DATA & COMPUTER COMMUNICATIONS

SIXTH EDITION



WILLIAM STALLINGS

Network System Technologies Ex. 2010
Samsung v. Network System Technologies - IPR 2024-00338

Library of Congress Cataloging-in-Publication Data
Stallings, William.
Data and computer communications / William Stallings.—6th ed.
p. cm.
Includes bibliographical references and index.
ISBN 0-13-084370-9
1. Data transmission systems. 2. Computer networks. I. Title.
TK5105.S73 1999 99-24419
004.6—dc20 CIP

Publisher: Alan Apt
Editor-in-chief: Marcia Horton
Project manager: Ana Arias Terry
Editorial/production supervision: Rose Kernan
Executive managing editor: Vince 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, 1996 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.

Reprinted with corrections June, 2000.

Printed in the United States of America

10 9 8 7 6 5 4 3

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*

Network System Technologies Ex. 2010
Samsung v. Network System Technologies - IPR 2024-00338

The next two items might have been included under exchange management, but they seem important enough to list separately. In all communications systems, there is a potential for error; transmitted signals are distorted to some extent before reaching their destination. **Error detection and correction** are required in circumstances where errors cannot be tolerated. This is usually the case with data processing systems. For example, in transferring a file from one computer to another, it is simply not acceptable for the contents of the file to be accidentally altered. **Flow control** is required to assure that the source does not overwhelm the destination by sending data faster than they can be processed and absorbed.

Next are the related but distinct concepts of **addressing** and **routing**. When a transmission facility is shared by more than two devices, a source system must indicate the identity of the intended destination. The transmission system must assure that the destination system, and only that system, receives the data. Further, the transmission system may itself be a network through which various paths may be taken. A specific route through this network must be chosen.

Recovery is a concept distinct from that of error correction. Recovery techniques are needed in situations in which an information exchange, such as a data base transaction or file transfer, is interrupted due to a fault somewhere in the system. The objective is either to be able to resume activity at the point of interruption or at least to restore the state of the systems involved to the condition prior to the beginning of the exchange.

Message formatting has to do with an agreement between two parties as to the form of the data to be exchanged or transmitted, such as the binary code for characters.

Frequently, it is important to provide some measure of **security** in a data communications system. The sender of data may wish to be assured that only the intended receiver actually receives the data. And the receiver of data may wish to be assured that the received data have not been altered in transit and that the data actually come from the purported sender.

Finally, a data communications facility is a complex system that cannot create or run itself. **Network management** capabilities are needed to configure the system, monitor its status, react to failures and overloads, and plan intelligently for future growth.

Thus, we have gone from the simple idea of data communication between source and destination to a rather formidable list of data communications tasks. In this book, we elaborate this list of tasks to describe and encompass the entire set of activities that can be classified under data and computer communications.

1.2 DATA COMMUNICATIONS

Following the first two chapters, this book is organized into four parts. Part Two deals with the must fundamental aspects of the communications function, focusing on the transmission of signals in a reliable and efficient manner. For want of a better name, we have given Part Two the title "Data Communications," although that term arguably encompasses some or even all of the topics of Parts Three through Five.

Network System Technologies Ex. 2010
Samsung v. Network System Technologies - IPR 2024-00338

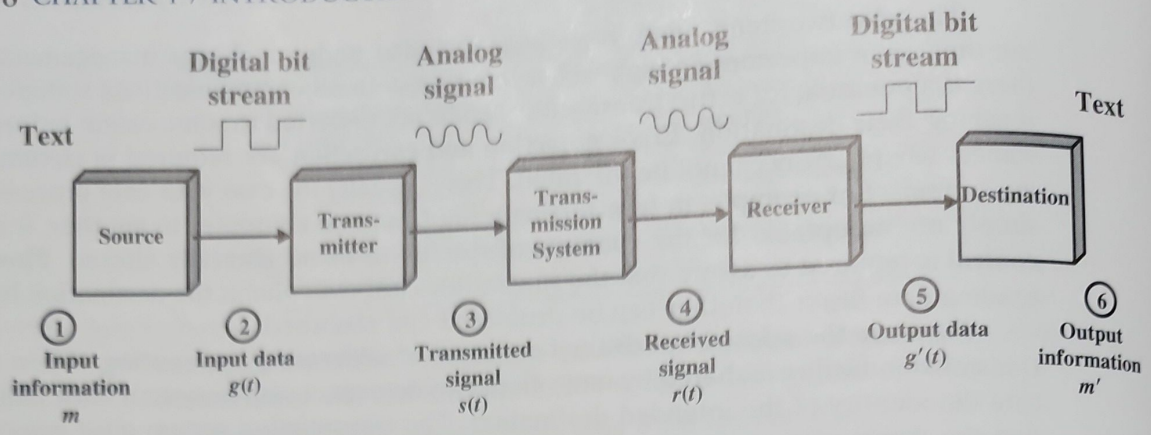


Figure 1.2 Simplified Data Communications Model

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.

Network System Technologies Ex. 2010
Samsung v. Network System Technologies - IPR 2024-00338

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

Network System Technologies Ex. 2010
Samsung v. Network System Technologies - IPR 2024-00338

- **Reduced overhead:** Connection identifiers are generally shorter than global identifiers. For example, in the X.25 protocol (discussed in Chapter 10) used over packet-switched networks, connection request packets contain both source and destination address fields, each with a system-defined length that may be a number of octets. After a logical connection, called a virtual circuit, is established, data packets contain just a 12-bit virtual circuit number.
- **Routing:** In setting up a connection, a fixed route may be defined. The connection identifier serves to identify the route to intermediate systems, such as packet-switching nodes, for handling future PDUs.
- **Multiplexing:** We address this function in more general terms later. Here we note that an entity may wish to enjoy more than one connection simultaneously. Thus, incoming PDUs must be identified by connection identifier.
- **Use of state information:** Once a connection is established, the end systems can maintain state information relating to the connection. This enables such functions as flow control and error control using sequence numbers. We see examples of this with HDLC (Chapter 7) and X.25 (Chapter 10).

Figure 2.4 shows several examples of connections. The logical connection between router J and host B is at the network level. For example, if network 2 is a packet-switching network using X.25, then this logical connection would be a virtual circuit. At a higher level, many transport-level protocols, such as TCP, support logical connections between users of the transport service. Thus, TCP can maintain a connection between two ports on different systems.

Another addressing concept is that of **addressing mode**. Most commonly, an address refers to a single system or port; in this case it is referred to as an individual or *unicast address*. It is also possible for an address to refer to more than one entity or port. Such an address identifies multiple simultaneous recipients for data. For example, a user might wish to send a memo to a number of individuals. The network control center may wish to notify all users that the network is going down. An address for multiple recipients may be *broadcast*, intended for all entities within a domain, or *multicast*, intended for a specific subset of entities. Table 2.1 illustrates the possibilities.

Multiplexing

Related to the concept of addressing is that of multiplexing. One form of multiplexing is supported by means of multiple connections into a single system. For

Table 2.1 Addressing Modes

Destination	Network Address	System Address	Port/SAP Address
Unicast	Individual	Individual	Individual
Multicast	Individual	Individual	Group
	Individual	All	Group
	All	All	Group
Broadcast	Individual	Individual	All
	Individual	All	All
	All	All	All

Network System Technologies Ex. 2010
Samsung v. Network System Technologies - IPR 2024-00338

- Circuit switching is used in public telephone networks and is the basis for private networks built on leased lines and using on-site circuit switches. Circuit switching was developed to handle voice traffic but can also handle digital data, although this latter use is often inefficient.
- With circuit switching, a dedicated path is established between two stations for communication. Switching and transmission resources within the network are reserved for the exclusive use of the circuit for the duration of the connection. The connection is transparent: Once it is established, it appears to attached devices as if there were a direct connection.
- Several important aspects of circuit-switching networks have changed dramatically in the wake of the increasing complexity and digitalization of public telecommunications networks. Simple hierarchical routing schemes have been replaced with more flexible and powerful nonhierarchical schemes. This reflects a corresponding change in the underlying architecture, which leads to increased efficiency and resilience. Simple inchannel control signaling methods have been replaced with more complex and higher-speed common channel signaling.

Since the invention of the telephone, circuit switching has been the dominant technology for voice communications, and it has remained so well into the ISDN era. This chapter begins with an introduction to the concept of a switched communications network and then looks at the key characteristics of a circuit-switching network.

9.1 SWITCHING NETWORKS

For transmission of data[1] beyond a local area, communication is typically achieved by transmitting data from source to destination through a network of intermediate switching nodes; this switched network design is sometimes used to implement LANs as well. The switching 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. Figure 9.1 illustrates a simple network. The end devices that wish to communicate may be referred to as *stations*. The stations may be computers, terminals, telephones, or other communicating devices. We shall refer to the switching devices whose purpose is to provide communication as *nodes*. The nodes are connected to one another in some topology by transmission links. Each station attaches to a node, and the collection of nodes is referred to as a *communications network*.

