. The message (m') as viewed by the user will usually be an exact copy of the original message (m).

Now consider a telephone conversation. In this case the input to the telephone is a message (m) in the form of sound waves. The sound waves are converted by the telephone into electrical signals of the same frequency. These signals are transmitted without modification over the telephone line. Hence the input signal $g(t)$ and the transmitted signal $s(t)$ are identical. The signal $s(t)$ will suffer some distortion over the medium, so that $r(t)$ will not be identical to $s(t)$. Nevertheless, the signal $r(t)$ is converted back into a sound wave with no attempt at correction or improvement of signal quality. Thus, m' is not an exact replica of m . However, the received sound message is generally comprehensible to the listener.

The discussion so far does not touch on other key aspects of data communications, including data link control techniques for controlling the flow of data and detecting and correcting errors, and multiplexing techniques for transmission efficiency. All of these topics are explored in Part Two.

1.3 DATA COMMUNICATION NETWORKING

It is often impractical for two communicating devices to be directly, point-to-point connected. This is so for one (or both) of the following contingencies:

- The devices are very far apart. It would be inordinately expensive, for example, to string a dedicated link between two devices thousands of kilometers apart.
- There is a set of devices, each of which may require a link to many of the others at various times. Examples are all of the telephones in the world and all of the terminals and computers owned by a single organization. Except for the case of a very few devices, it is impractical to provide a dedicated wire between each pair of devices.

The solution to this problem is to attach each device to a communication network. Figure 1.3 relates this discussion to the communications model of Figure 1.1a and also suggests the two major categories into which communications networks are traditionally classified: wide area networks (WANs) and local area networks (LANs). The distinction between the two, both in terms of technology and application, has become somewhat blurred in recent years, but it remains a useful way of organizing the discussion.

Wide Area Networks

Wide area networks generally cover a large geographical area, require the crossing of public right-of-ways, and rely at least in part on circuits provided by a common carrier. Typically, a WAN consists of a number of interconnected switching nodes. A transmission from any one device is routed through these internal nodes to the specified destination device. These nodes (including the boundary nodes) are not concerned with the content of the data; rather, their purpose is to provide a switching facility that will move the data from node to node until they reach their destination.

Traditionally, WANs have implemented using one of two technologies: circuit switching and packet switching. More recently, frame relay and ATM networks have assumed major roles.

Circuit Switching

In a circuit-switching network, a dedicated communications path is established between two stations through the nodes of the network. That path is a connected sequence of physical links between nodes. On each link, a logical channel is dedicated to the connection. Data generated by the source station are transmitted along the dedicated path as rapidly as possible. At each node, incoming data are routed

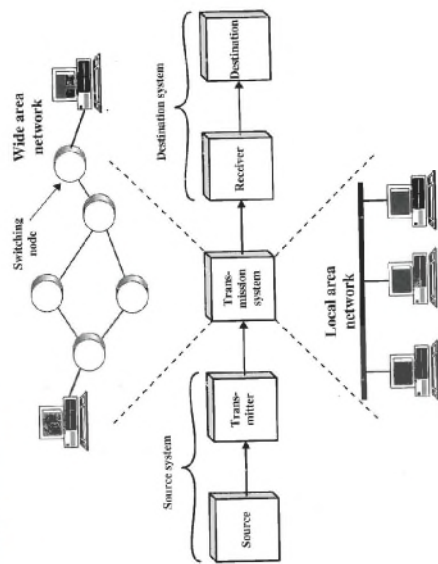


Figure 1.3 Simplified Network Models

or switched to the appropriate outgoing channel without delay. The most common example of circuit switching is the telephone network.

Packet Switching

A quite different approach is used in a packet-switching network. In this case, it is not necessary to dedicate transmission capacity along a path through the network. Rather, data are sent out in a sequence of small chunks, called packets. Each packet is passed through the network from node to node along some path leading from source to destination. At each node, the entire packet is received, stored briefly, and then transmitted to the next node. Packet-switching networks are commonly used for terminal-to-computer and computer-to-computer communications.

Frame Relay

Packet switching was developed at a time when digital long-distance transmission facilities exhibited a relatively high error rate compared to today's facilities. As a result, there is a considerable amount of overhead built into packet-switching schemes to compensate for errors. The overhead includes additional bits added to each packet to introduce redundancy and additional processing at the end stations and the intermediate switching nodes to detect and recover from errors.

With modern high-speed telecommunications systems, this overhead is unnecessary and counterproductive. It is unnecessary because the rate of errors has been dramatically lowered and any remaining errors can easily be caught in the end systems by logic that operates above the level of the packet-switching logic. It is counterproductive because the overhead involved soaks up a significant fraction of the high capacity provided by the network.

Frame relay was developed to take advantage of these high data rates and low error rates. Whereas the original packet-switching networks were designed with a data rate to the end user of about 64 kbps, frame relay networks are designed to operate efficiently at user data rates of up to 2 Mbps. The key to achieving these high data rates is to strip out most of the overhead involved with error control.

ATM

Asynchronous transfer mode (ATM), sometimes referred to as cell relay, is a culmination of all of the developments in circuit switching and packet switching over the past 25 years.

ATM can be viewed as an evolution from frame relay. The most obvious difference between frame relay and ATM is that frame relay uses variable-length packets, called frames, and ATM uses fixed-length packets, called cells. As with frame relay, ATM provides little overhead for error control, depending on the inherent reliability of the transmission system and on higher layers of logic in the end systems to catch and correct errors. By using a fixed packet length, the processing overhead is reduced even further for ATM compared to frame relay. The result is that ATM is designed to work in the range of 10s and 100s of Mbps, and in the Gbps range.

ATM can also be viewed as an evolution from circuit switching. With circuit switching, only fixed-data-rate circuits are available to the end system. ATM allows the definition of multiple virtual channels with data rates that are dynamically defined at the time the virtual channel is created. By using small, fixed-size cells, ATM is so efficient that it can offer a constant-data-rate channel even though it is using a packet switching technique. Thus, ATM extends circuit switching to allow multiple channels with the data rate on each channel dynamically set on demand.

ISDN and Broadband ISDN

Merging and evolving communications and computing technologies, coupled with increasing demands for efficient and timely collection, processing, and dissemination of information, are leading to the development of integrated systems that transmit and process all types of data. A significant outgrowth of these trends is the integrated services digital network (ISDN).

The ISDN is designed to replace existing public telecommunications networks and deliver a wide variety of services. The ISDN is defined by the standardization of user interfaces and implemented as a set of digital switches and paths supporting a broad range of traffic types and providing value-added processing services. In practice, there are multiple networks, implemented within national boundaries, but from the user's point of view, there is a single, uniformly accessible, worldwide network.

Despite the fact that ISDN has yet to achieve the universal deployment hoped for, it is already in its second generation. The first generation, sometimes referred

CHAPTER 8

MULTIPLEXING

- 8.1 Frequency-Division Multiplexing
 - Characteristics
 - Analog Carrier Systems
- 8.2 Synchronous Time-Division Multiplexing
 - Characteristics
 - TDM Link Control
 - Digital Carrier Systems
 - ISDN User-Network Interface
 - SONET/SDH
- 8.3 Statistical Time-Division Multiplexing
 - Characteristics
 - Performance
- 8.4 Asymmetric Digital Subscriber Line
 - ADSL Design
 - Discrete Multitone
- 8.5 xDSL
 - High Data Rate Digital Subscriber Line
 - Single Line Digital Subscriber Line
 - Very High Data Rate Digital Subscriber Line
- 8.6 Recommended Reading and Web Sites
- 8.7 Problems

- ◆ To make efficient use of high-speed telecommunications lines, some form of multiplexing is used. Multiplexing allows several transmission sources to share a larger transmission capacity. The two common forms of multiplexing are frequency-division multiplexing (FDM) and time-division multiplexing (TDM).
- ◆ Frequency-division multiplexing can be used with analog signals. A number of signals are carried simultaneously on the same medium by allocating to each signal a different frequency band. Modulation equipment is needed to move each signal to the required frequency band, and multiplexing equipment is needed to combine the modulated signals.
- ◆ Synchronous time-division multiplexing can be used with digital signals or analog signals carrying digital data. In this form of multiplexing, data from various sources are carried in repetitive frames. Each frame consists of a set of time slots, and each source is assigned one or more time slots per frame. The effect is to interleave bits of data from the various sources.
- ◆ Statistical time-division multiplexing provides a generally more efficient service than synchronous TDM for the support of terminals. With statistical TDM, time slots are not pre-assigned to particular data sources. Rather, user data are buffered and transmitted as rapidly as possible using available time slots.

In Chapter 7, we described efficient techniques for utilizing a data link under heavy load. Specifically, with two devices connected by a point-to-point link, it is generally desirable to have multiple frames outstanding so that the data link does not become a bottleneck between the stations. Now consider the opposite problem. Typically, two communicating stations will not utilize the full capacity of a data link. For efficiency, it should be possible to share that capacity. A generic term for such sharing is *multiplexing*.