[1]We use this term here in a very general sense, to include voice, image, and video, as well as ordinary data (e.g., numerical, text).

Network System Technologies Ex. 2010
Samsung v. Network System Technologies - IPR 2024-00338

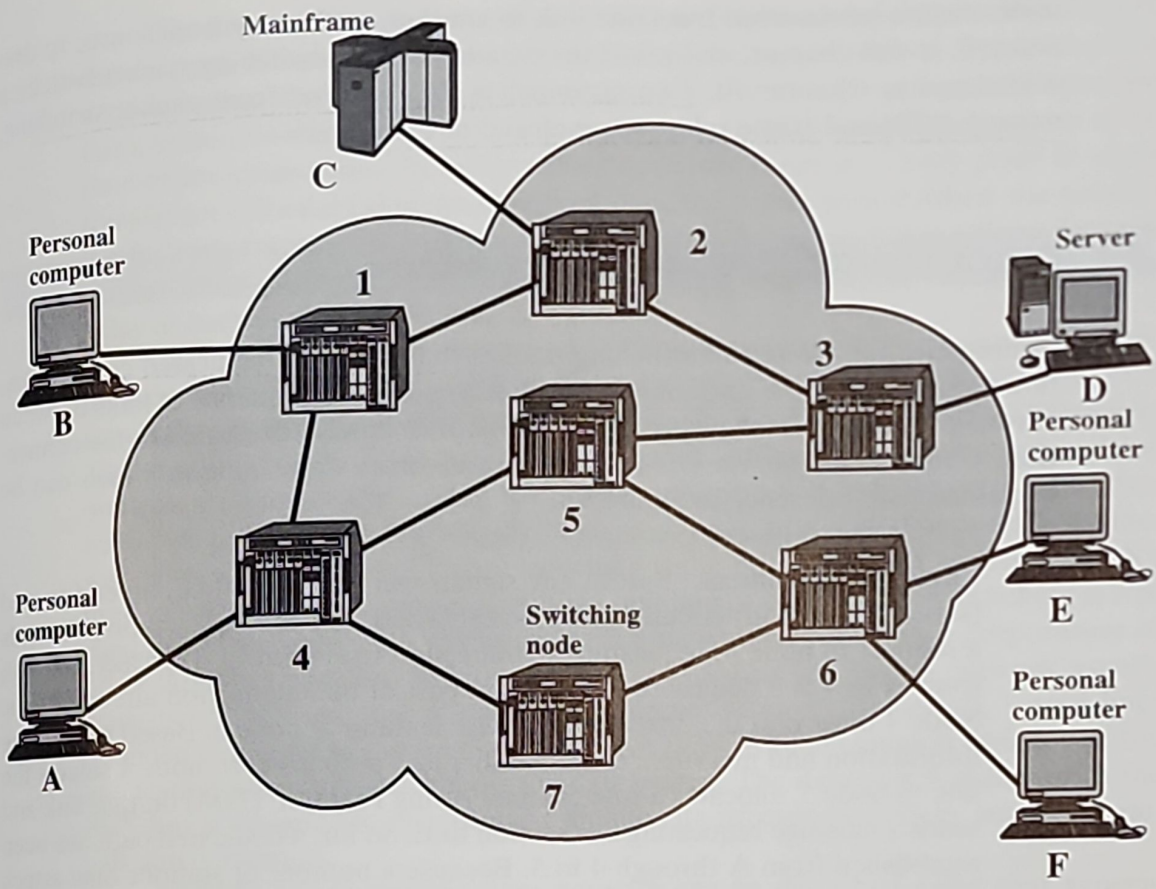


Figure 9.1 Simple Switching Network

The types of networks that are discussed in this and the next three chapters are referred to as *switched communication networks*. Data entering the network from a station are routed to the destination by being switched from node to node. For example, in Figure 9.1, data from station A intended for station F are sent to node 4. They may then be routed via nodes 5 and 6 or nodes 7 and 6 to the destination. Several observations are in order:

1. Some nodes connect only to other nodes (e.g., 5 and 7). Their sole task is the internal (to the network) switching of data. Other nodes have one or more stations attached as well; in addition to their switching functions, such nodes accept data from and deliver data to the attached stations.
2. Node-node links are usually multiplexed, using either frequency-division multiplexing (FDM) or time-division multiplexing (TDM).
3. Usually, the network is not fully connected; that is, there is not a direct link between every possible pair of nodes. However, it is always desirable to have more than one possible path through the network for each pair of stations. This enhances the reliability of the network.

Two quite different technologies are used in wide area switched networks: circuit switching and packet switching. These two technologies differ in the way the

Network System Technologies Ex. 2010
Samsung v. Network System Technologies - IPR 2024-00338

10.1 PACKET-SWITCHING PRINCIPLES

The long-haul circuit-switching telecommunications network was originally designed to handle voice traffic, and the majority of traffic on these networks continues to be voice. A key characteristic of circuit-switching networks is that resources within the network are dedicated to a particular call. For voice connections, the resulting circuit will enjoy a high percentage of utilization because, most of the time, one party or the other is talking. However, as the circuit-switching network began to be used increasingly for data connections, two shortcomings became apparent:

- In a typical user/host data connection (e.g., personal computer user logged on to a database server), much of the time the line is idle. Thus, with data connections, a circuit-switching approach is inefficient.
- In a circuit-switching network, the connection provides for transmission at a constant data rate. Thus, each of the two devices that are connected must transmit and receive at the same data rate as the other. This limits the utility of the network in interconnecting a variety of host computers and workstations.

To understand how packet switching addresses these problems, let us briefly summarize packet-switching operation. Data are transmitted in short packets. A typical upper bound on packet length is 1000 octets (bytes). If a source has a longer message to send, the message is broken up into a series of packets (Figure 10.1). Each packet contains a portion (or all for a short message) of the user's data plus some control information. The control information, at a minimum, includes the information that the network requires to be able to route the packet through the network and deliver it to the intended destination. At each node en route, the packet is received, stored briefly, and passed on to the next node.

Let us return to Figure 9.1, but now assume that it depicts a simple packet-switching network. Consider a packet to be sent from station A to station E. The packet includes control information that indicates that the intended destination is E. The packet is sent from A to node 4. Node 4 stores the packet, determines the next leg of the route (say 5), and queues the packet to go out on that link (the 4-5 link). When the link is available, the packet is transmitted to node 5, which forwards the packet to node 6, and finally to E. This approach has a number of advantages over circuit switching:

- Line efficiency is greater, because a single node-to-node link can be dynamically shared by many packets over time. The packets are queued up and transmitted as rapidly as possible over the link. By contrast, with circuit switching, time on a node-to-node link is preallocated using synchronous time-division multiplexing. Much of the time, such a link may be idle because a portion of its time is dedicated to a connection that is idle.
- A packet-switching network can perform data-rate conversion. Two stations of different data rates can exchange packets because each connects to its node at its proper data rate.

Network System Technologies Ex. 2010
Samsung v. Network System Technologies - IPR 2024-00338

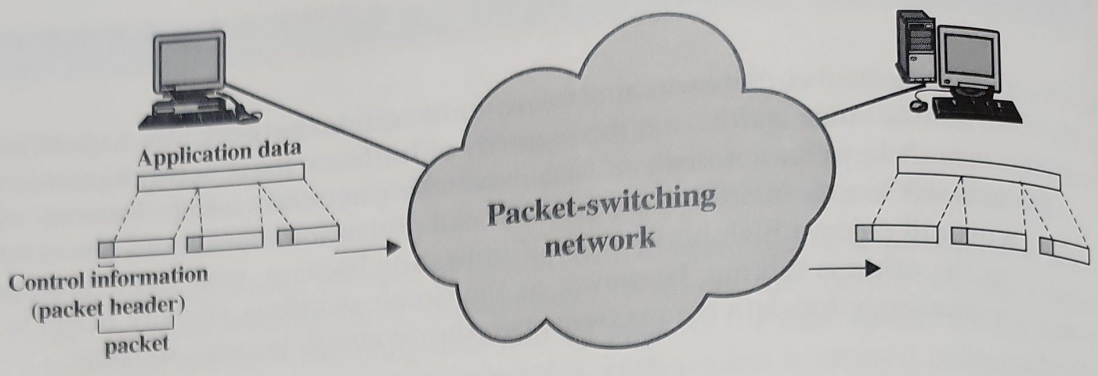


Figure 10.1 The Use of Packets

- When traffic becomes heavy on a circuit-switching network, some calls are blocked; that is, the network refuses to accept additional connection requests until the load on the network decreases. On a packet-switching network, packets are still accepted, but delivery delay increases.
- Priorities can be used. Thus, if a node has a number of packets queued for transmission, it can transmit the higher-priority packets first. These packets will therefore experience less delay than lower-priority packets.