A common application of multiplexing is in long-haul communications. Trunks on long-haul networks are high-capacity fiber, coaxial, or microwave links. These links can carry large numbers of voice and data transmissions simultaneously using multiplexing.

Figure 8.1 depicts the multiplexing function in its simplest form. There are n inputs to a multiplexer. The multiplexer is connected by a single data link to a

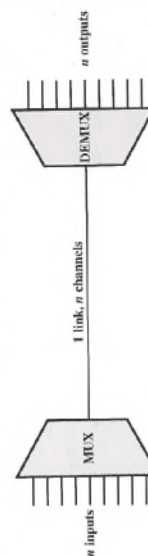


Figure 8.1 Multiplexing

demultiplexer. The link is able to carry n separate channels of data. The multiplexer combines (multiplexes) data from the n input lines and transmits over a higher-capacity data link. The demultiplexer accepts the multiplexed data stream, separates (demultiplexes) the data according to channel, and delivers them to the appropriate output lines.

The widespread use of multiplexing in data communications can be explained by the following:

- The higher the data rate, the more cost-effective the transmission facility. That is, for a given application and over a given distance, the cost per kbps declines with an increase in the data rate of the transmission facility. Similarly, the cost of transmission and receiving equipment, per kbps, declines with increasing data rate.
- Most individual data communicating devices require relatively modest data rate support. For example, for most terminal and personal computer applications, a data rate of between 9600 bps and 64 kbps is generally adequate.

The preceding statements were phrased in terms of data communicating devices. Similar statements apply to voice communications. That is, the greater the capacity of a transmission facility, in terms of voice channels, the less the cost per individual voice channel, and the capacity required for a single voice channel is modest.

This chapter concentrates on three types of multiplexing techniques. The first, frequency-division multiplexing (FDM), is the most heavily used and is familiar to anyone who has ever used a radio or television set. The second is a particular case of time-division multiplexing (TDM) known as synchronous TDM. This is commonly used for multiplexing digitized voice streams and data streams. The third type seeks to improve on the efficiency of synchronous TDM by adding complexity to the multiplexer. It is known by a variety of names, including statistical TDM, asynchronous TDM, and intelligent TDM. This book uses the term *statistical TDM*, which highlights one of its chief properties. Finally, we look at the digital subscriber line, which combines FDM and synchronous TDM technologies.

8.1 FREQUENCY-DIVISION MULTIPLEXING

Characteristics

FDM is possible when the useful bandwidth of the transmission medium exceeds the required bandwidth of signals to be transmitted. A number of signals can be carried simultaneously if each signal is modulated onto a different carrier frequency and the carrier frequencies are sufficiently separated that the bandwidths of the signals do not significantly overlap. A general case of FDM is shown in Figure 8.2a. Six signal sources are fed into a multiplexer, which modulates each signal onto a different frequency ($f_1, \dots, f_6$). Each modulated signal requires a certain bandwidth centered around its carrier frequency, referred to as a *channel*. To prevent interference, the channels are separated by guard bands, which are unused portions of the spectrum.

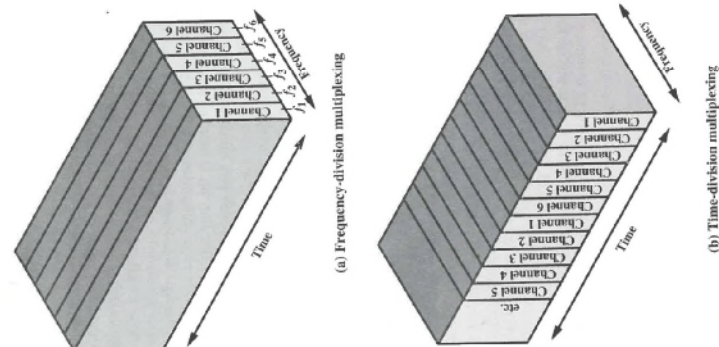


Figure 8.2 FDM and TDM

The composite signal transmitted across the medium is analog. Note, however, that the input signals may be either digital or analog. In the case of digital input, the input signals must be passed through modems to be converted to analog. In either case, each input analog signal must then be modulated to move it to the appropriate frequency band.