Switching Technique

If a station has a message to send through a packet-switching network that is of length greater than the maximum packet size, it breaks the message up into packets and sends these packets, one at a time, to the network. A question arises as to how the network will handle this stream of packets as it attempts to route them through the network and deliver them to the intended destination. There are two approaches that are used in contemporary networks: datagram and virtual circuit.

In the **datagram** approach, each packet is treated independently, with no reference to packets that have gone before. Let us consider the implication of this approach. Suppose that station A in Figure 9.1 has a three-packet message to send to E. It transmits the packets, 1-2-3, to node 4. Each packet contains the address of its destination, which in this case is E. For each packet, node 4 must make a routing decision. Packet 1 arrives for delivery to E. Node 4 could plausibly forward this packet to either node 5 or node 7 as the next step in the route. In this case, node 4 determines that its queue of packets for node 5 is shorter than for node 7, so it queues the packet for node 5. Ditto for packet 2. But for packet 3, node 4 finds that its queue for node 7 is now shorter and so queues packet 3 for that node. So the packets, each with the same destination address, do not all follow the same route. As a result, it is possible that packet 3 will beat packet 2 and even packet 1 to node 6. Thus, it is also possible that the packets will be delivered to E in a different sequence from the one in which they were sent. It is up to E to figure out how to reorder them. Also, it is possible for a packet to be destroyed in the network. For example, if a packet-switching node crashes momentarily, all of its queued packets may be lost. If this were to happen to one of the packets in our

Network System Technologies Ex. 2010
Samsung v. Network System Technologies - IPR 2024-00338

example, node 6 has no way of knowing that one of the packets in the sequence of packets has been lost. Again, it is up to E to detect the loss of a packet and figure out how to recover it. In this technique, each packet, treated independently, is referred to as a datagram.

In the **virtual circuit** approach, a preplanned route is established before any packets are sent. For example, suppose that A has one or more messages to send to E. It first sends a special control packet, referred to as a Call Request packet, to 4, requesting a logical connection to E. Node 4 decides to route the request and all subsequent packets to 5, which decides to route the request and all subsequent packets to 6, which finally delivers the Call Request packet to E. If E is prepared to accept the connection, it sends a Call Accept packet to 6. This packet is passed back through nodes 5 and 4 to A. Stations A and E may now exchange data over the route that has been established. Because the route is fixed for the duration of the logical connection, it is somewhat similar to a circuit in a circuit-switching network and is referred to as a virtual circuit. Each packet contains a virtual circuit identifier, instead of a destination address, as well as data. Each node on the preestablished route knows where to direct such packets; no routing decisions are required. Thus, every data packet from A intended for E traverses nodes 4, 5, and 6; every data packet from E intended for A traverses nodes 6, 5, and 4. Eventually, one of the stations terminates the connection with a Clear Request packet. At any time, each station can have more than one virtual circuit to any other station and can have virtual circuits to more than one station.

So the main characteristic of the virtual circuit technique is that a route between stations is set up prior to data transfer. Note that this does not mean that this is a dedicated path, as in circuit switching. A packet is still buffered at each node, and queued for output over a line, while other packets on other virtual circuits may share the use of the line. The difference from the datagram approach is that, with virtual circuits, the node need not make a routing decision for each packet. It is made only once for all packets using that virtual circuit.

If two stations wish to exchange data over an extended period of time, there are certain advantages to virtual circuits. First, the network may provide services related to the virtual circuit, including sequencing and error control. Sequencing refers to the fact that, because all packets follow the same route, they arrive in the original order. Error control is a service that assures not only that packets arrive in proper sequence, but that all packets arrive correctly. For example, if a packet in a sequence from node 4 to node 6 fails to arrive at node 6, or arrives with an error, node 6 can request a retransmission of that packet from node 4. Another advantage is that packets should transit the network more rapidly with a virtual circuit; it is not necessary to make a routing decision for each packet at each node.

One advantage of the datagram approach is that the call setup phase is avoided. Thus, if a station wishes to send only one or a few packets, datagram delivery will be quicker. Another advantage of the datagram service is that, because it is more primitive, it is more flexible. For example, if congestion develops in one part of the network, incoming datagrams can be routed away from the congestion. With the use of virtual circuits, packets follow a predefined route, and thus it is more difficult for the network to adapt to congestion. A third advantage is that datagram delivery is inherently more reliable. With the use of virtual circuits, if a node fails,

Network System Technologies Ex. 2010
Samsung v. Network System Technologies - IPR 2024-00338

all virtual circuits that pass through that node are lost. With datagram delivery, if a node fails, subsequent packets may find an alternate route that bypasses that node.

Most currently available packet-switching networks make use of virtual circuits for their internal operation. To some degree, this reflects a historical motivation to provide a network that presents a service as reliable (in terms of sequencing) as a circuit-switching network. There are, however, several providers of private packet-switching networks that make use of datagram operation. From the user's point of view, there should be very little difference in the external behavior based on the use of datagrams or virtual circuits. A datagram-style of operation is common in internetworks, discussed in Part Five.

Packet Size

There is a significant relationship between packet size and transmission time, as shown in Figure 10.2. In this example, it is assumed that there is a virtual circuit from station X through nodes a and b to station Y. The message to be sent comprises 40 octets, and each packet contains 3 octets of control information, which is placed at the beginning of each packet and is referred to as a header. If the entire message is sent as a single packet of 43 octets (3 octets of header plus 40 octets of data), then the packet is first transmitted from station X to node a (Figure 10.2a). When the entire packet is received, it can then be transmitted from a to b. When the entire packet is received at node b, it is then transferred to station Y. Ignoring switching time, total transmission time is 129 octet-times (43 octets $\times$ 3 packet transmissions).

Suppose now that we break the message up into two packets, each containing 20 octets of the message and, of course, 3 octets each of header, or control information. In this case, node a can begin transmitting the first packet as soon as it has arrived from X, without waiting for the second packet. Because of this overlap in transmission, the total transmission time drops to 92 octet-times. By breaking the message up into five packets, each intermediate node can begin transmission even sooner and the savings in time is greater, with a total of 77 octet-times for transmission. However, this process of using more and smaller packets eventually results in increased, rather than reduced, delay as illustrated in Figure 10.2d. This is because each packet contains a fixed amount of header, and more packets mean more of these headers. Furthermore, the example does not show the processing and queuing delays at each node. These delays are also greater when more packets are handled for a single message. However, we shall see in the next chapter that an extremely small packet size (53 octets) can result in an efficient network design.

Comparison of Circuit Switching and Packet Switching

Having looked at the internal operation of packet switching, we can now return to a comparison of this technique with circuit switching. We first look at the important issue of performance and then examine other characteristics.

Performance

A simple comparison of circuit switching and the two forms of packet switching is provided in Figure 10.3. The figure depicts the transmission of a message

Network System Technologies Ex. 2010
Samsung v. Network System Technologies - IPR 2024-00338

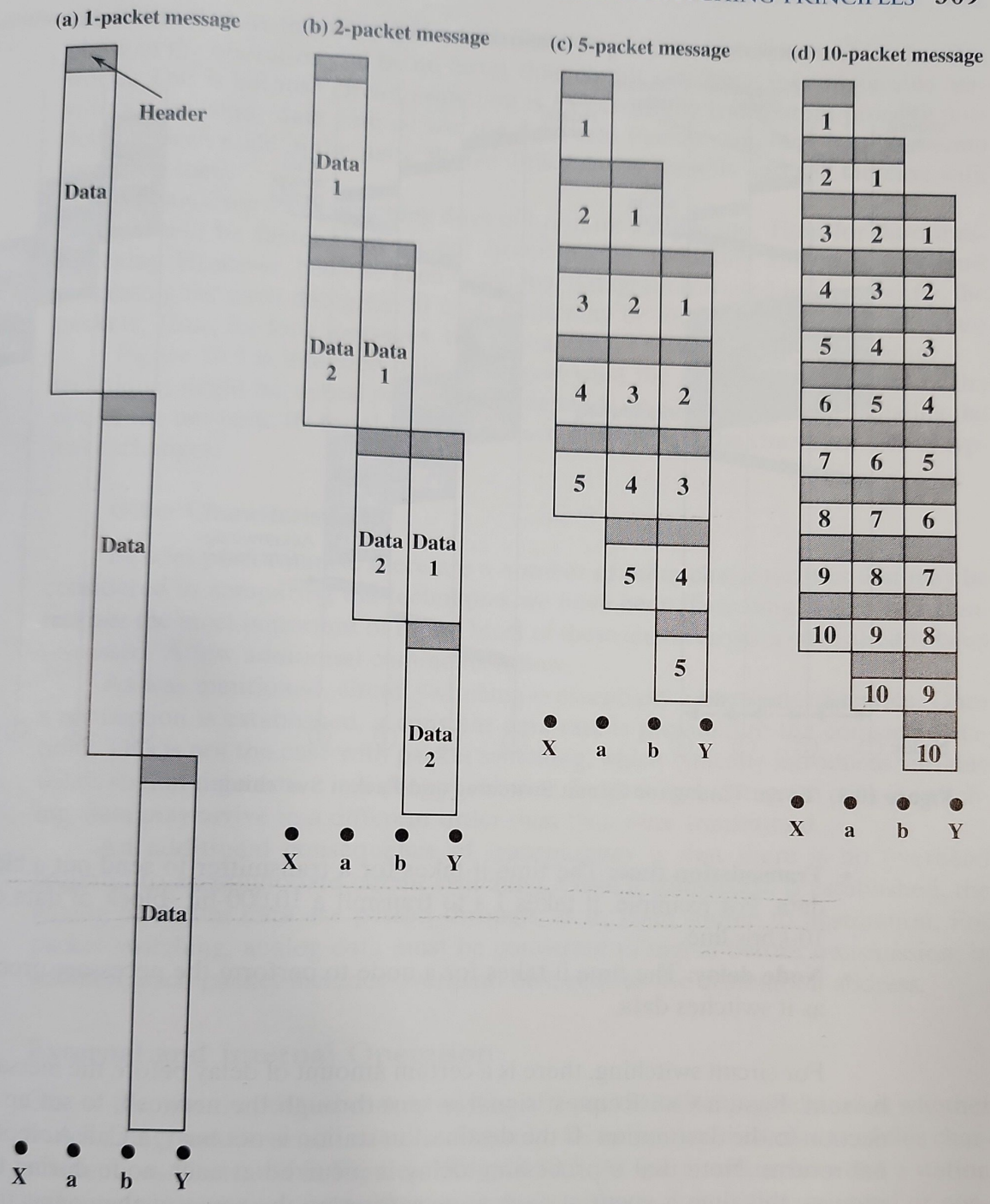


Figure 10.2 Effect of Packet Size on Transmission Frame

across four nodes, from a source station attached to node 1 to a destination station attached to node 4. In this figure, we are concerned with three types of delay:

- **Propagation delay:** The time it takes a signal to propagate from one node to the next. This time is generally negligible. The speed of electromagnetic signals through a wire medium, for example, is typically 2×10^8 m/s.

Network System Technologies Ex. 2010
Samsung v. Network System Technologies - IPR 2024-00338

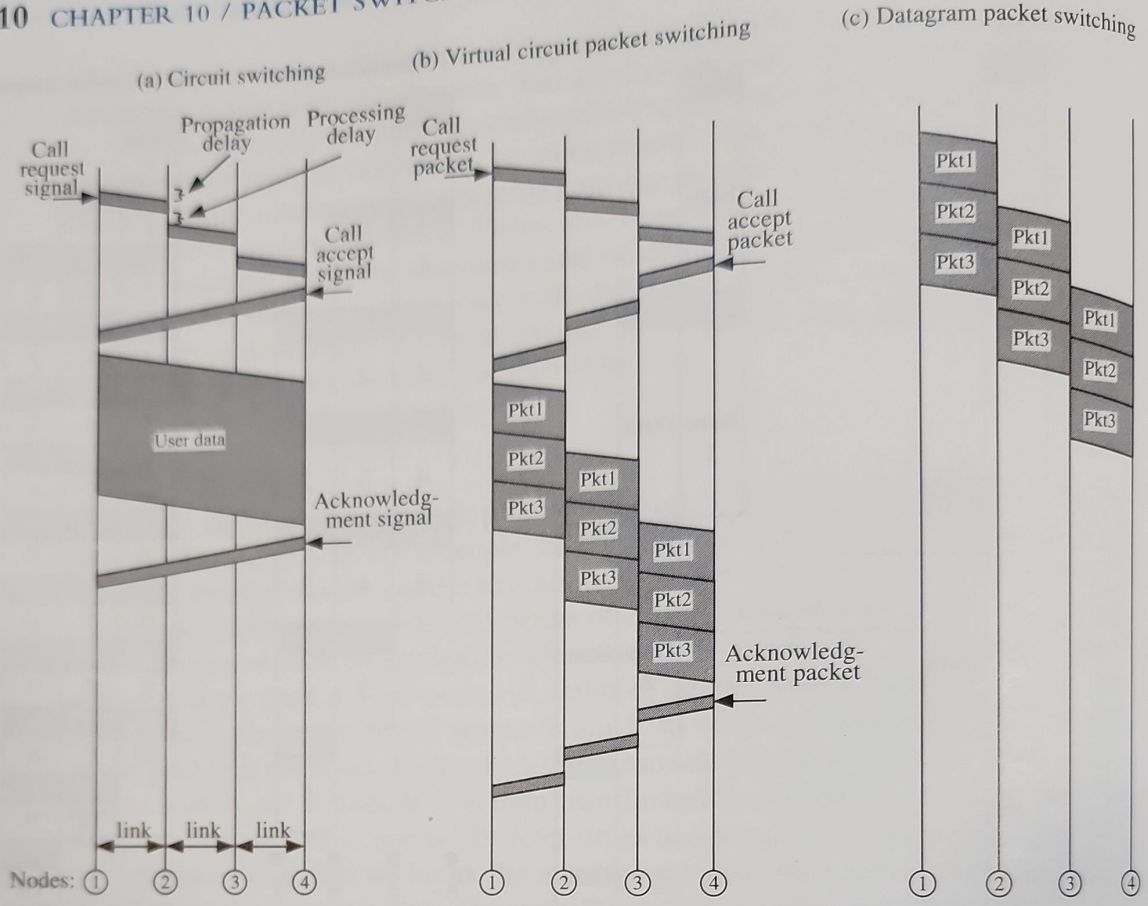


Figure 10.3 Event Timing for Circuit Switching and Packet Switching

- **Transmission time:** The time it takes for a transmitter to send out a block of data. For example, it takes 1 s to transmit a 10,000-bit block of data onto a 10-kbps line.
- **Node delay:** The time it takes for a node to perform the necessary processing as it switches data.

For circuit switching, there is a certain amount of delay before the message can be sent. First, a Call Request signal is sent through the network, to set up a connection to the destination. If the destination station is not busy, a Call Accepted signal returns. Note that a processing delay is incurred at each node during the call request; this time is spent at each node setting up the route of the connection. On the return, this processing is not needed because the connection is already set up. After the connection is set up, the message is sent as a single block, with no noticeable delay at the switching nodes.

Virtual circuit packet switching appears quite similar to circuit switching. A virtual circuit is requested using a Call Request packet, which incurs a delay at each node. The virtual circuit is accepted with a Call Accept packet. In contrast to the circuit-switching case, the call acceptance also experiences node delays, even though the virtual circuit route is now established. The reason is that this packet is queued at each node and must wait its turn for transmission. Once the virtual cir-

Network System Technologies Ex. 2010
Samsung v. Network System Technologies - IPR 2024-00338

cuit is established, the message is transmitted in packets. It should be clear that this phase of the operation can be no faster than circuit switching, for comparable networks. This is because circuit switching is an essentially transparent process, providing a constant data rate across the network. Packet switching involves some delay at each node in the path. Worse, this delay is variable and will increase with increased load.

Datagram packet switching does not require a call setup. Thus, for short messages, it will be faster than virtual circuit packet switching and perhaps circuit switching. However, because each individual datagram is routed independently, the processing for each datagram at each node may be longer than for virtual circuit packets. Thus, for long messages, the virtual circuit technique may be superior.

Figure 10.3 is intended only to suggest what the relative performance of the techniques might be; actual performance depends on a host of factors, including the size of the network, its topology, the pattern of load, and the characteristics of typical exchanges.

Other Characteristics

Besides performance, there are a number of other characteristics that may be considered in comparing the techniques we have been discussing. Table 10.1 summarizes the most important of these. Most of these characteristics have already been discussed. A few additional comments follow.

As was mentioned, circuit switching is essentially a transparent service. Once a connection is established, a constant data rate is provided to the connected stations. This is not the case with packet switching, which typically introduces variable delay, so that data arrive in a choppy manner. Indeed, with datagram packet switching, data may arrive in a different order than they were transmitted.