A familiar example of FDM is broadcast and cable television. The television signal discussed in Chapter 3 fits comfortably into a 6-MHz bandwidth. Figure 8.3 depicts the transmitted TV signal and its bandwidth. The black-and-white video signal is AM modulated on a carrier signal f_{cv} . Because the base-band video signal has a bandwidth of 4 MHz, we would expect the modulated signal to have a bandwidth of 8 MHz centered on f_{cv} . To conserve bandwidth, the signal is passed through a sideband filter so that most of the lower sideband is suppressed. The resulting signal extends from about $f_{cv} - 0.75$ MHz to $f_{cv} + 4.2$ MHz. A separate color subcarrier, f_{cc} , is used to transmit color information. This is spaced far enough from f_{cv} that there is essentially no interference. Finally, the audio portion of the signal is modulated on f_{ca} , outside the effective bandwidth of the other two signals. A bandwidth of 50

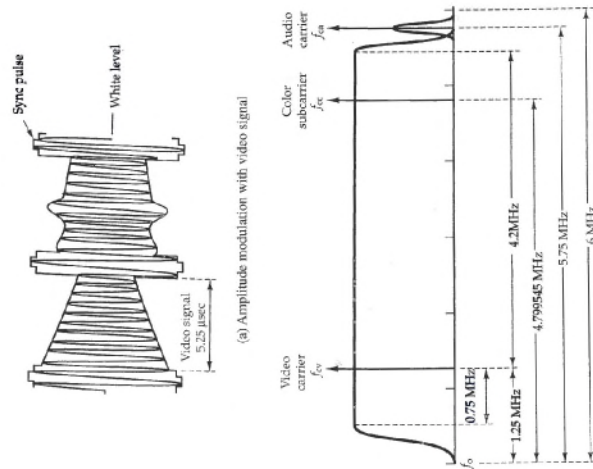


Figure 8.3 Transmitted TV signal

There are several variations to supergroup formation. Each of the five inputs to the supergroup multiplexer may be a group channel containing 12 multiplexed voice signals. In addition, any signal up to 48 kHz wide whose bandwidth is contained within 60 to 108 kHz may be used as input to the supergroup multiplexer. As another variation, it is possible to combine 60 voiceband channels into a supergroup. This may reduce multiplexing costs where an interface with existing group multiplexer is not required.

The next level of the hierarchy is the mastergroup, which combines 10 supergroup inputs. Again, any signal with a bandwidth of 240 kHz in the range 312 to 552 kHz can serve as input to the mastergroup multiplexer. The mastergroup has a bandwidth of 2.52 MHz and can support 600 voice frequency (VF) channels. Higher-level multiplexing is defined above the mastergroup, as shown in Table 8.2.

Note that the original voice or data signal may be modulated many times. For example, a data signal may be encoded using QPSK to form an analog voice signal. This signal could then be used to modulate a 76-kHz carrier to form a component of a group signal. This group signal could then be used to modulate a 516-kHz carrier to form a component of a supergroup signal. Each stage can distort the original data; this is so, for example, if the modulator/multiplexer contains nonlinearities or introduces noise.

8.2 SYNCHRONOUS TIME-DIVISION MULTIPLEXING

Characteristics

Synchronous time-division multiplexing is possible when the achievable data rate (sometimes, unfortunately, called bandwidth) of the medium exceeds the data rate of digital signals to be transmitted. Multiple digital signals (or analog signals carrying digital data) can be carried on a single transmission path by interleaving portions of each signal in time. The interleaving can be at the bit level or in blocks of bytes or larger quantities. For example, the multiplexer in Figure 8.2b has six inputs that might each be, say, 9.6 kbps. A single line with a capacity of at least 57.6 kbps (plus overhead capacity) could accommodate all six sources.

A generic depiction of a synchronous TDM system is provided in Figure 8.6. A number of signals $m_i(t)$, $i = 1, n$ are to be multiplexed onto the same transmission medium. The signals carry digital data and are generally digital signals. The incoming data from each source are briefly buffered. Each buffer is typically one bit or one character in length. The buffers are scanned sequentially to form a composite digital data stream $m_d(t)$. The scan operation is sufficiently rapid so that each buffer is emptied before more data can arrive. Thus, the data rate of $m_d(t)$ must at least equal the sum of the data rates of the $m_i(t)$. The digital signal $m_d(t)$ may be transmitted directly, or passed through a modem so that an analog signal is transmitted. In either case, transmission is typically synchronous.

The transmitted data may have a format something like Figure 8.6b. The data are organized into frames. Each frame contains a cycle of time slots. In each frame, one or more slots is dedicated to each data source. The sequence of slots dedicated to one source, from frame to frame, is called a channel. The slot length equals the transmitter buffer length, typically a bit or a character.