An additional consequence of transparency is that there is no overhead required to accommodate circuit switching. Once a connection is established, the analog or digital data are passed through, as is, from source to destination. For packet switching, analog data must be converted to digital before transmission; in addition, each packet includes overhead bits, such as the destination address.

External and Internal Operation

One of the most important characteristics of a packet-switching network is whether it uses datagrams or virtual circuits. Actually, there are two dimensions of this characteristic, as illustrated in Figures 10.4 and 10.5. At the interface between a station and a network node, a network may provide either a connection-oriented or connectionless service. With a connection-oriented service, a station performs a call request to set up a logical connection to another station. All packets presented to the network are identified as belonging to a particular logical connection and are numbered sequentially. The network undertakes to deliver packets in sequence-number order. The logical connection is usually referred to as a virtual circuit, and the connection-oriented service is referred to as an **external virtual circuit service**; unfortunately, this external service is distinct from the concept of **internal virtual circuit operation**, as we shall see. An important example of an external virtual circuit service is X.25, which is examined in Section 10.3.

Network System Technologies Ex. 2010
Samsung v. Network System Technologies - IPR 2024-00338

Table 10.1 Comparison of Communication Switching Techniques

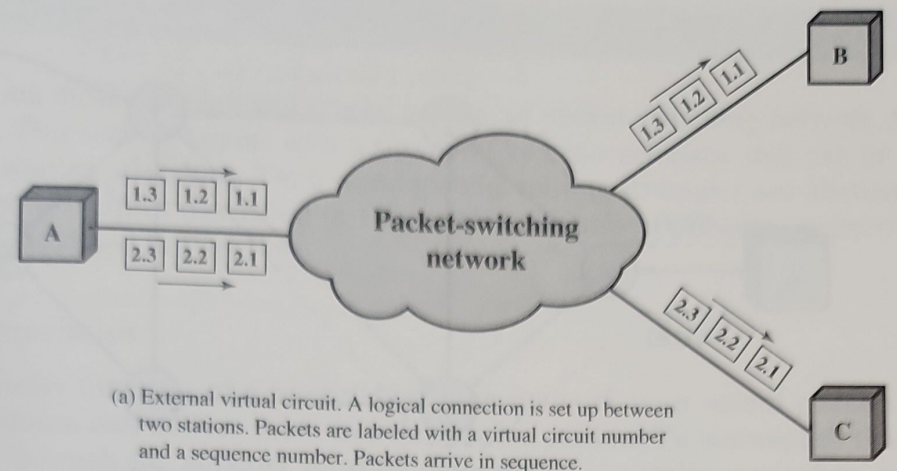
Circuit switching	Datagram packet switching	Virtual-circuit packet switching
Dedicated transmission path	No dedicated path	No dedicated path
Continuous transmission of data	Transmission of packets	Transmission of packets
Fast enough for interactive	Fast enough for interactive	Fast enough for interactive
Messages are not stored	Packets may be stored until delivered	Packets stored until delivered
The path is established for entire conversation	Route established for each packet	Route established for entire conversation
Call setup delay; negligible transmission delay	Packet transmission delay	Call setup delay; packet transmission delay
Busy signal if called party busy	Sender may be notified if packet not delivered	Sender notified of connection denial
Overload may block call setup; no delay for established calls	Overload increases packet delay	Overload may block call setup; increases packet delay
Electromechanical or computerized switching	Small switching nodes nodes	Small switching nodes
User responsible for message loss protection	Network may be responsible for individual packets	Network may be responsible for packet sequences
Usually no speed or code conversion	Speed and code conversion	Speed and code conversion
Fixed bandwidth	Dynamic use of bandwidth	Dynamic use of bandwidth
No overhead bits after call setup	Overhead bits in each packet	Overhead bits in each packet

With connectionless service, the network only agrees to handle packets independently, and may not deliver them in order or reliably. This type of service is sometimes known as an **external datagram service**; again, this is a distinct concept from that of **internal datagram operation**. Internally, the network may actually construct a fixed route between endpoints (virtual circuit) or not (datagram).

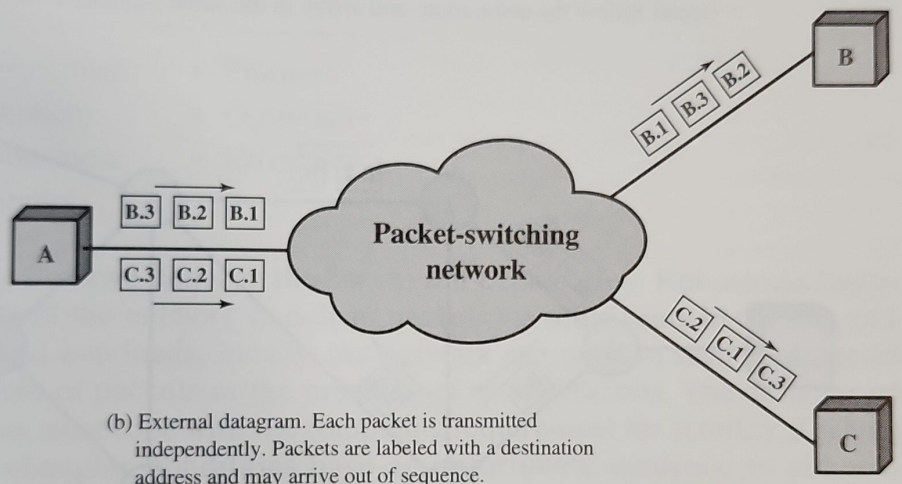
These internal and external design decisions need not coincide:

- **External virtual circuit, internal virtual circuit:** When the user requests a virtual circuit, a dedicated route through the network is constructed. All packets follow that same route.
- **External virtual circuit, internal datagram:** The network handles each packet separately. Thus, different packets for the same external virtual circuit may take different routes. However, the network buffers packets at the destination

Network System Technologies Ex. 2010
Samsung v. Network System Technologies - IPR 2024-00338



(a) External virtual circuit. A logical connection is set up between two stations. Packets are labeled with a virtual circuit number and a sequence number. Packets arrive in sequence.



(b) External datagram. Each packet is transmitted independently. Packets are labeled with a destination address and may arrive out of sequence.

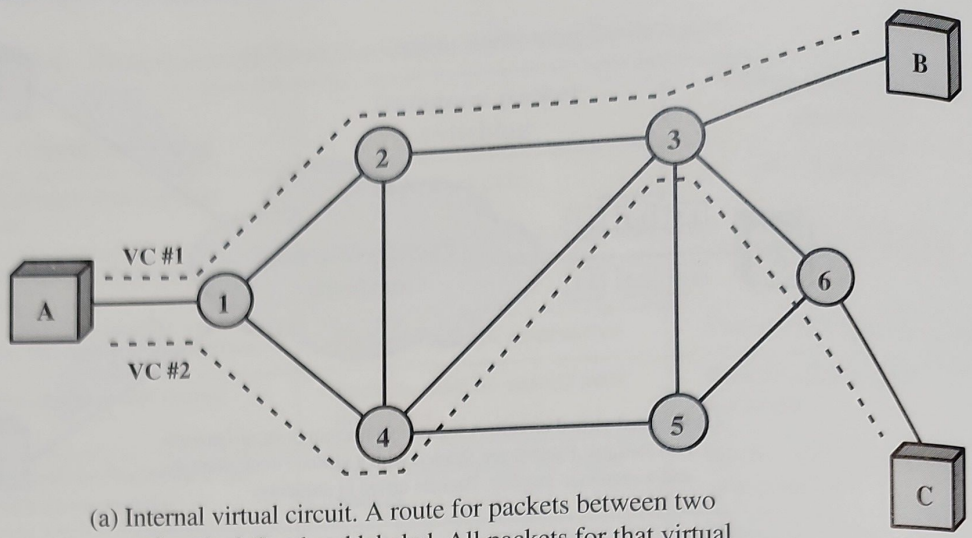
Figure 10.4 External Virtual Circuit and Datagram Operation

node, if necessary, so that they are delivered to the destination station in the proper order.

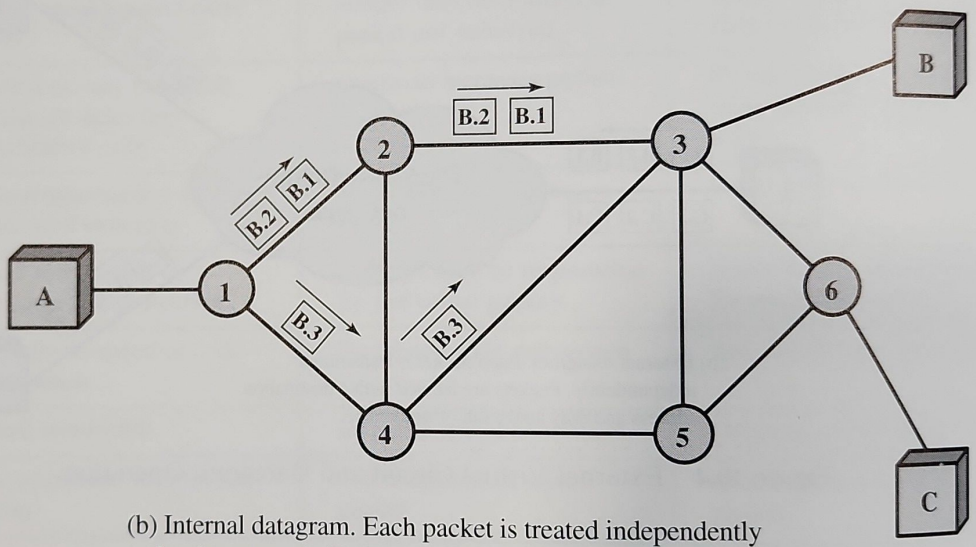
- **External datagram, internal datagram:** Each packet is treated independently from both the user's and the network's point of view.
- **External datagram, internal virtual circuit:** The external user does not see any connections, simply sending packets one at a time. The network, however, sets up a logical connection between stations for packet delivery. The network may leave such connections in place for an extended period, to satisfy anticipated future needs.

The question arises as to the choice of virtual circuits or datagrams, both internally and externally. This will depend on the specific design objectives for the communication network and the cost factors that prevail. We have already made some comments concerning the relative merits of internal datagram versus virtual circuit

Network System Technologies Ex. 2010
Samsung v. Network System Technologies - IPR 2024-00338



(a) Internal virtual circuit. A route for packets between two stations is defined and labeled. All packets for that virtual circuit follow the same route and arrive in the same sequence.



(b) Internal datagram. Each packet is treated independently by the network. Packets are labeled with a destination address and may arrive at the destination node out of sequence.

Figure 10.5 Internal Virtual Circuit and Datagram Operation

operation. With respect to external service, we can make the following observations. The datagram service, coupled with internal datagram operation, allows for efficient use of the network; there is no call setup and no need to hold up packets while a packet in error is retransmitted. This latter feature is desirable in some real-time applications. The virtual circuit service can provide end-to-end sequencing and error control. This service is attractive for supporting connection-oriented applications, such as file transfer and remote terminal access. In practice, the virtual circuit service is much more common than the datagram service. The reliability and convenience of a connection-oriented service is seen as more attractive than the benefits of the datagram service.

Network System Technologies Ex. 2010
Samsung v. Network System Technologies - IPR 2024-00338

13.7 BRIDGES

In virtually all cases, there is a need to expand beyond the confines of a single LAN, to provide interconnection to other LANs and to wide area networks. Two general approaches are used for this purpose: bridges and routers. The bridge is the simpler of the two devices and provides a means of interconnecting similar LANs. The router is a more general-purpose device, capable of interconnecting a variety of LANs and WANs. We explore bridges in this section and look at routers in Part Five.

The bridge is designed for use between local area networks (LANs) that use identical protocols for the physical and link layers (e.g., all conforming to IEEE 802.3). Because the devices all use the same protocols, the amount of processing required at the bridge is minimal. More sophisticated bridges are capable of mapping from one MAC format to another (e.g., to interconnect an Ethernet and a token ring LAN).

Because the bridge is used in a situation in which all the LANs have the same characteristics, the reader may ask, why not simply have one large LAN? Depending on circumstance, there are several reasons for the use of multiple LANs connected by bridges:

- **Reliability:** The danger in connecting all data processing devices in an organization to one network is that a fault on the network may disable communication for all devices. By using bridges, the network can be partitioned into self-contained units.
- **Performance:** In general, performance on a LAN declines with an increase in the number of devices or the length of the wire. A number of smaller LANs will often give improved performance if devices can be clustered so that intranetwork traffic significantly exceeds internetwork traffic.
- **Security:** The establishment of multiple LANs may improve security of communications. It is desirable to keep different types of traffic (e.g., accounting, personnel, strategic planning) that have different security needs on physically separate media. At the same time, the different types of users with different levels of security need to communicate through controlled and monitored mechanisms.
- **Geography:** Clearly, two separate LANs are needed to support devices clustered in two geographically distant locations. Even in the case of two buildings separated by a highway, it may be far easier to use a microwave bridge link than to attempt to string coaxial cable between the two buildings.

Functions of a Bridge

Figure 13.14 illustrates the action of a bridge connecting two LANs, A and B, using the same MAC protocol. In this example, a single bridge attaches to both LANs; frequently, the bridge function is performed by two "half-bridges," one on each LAN. The functions of the bridge are few and simple:

- Read all frames transmitted on A and accept those addressed to any station on B.
- Using the medium access control protocol for B, retransmit each frame on B.
- Do the same for B-to-A traffic.

Network System Technologies Ex. 2010
Samsung v. Network System Technologies - IPR 2024-00338

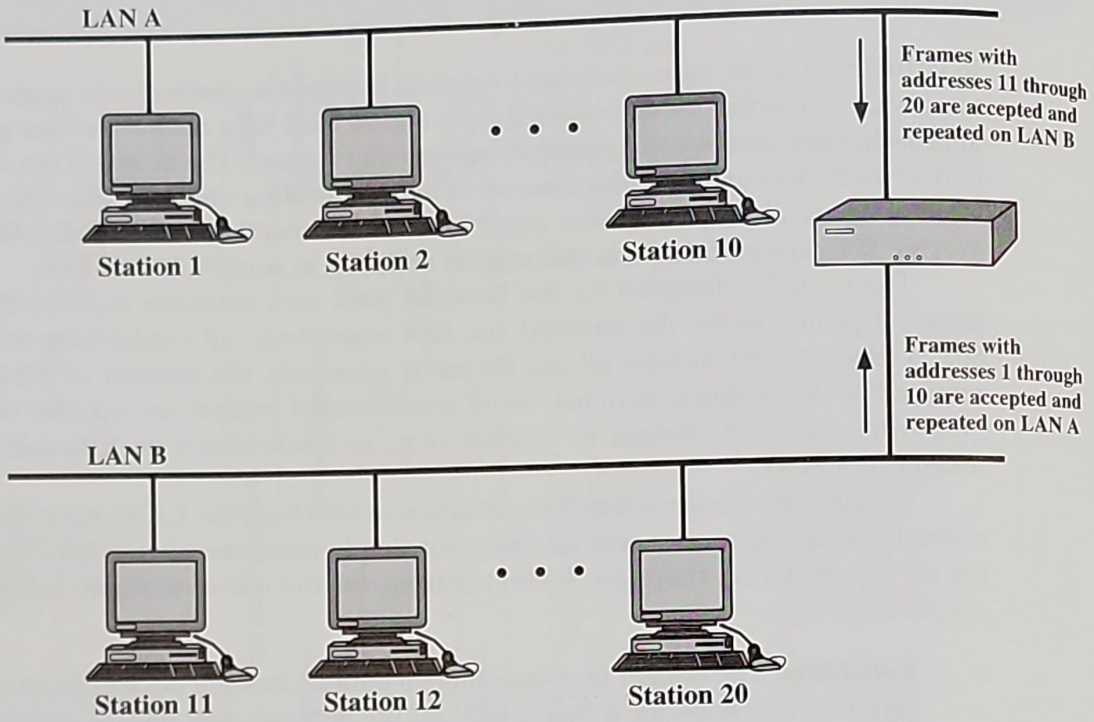


Figure 13.14 Bridge Operation

Several design aspects of a bridge are worth highlighting:

- The bridge makes no modification to the content or format of the frames it receives, nor does it encapsulate them with an additional header. Each frame to be transferred is simply copied from one LAN and repeated with exactly the same bit pattern as the other LAN. Because the two LANs use the same LAN protocols, it is permissible to do this.
- The bridge should contain enough buffer space to meet peak demands. Over a short period of time, frames may arrive faster than they can be retransmitted.
- The bridge must contain addressing and routing intelligence. At a minimum, the bridge must know which addresses are on each network to know which frames to pass. Further, there may be more than two LANs interconnected by a number of bridges. In that case, a frame may have to be routed through several bridges in its journey from source to destination.
- A bridge may connect more than two LANs.

In summary, the bridge provides an extension to the LAN that requires no modification to the communications software in the stations attached to the LANs. It appears to all stations on the two (or more) LANs that there is a single LAN on which each station has a unique address. The station uses that unique address and need not explicitly discriminate between stations on the same LAN and stations on other LANs; the bridge takes care of that.