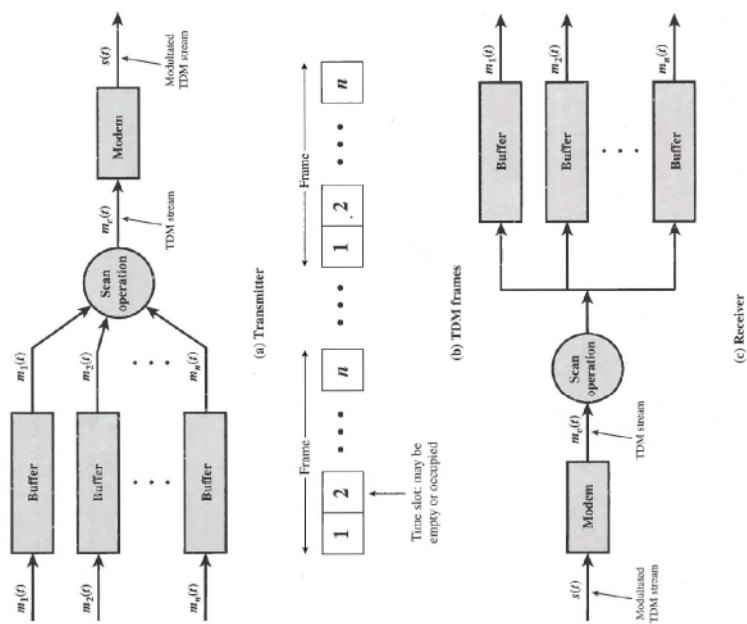


Figure 8.6 Synchronous TDM System

The character-interleaving technique is used with asynchronous sources. Each time slot contains one character of data. Typically, the start and stop bits of each character are eliminated before transmission and reinserted by the receiver, thus improving efficiency. The bit-interleaving technique is used with synchronous sources and may also be used with asynchronous sources. Each time slot contains just one bit.

At the receiver, the interleaved data are demultiplexed and routed to the appropriate destination buffer. For each input source $m(i)$, there is an identical output source that will receive the input data at the same rate at which it was generated.

Synchronous TDM is called synchronous not because synchronous transmission is used, but because the time slots are preassigned to sources and fixed. The time slots for each source are transmitted whether or not the source has data to send. This is, of course, also the case with FDM. In both cases, capacity is wasted to achieve simplicity of implementation. Even when fixed assignment is used, however, it is possible for a synchronous TDM device to handle sources of different data rates. For example, the slowest input device could be assigned one slot per cycle, while faster devices are assigned multiple slots per cycle.

TDM Link Control

The reader will note that the transmitted data stream depicted in Figure 8.6 does not contain the headers and trailers that we have come to associate with synchronous transmission. The reason is that the control mechanisms provided by a data link protocol are not needed. It is instructive to ponder this point, and we do so by considering two key data link control mechanisms: flow control and error control. It should be clear that, as far as the multiplexer and demultiplexer (Figure 8.1) are concerned, flow control is not needed. The data rate on the multiplexed line is fixed, and the multiplexer and demultiplexer are designed to operate at that rate. But suppose that one of the individual output lines attaches to a device that is temporarily unable to accept data. Should the transmission of TDM frames cease? Clearly not, because the remaining output lines are expecting to receive data at predetermined times. The solution is for the saturated output device to cause the flow of data from the corresponding input device to cease. Thus, for a while, the channel in question will carry empty slots, but the frames as a whole will maintain the same transmission rate.

The reasoning for error control is the same. It would not do to request retransmission of an entire TDM frame because an error occurs on one channel. The devices using the other channels do not want a retransmission nor would they know that a retransmission has been requested by some other device on another channel. Again, the solution is to apply error control on a per-channel basis.

Flow control and error control can be provided on a per-channel basis by using a data link control protocol such as HDLC on a per-channel basis. A simplified example is shown in Figure 8.7. We assume two data sources, each using HDLC. One is transmitting a stream of HDLC frames containing three octets of data each, and the other is transmitting HDLC frames containing four octets of data. For clarity, we assume that character-interleaved multiplexing is used, although bit interleaving is more typical. Notice what is happening. The octets of the HDLC frames from the two sources are shuffled together for transmission over the multiplexed line. The reader may initially be uncomfortable with this diagram, because the HDLC frames have lost their integrity in some sense. For example, each frame check sequence (FCS) on the line applies to a disjointed set of bits. Even the FCS is not in one piece. However, the pieces are reassembled correctly before they are seen by the device on the other end of the HDLC protocol. In this sense, the multiplexing/demultiplexing operation is transparent to the attached stations; to each communicating pair of stations, it appears that they have a dedicated link.

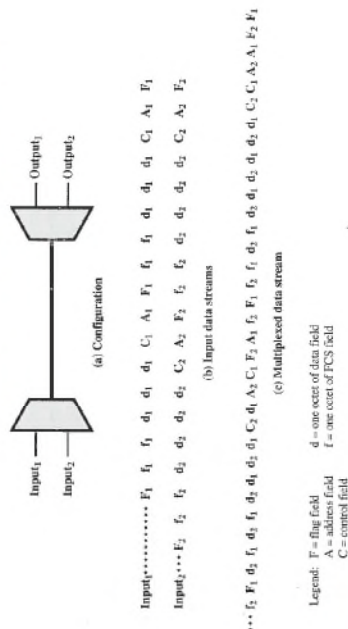


Figure 8.7 Use of Data Link Control on TDM Channels

One refinement is needed in Figure 8.7. Both ends of the line need to be a combination multiplexer/demultiplexer with a full-duplex line in between. Then each channel consists of two sets of slots, one traveling in each direction. The individual devices attached at each end can, in pairs, use HDLC to control their own channel. The multiplexer/demultiplexers need not be concerned with these matters.

Framing

We have seen that a link control protocol is not needed to manage the overall TDM link. There is, however, a basic requirement for framing. Because we are not providing flag or SYNC characters to bracket TDM frames, some means is needed to assure frame synchronization. It is clearly important to maintain framing synchronization because, if the source and destination are out of step, data on all channels are lost.

Perhaps the most common mechanism for framing is known as added-digit framing. In this scheme, typically, one control bit is added to each TDM frame. An identifiable pattern of bits, from frame to frame, is used on this "control channel." A typical example is the alternating bit pattern, 1010101010101010. This is a pattern unlikely to be sustained on a data channel. Thus, to synchronize, a receiver compares the incoming bits of one frame position to the expected pattern. If the pattern does not match, successive bit positions are searched until the pattern persists over multiple frames. Once framing synchronization is established, the receiver continues to monitor the framing bit channel. If the pattern breaks down, the receiver must again enter a framing search mode.

Pulse Stuffing

Perhaps the most difficult problem in the design of a synchronous time-division multiplexer is that of synchronizing the various data sources. If each source

Table 15.3 IP Addresses and Subnet Masks [STEI95]

(a) Dotted decimal and binary representations of IP address and subnet masks

	Binary Representation	Dotted Decimal
IP address	11000000.11100100.00010001.00111001	192.228.17.57
Subnet mask	11111111.11111111.11111111.11100000	255.255.255.224
Bitwise AND of address and mask (resultant network/subnet number)	11000000.11100100.00010001.00100000	192.228.17.32
Subnet number	11000000.11100100.00010001.001	1
Host number	00000000.00000000.00000000.00011001	25

(b) Default subnet masks

	Binary Representation	Dotted Decimal
Class A default mask	11111111.00000000.00000000.00000000	255.0.0.0
Example Class A mask	11111111.11000000.00000000.00000000	255.192.0.0
Class B default mask	11111111.11111111.00000000.00000000	255.255.0.0
Example Class B mask	11111111.11111111.11110000.00000000	255.255.248.0
Class C default mask	11111111.11111111.11111111.00000000	255.255.255.0
Example Class C mask	11111111.11111111.11111111.11111100	255.255.255.252

leftmost three octets are the network number and the rightmost octet contains a host number x . Both routers R1 and R2 are configured with a subnet mask with the value 255.255.255.224 (see Table 15.3a). For example, if a datagram with the destination address 192.228.17.57 arrives at R1 from either the rest of the internet or from LAN Y, R1 applies the subnet mask to determine that this address refers to subnet 1, which is LAN X, and so forwards the datagram to LAN X. Similarly, if a datagram with that destination address arrives at R2 from LAN Z, R2 applies the mask and then determines from its forwarding database that datagrams destined for subnet 1 should be forwarded to R1. Hosts most also employ a subnet mask to make routing decisions.

The default subnet mask for a given class of addresses is a null mask (Table 15.3b), which yields the same network and host number as the non-subnetted address.