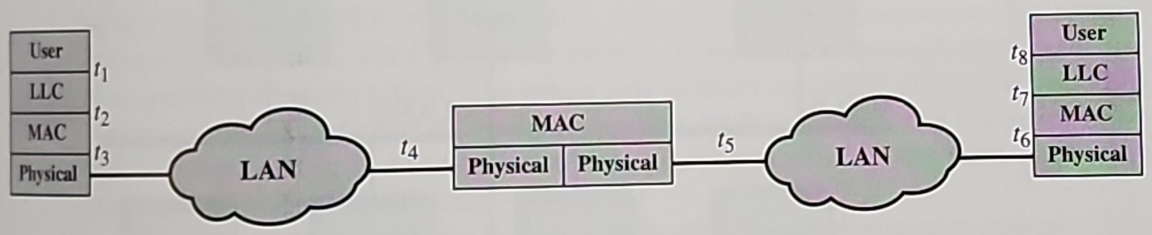
Network System Technologies Ex. 2010
Samsung v. Network System Technologies - IPR 2024-00338

Bridge Protocol Architecture

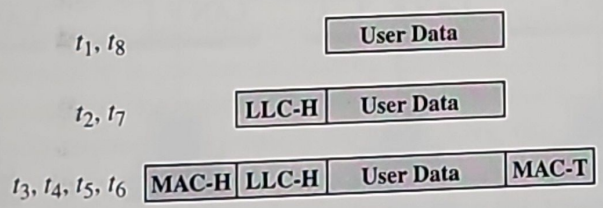
The IEEE 802.1D specification defines the protocol architecture for MAC bridges. Within the 802 architecture, the endpoint or station address is designated at the MAC level. Thus, it is at the MAC level that a bridge can function. Figure 13.15 shows the simplest case, which consists of two LANs connected by a single bridge. The LANs employ the same MAC and LLC protocols. The bridge operates as previously described. A MAC frame whose destination is not on the immediate LAN is captured by the bridge, buffered briefly, and then transmitted on the other LAN. As far as the LLC layer is concerned, there is a dialogue between peer LLC entities in the two endpoint stations. The bridge need not contain an LLC layer because it is merely serving to relay the MAC frames.

Figure 13.15b indicates the way in which data are encapsulated using a bridge. Data are provided by some user to LLC. The LLC entity appends a header and passes the resulting data unit to the MAC entity, which appends a header and a trailer to form a MAC frame. On the basis of the destination MAC address in the frame, it is captured by the bridge. The bridge does not strip off the MAC fields; its function is to relay the MAC frame intact to the destination LAN. Thus, the frame is deposited on the destination LAN and captured by the destination station.

The concept of a MAC relay bridge is not limited to the use of a single bridge to connect two nearby LANs. If the LANs are some distance apart, then they can be connected by two bridges that are in turn connected by a communications facility. The intervening communications facility can be a network, such as a wide area packet-switching network, or a point-to-point link. In such cases, when a bridge captures a MAC frame, it must encapsulate the frame in the appropriate packaging and



(a) Architecture



(b) Operation

Figure 13.15 Connection of Two LANs by a Bridge

Network System Technologies Ex. 2010
Samsung v. Network System Technologies - IPR 2024-00338

transmit it over the communications facility to a target bridge. The target bridge strips off these extra fields and transmits the original, unmodified MAC frame to the destination station.

Fixed Routing

There is a trend within many organizations to an increasing number of LANs interconnected by bridges. As the number of LANs grows, it becomes important to provide alternate paths between LANs via bridges for load balancing and reconfiguration in response to failure. Thus, many organizations will find that static, preconfigured routing tables are inadequate and that some sort of dynamic routing is needed.

Consider the configuration of Figure 13.16. Suppose that station 1 transmits a frame on LAN A intended for station 6. The frame will be read by bridges 101, 102, and 107. For each bridge, the addressed station is not on a LAN to which the bridge is attached. Therefore, each bridge must make a decision whether or not to re-

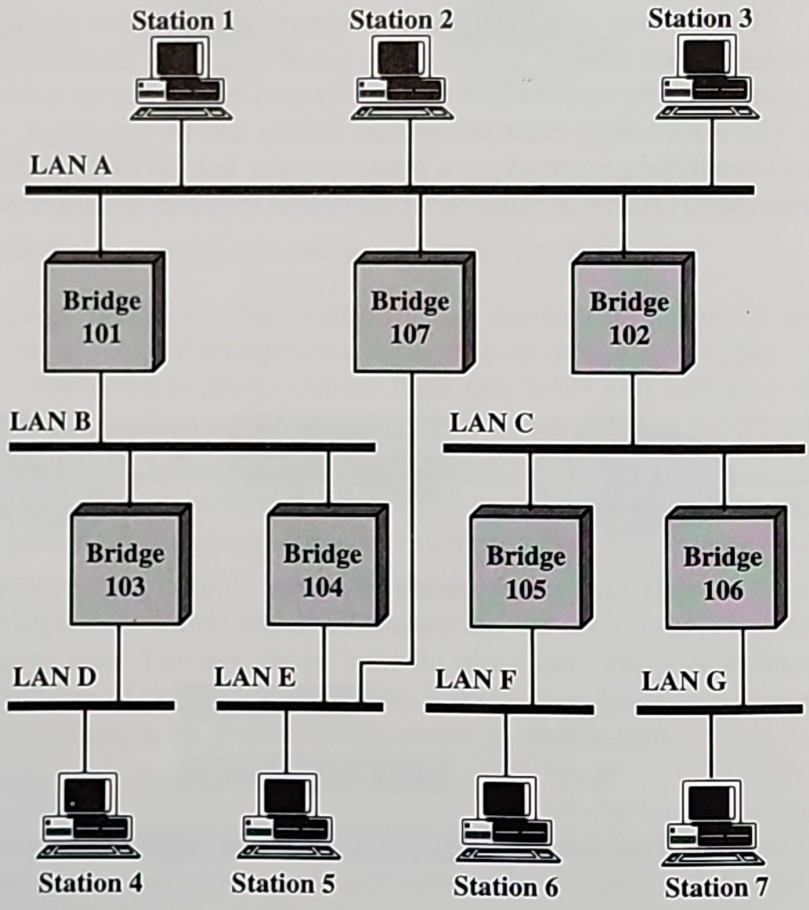


Figure 13.16 Configuration of Bridges and LANs, with Alternate Routes

Network System Technologies Ex. 2010
Samsung v. Network System Technologies - IPR 2024-00338

transmit the frame on its other LAN, in order to move it closer to its intended destination. In this case, bridge 102 should repeat the frame on LAN C, whereas bridges 101 and 107 should refrain from retransmitting the frame. Once the frame has been transmitted on LAN C, it will be picked up by both bridges 105 and 106. Again, each must decide whether or not to forward the frame. In this case, bridge 105 should retransmit the frame on LAN F, where it will be received by the destination, station 6.

Thus we see that, in the general case, the bridge must be equipped with a routing capability. When a bridge receives a frame, it must decide whether or not to forward it. If the bridge is attached to two or more networks, then it must decide whether or not to forward the frame and, if so, on which LAN the frame should be transmitted.

The routing decision may not always be a simple one. Figure 13.16 also shows that there are two routes between LAN A and LAN E. Such redundancy provides for higher overall internet availability and creates the possibility for load balancing. In this case, if station 1 transmits a frame on LAN A intended for station 5 on LAN E, then either bridge 101 or bridge 107 could forward the frame. It would appear preferable for bridge 107 to forward the frame, since it will involve only one hop, whereas if the frame travels through bridge 101, it must suffer two hops. Another consideration is that there may be changes in the configuration. For example, bridge 107 may fail, in which case subsequent frames from station 1 to station 5 should go through bridge 101. So we can say that the routing capability must take into account the topology of the internet configuration and may need to be dynamically altered.

A variety of routing strategies have been proposed and implemented in recent years. The simplest and most common strategy is **fixed routing**. This strategy is suitable for small internets and for internets that are relatively stable. In addition, two groups within the IEEE 802 committee have developed specifications for routing strategies. The IEEE 802.1 group has issued a standard for routing based on the use of a **spanning tree** algorithm. The token ring committee, IEEE 802.5, has issued its own specification, referred to as **source routing**. In the remainder of this section, we look at fixed routing and the spanning tree algorithm, which is the most commonly used bridge routing algorithm.

For fixed routing, a route is selected for each source–destination pair of LANs in the configuration. If alternate routes are available between two LANs, then typically the route with the least number of hops is selected. The routes are fixed, or at least only change when there is a change in the topology of the internet.