Internet Control Message Protocol (ICMP)

The IP standard specifies that a compliant implementation must also implement ICMP (RFC 792). ICMP provides a means for transferring messages from routers and other hosts to a host. In essence, ICMP provides feedback about problems in the communication environment. Examples of its use are as follows: when a datagram cannot reach its destination, when the router does not have the buffering capacity to forward a datagram, and when the router can direct the station to send traffic on a shorter route. In most cases, an ICMP message is sent in response to a datagram, either by a router along the datagram's path or by the intended destination host.

Although ICMP is, in effect, at the same level as IP in the TCP/IP architecture, it is a user of IP. An ICMP message is constructed and then passed down to IP.

which encapsulates the message with an IP header and then transmits the resulting datagram in the usual fashion. Because ICMP messages are transmitted in IP datagrams, their delivery is not guaranteed and their use cannot be considered reliable.

Figure 15.9 shows the format of the various ICMP message types. An ICMP message starts with a 64-bit header consisting of the following:

- **Type (8 bits):** Specifies the type of ICMP message.
- **Code (8 bits):** Used to specify parameters of the message that can be encoded in one or a few bits.
- **Checksum (16 bits):** Checksum of the entire ICMP message. This is the same checksum algorithm used for IP.
- **Parameters (32 bits):** Used to specify more lengthy parameters.

These fields are generally followed by additional information fields that further specify the content of the message.

In those cases in which the ICMP message refers to a prior datagram, the information field includes the entire IP header plus the first 64 bits of the data field of the original datagram. This enables the source host to match the incoming ICMP message with the prior datagram. The reason for including the first 64 bits of the data field is that this will enable the IP module in the host to determine which upper-level protocol or protocols were involved. In particular, the first 64 bits would include a portion of the TCP header or other transport-level header.