The strategy for developing a fixed routing configuration for bridges is similar to that employed in a packet-switching network (Figure 10.7). A central routing matrix is created, to be stored perhaps at a network control center. The matrix shows, for each source–destination pair of LANs, the identity of the first bridge on the route. So, for example, the route from LAN E to LAN F begins by going through bridge 107 to LAN A. Again consulting the matrix, the route from LAN A to LAN F goes through bridge 102 to LAN C. Finally, the route from LAN C to LAN F is directly through bridge 105. Thus the complete route from LAN E to LAN F is bridge 107, LAN A, bridge 102, LAN C, bridge 105.

Network System Technologies Ex. 2010
Samsung v. Network System Technologies - IPR 2024-00338

From this overall matrix, routing tables can be developed and stored at each bridge. Each bridge needs one table for each LAN to which it attaches. The information for each table is derived from a single row of the matrix. For example, bridge 105 has two tables, one for frames arriving from LAN C and one for frames arriving from LAN F. The table shows, for each possible destination MAC address, the identity of the LAN to which the bridge should forward the frame.

Once the directories have been established, routing is a simple matter. A bridge copies each incoming frame on each of its LANs. If the destination MAC address corresponds to an entry in its routing table, the frame is retransmitted on the appropriate LAN.

The fixed routing strategy is widely used in commercially available products. It requires that a network manager manually load the data into the routing tables. It has the advantage of simplicity and minimal processing requirements. However, in a complex internet, in which bridges may be dynamically added and in which failures must be allowed for, this strategy is too limited.

The Spanning Tree Approach

The spanning tree approach is a mechanism in which bridges automatically develop a routing table and update that table in response to changing topology. The algorithm consists of three mechanisms: frame forwarding, address learning, and loop resolution.

Frame Forwarding

In this scheme, a bridge maintains a **forwarding database** for each port attached to a LAN. The database indicates the station addresses for which frames should be forwarded through that port. We can interpret this in the following fashion. For each port, a list of stations is maintained. A station is on the list if it is on the "same side" of the bridge as the port. For example, for bridge 102 of Figure 13.16, stations on LANs C, F, and G are on the same side of the bridge as the LAN C port, and stations on LANs A, B, D, and E are on the same side of the bridge as the LAN A port. When a frame is received on any port, the bridge must decide whether that frame is to be forwarded through the bridge and out through one of the bridge's other ports. Suppose that a bridge receives a MAC frame on port x. The following rules are applied:

1. Search the forwarding database to determine if the MAC address is listed for any port except port x.
2. If the destination MAC address is not found, forward frame out all ports except the one from which is was received. This is part of the learning process described subsequently.
3. If the destination address is in the forwarding database for some port y, then determine whether port y is in a blocking or forwarding state. For reasons explained later, a port may sometimes be blocked, which prevents it from receiving or transmitting frames.
4. If port y is not blocked, transmit the frame through port y onto the LAN to which that port attaches.

Network System Technologies Ex. 2010
Samsung v. Network System Technologies - IPR 2024-00338

Address Learning

The preceding scheme assumes that the bridge is already equipped with a forwarding database that indicates the direction, from the bridge, of each destination station. This information can be preloaded into the bridge, as in fixed routing. However, an effective automatic mechanism for learning the direction of each station is desirable. A simple scheme for acquiring this information is based on the use of the source address field in each MAC frame.

The strategy is this. When a frame arrives on a particular port, it clearly has come from the direction of the incoming LAN. The source address field of the frame indicates the source station. Thus, a bridge can update its forwarding database for that port on the basis of the source address field of each incoming frame. To allow for changes in topology, each element in the database is equipped with a timer. When a new element is added to the database, its timer is set. If the timer expires, then the element is eliminated from the database, since the corresponding direction information may no longer be valid. Each time a frame is received, its source address is checked against the database. If the element is already in the database, the entry is updated (the direction may have changed) and the timer is reset. If the element is not in the database, a new entry is created, with its own timer.

Spanning Tree Algorithm

The address learning mechanism described previously is effective if the topology of the internet is a tree; that is, if there are no alternate routes in the network. The existence of alternate routes means that there is a closed loop. For example in Figure 13.16, the following is a closed loop: LAN A, bridge 101, LAN B, bridge 104, LAN E, bridge 107, LAN A.

To see the problem created by a closed loop, consider Figure 13.17. At time t_0, station A transmits a frame addressed to station B. The frame is captured by both bridges. Each bridge updates its database to indicate that station A is in the direction of LAN X, and retransmits the frame on LAN Y. Say that bridge α retransmits at time t_1 and bridge β a short time later t_2. Thus B will receive two copies of the frame. Furthermore, each bridge will receive the other's transmission on LAN Y. Note that each transmission is a frame with a source address of A and a destination address of B. Thus each bridge will update its database to indicate that station A is in the direction of LAN Y. Neither bridge is now capable of forwarding a frame addressed to station A.

To overcome this problem, a simple result from graph theory is used: For any connected graph, consisting of nodes and edges connecting pairs of nodes, there is a spanning tree of edges that maintains the connectivity of the graph but contains no closed loops. In terms of internets, each LAN corresponds to a graph node, and each bridge corresponds to a graph edge. Thus, in Figure 13.16, the removal of one (and only one) of bridges 107, 101, and 104, results in a spanning tree. What is desired is to develop a simple algorithm by which the bridges of the internet can exchange sufficient information to automatically (without user intervention) derive a spanning tree. The algorithm must be dynamic. That is, when a topology change occurs, the bridges must be able to discover this fact and automatically derive a new spanning tree.

Network System Technologies Ex. 2010
Samsung v. Network System Technologies - IPR 2024-00338

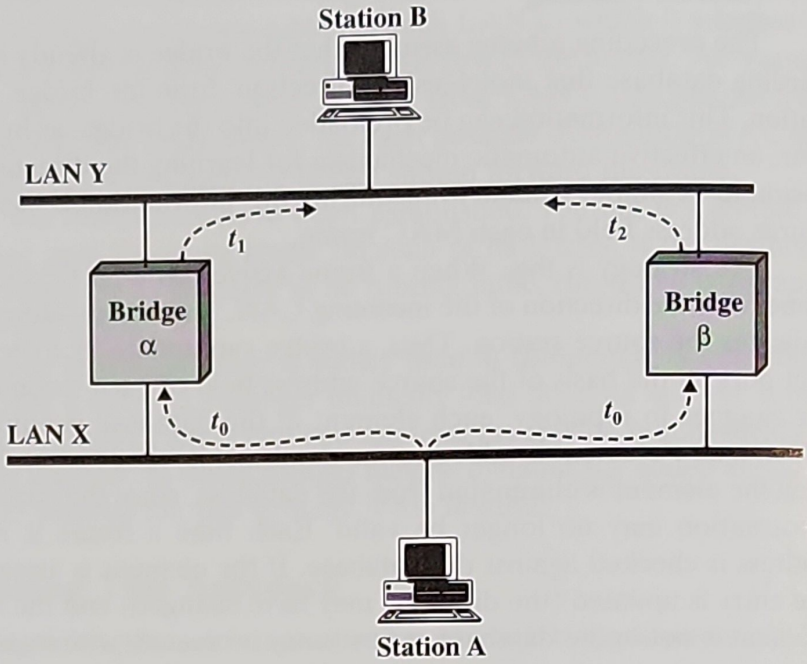


Figure 13.17 Loop of Bridges

The spanning tree algorithm developed by IEEE 802.1, as the name suggests, is able to develop such a spanning tree. All that is required is that each bridge be assigned a unique identifier and that costs be assigned to each bridge port. In the absence of any special considerations, all costs could be set equal; this produce a minimum-hop tree. The algorithm involves a brief exchange of messages among all of the bridges to discover the minimum-cost spanning tree. Whenever there is a change in topology, the bridges automatically recalculate the spanning tree.

13.8 RECOMMENDED READING AND WEB SITES

The literature on LANs and MANs is vast. The material in this chapter is covered in much more depth in [STAL00]. [NAUG96] is a useful survey, while [TRUL97] provides detailed coverage of LAN wiring.

[PAHL95] and [BANT94] are excellent survey articles on wireless LANs. [KAHN97] provides good coverage of infrared LANs. [GEIR99] is an excellent book-length treatment, with detailed coverage of the technology, the IEEE 802.11 standards, and numerous case studies.

BANT94 Bantz, D., and Bauchot, F. "Wireless LAN Design Alternatives." *IEEE Network*, March/April, 1994.

GEIE99 Geier, J. *Wireless LANs*. New York: Macmillan Technical Publishing, 1999.

KAHN97 Kahn, J., and Barry, J. "Wireless Infrared Communications." *Proceedings of the IEEE*, February 1997.

NAUG96 Naugle, M. *Local Area Networking*. New York: McGraw-Hill, 1996.

Network System Technologies Ex. 2010
Samsung v. Network System Technologies - IPR 2024-00338