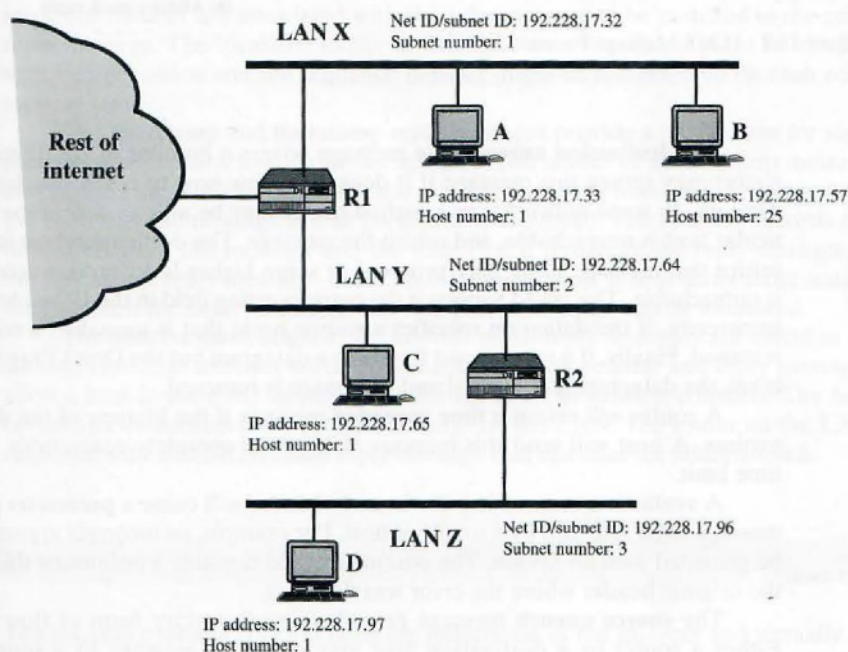


Figure 15.8 Example of Subnetworking

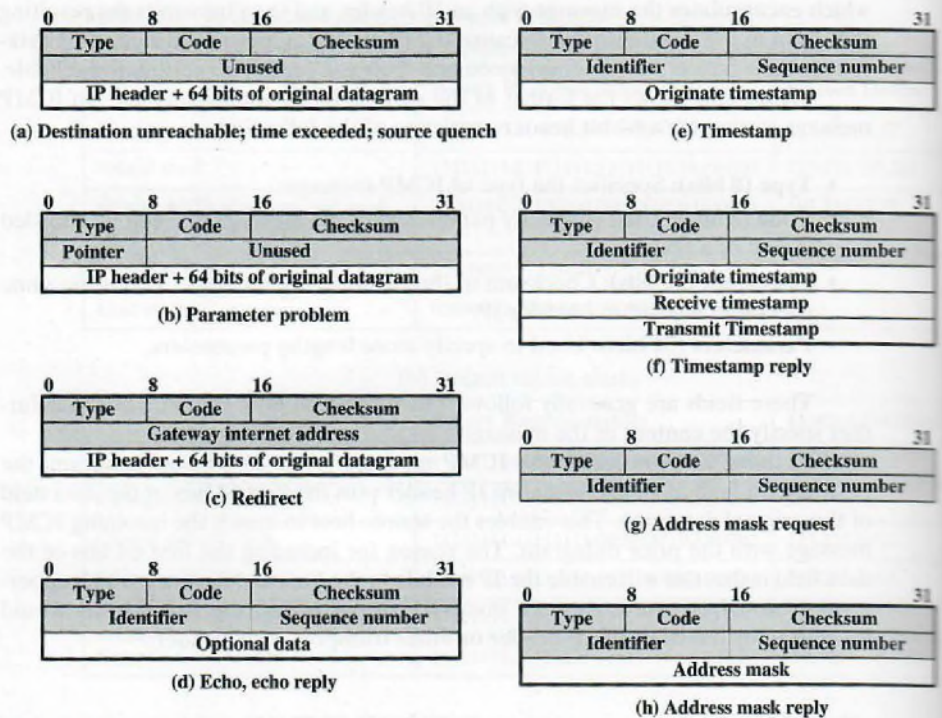


Figure 15.9 ICMP Message Formats

The **destination unreachable** message covers a number of contingencies. A router may return this message if it does not know how to reach the destination network. In some networks, an attached router may be able to determine if a particular host is unreachable, and return the message. The destination host itself may return this message if the user protocol or some higher-level service access point is unreachable. This could happen if the corresponding field in the IP header was set incorrectly. If the datagram specifies a source route that is unusable, a message is returned. Finally, if a router must fragment a datagram but the Don't Fragment flag is set, the datagram is discarded and a message is returned.

A router will return a **time exceeded** message if the lifetime of the datagram expires. A host will send this message if it cannot complete reassembly within a time limit.

A syntactic or semantic error in an IP header will cause a **parameter problem** message to be returned by a router or host. For example, an incorrect argument may be provided with an option. The parameter field contains a pointer to the octet in the original header where the error was detected.

The **source quench** message provides a rudimentary form of flow control. Either a router or a destination host may send this message to a source host.

requesting that it reduce the rate at which it is sending traffic to the internet destination. On receipt of a source quench message, the source host should cut back the rate at which it is sending traffic to the specified destination until it no longer receives source quench messages. The source quench message can be used by a router or host that must discard datagrams because of a full buffer. In that case, the router or host will issue a source quench message for every datagram that it discards. In addition, a system may anticipate congestion and issue source quench messages when its buffers approach capacity. In that case, the datagram referred to in the source quench message may well be delivered. Thus, receipt of a source quench message does not imply delivery or nondelivery of the corresponding datagram.

A router sends a **redirect** message to a host on a directly connected router to advise the host of a better route to a particular destination. The following is an example, using Figure 15.8. Router R1 receives a datagram from host C on network Y, to which R1 is attached. R1 checks its routing table and obtains the address for the next router, R2, on the route to the datagram's internet destination network, Z. Because R2 and the host identified by the internet source address of the datagram are on the same network, R1 sends a redirect message to C. The redirect message advises the host to send its traffic for network Z directly to router R2, because this is a shorter path to the destination. The router forwards the original datagram to its internet destination (via R2). The address of R2 is contained in the parameter field of the redirect message.

The **echo** and **echo reply** messages provide a mechanism for testing that communication is possible between entities. The recipient of an echo message is obligated to return the message in an echo reply message. An identifier and sequence number are associated with the echo message to be matched in the echo reply message. The identifier might be used like a service access point to identify a particular session and the sequence number might be incremented on each echo request sent.

The **timestamp** and **timestamp reply** messages provide a mechanism for sampling the delay characteristics of the internet. The sender of a timestamp message may include an identifier and sequence number in the parameters field and include the time that the message is sent (originate timestamp). The receiver records the time it received the message and the time that it transmits the reply message in the timestamp reply message. If the timestamp message is sent using strict source routing, then the delay characteristics of a particular route can be measured.

The **address mask request** and **address mask reply** messages are useful in an environment that includes subnets. The address mask request and reply messages allow a host to learn the address mask for the LAN to which it connects. The host broadcasts an address mask request message on the LAN. The router on the LAN responds with an address mask reply message that contains the address mask.

15.4 IPV6

The internet protocol (IP) has been the foundation of the Internet and virtually all multivendor private internetworks. This protocol is reaching the end of its useful life