Part Three

Wide Area Networks

Part Two dealt with the transfer of data between devices that are directly connected, generally by a point-to-point link. Often, however, this arrangement is impractical, and a data communication network is required to transmit data between devices, either because the devices are very far apart or because there are many devices to be interconnected. In general terms, communications networks can be categorized as wide area networks (WANs) and local area networks (LANs). Part Three focuses on WANs, while Part Four covers LANs.

Two perspectives on the material in this part are of particular significance as they relate to the treatment of internetworking in Part Five. First, the constituent networks of the Internet and other internetworks are LANs and WANs. Thus, a full understanding of the technology and architecture of internetworks includes an understanding of the underlying networks. Second, and perhaps more important, many of the technologies developed for switched data WANs (including packet-switching, frame relay, and ATM networks) carry over into the design of internetworks. This is especially true in the cases of routing and congestion control.

ROAD MAP FOR PART THREE

Chapter 10 Circuit Switching and Packet Switching

Our treatment of the technology and architecture of circuit-switching networks begins with the internal operation of a single switch. This is in contrast to packet-switching networks, which are best explained by the collective behavior of the set of switches that make up a network. Thus, Chapter 10 begins by examining digital switching concepts, including space and time division switching.

The remainder of Chapter 10 introduces packet-switching technology. The chapter covers the basic principles of packet switching and analyzes datagram and virtual circuit approaches. The chapter also covers frame relay networks.

Chapter 11 Asynchronous Transfer Mode

Chapter 11 focuses on the transmission technology that is the foundation of broadband ISDN: asynchronous transfer mode (ATM). ATM is also finding widespread application beyond its use as part of broadband ISDN. ATM is in essence a packet-switching technology, but it is far more streamlined and efficient than traditional packet switching and is designed to support very high data rates. This chapter begins with a description of the ATM protocol and format. Then the physical layer issues relating to the transmission of ATM cells are discussed.

Chapter 12 Routing in Switched Networks

One significant technical issue associated with switched networks is routing. Because the source and destination nodes are not directly connected, the network must route each packet, from node to node, through the network. Chapter 12 discusses the common routing algorithms used both in switched data networks, such as frame relay and ATM, and in the Internet.

Chapter 13 Congestion Control in Data Networks

A critical design issue for switched data networks and the Internet is congestion control. Chapter 13 begins with an explanation of the nature of congestion in switched networks and both the importance and difficulty of controlling congestion. The chapter provides a general discussion of congestion control in traditional packet-switching networks and also examines frame relay congestion control. The focus of the remainder of the chapter is on congestion and traffic control for ATM networks. This is one of the most complex aspects of ATM and is the subject of intensive ongoing research. This chapter surveys those techniques that have been accepted as having broad utility in ATM environments. All of the techniques surveyed in this chapter also have relevance in the context of the Internet.

Chapter 14: Cellular Wireless Networks

Chapter 14 begins with a discussion of the important design issues related to cellular wireless networks. Next, the chapter covers the traditional mobile telephony service, now known as first-generation analog. Chapter 14 then examines second-generation digital cellular networks. Finally, an overview of third-generation networks is provided.

CHAPTER 10

Circuit Switching and Packet Switching

10.1 Switched Communications Networks

10.2 Circuit-Switching Networks

10.3 Circuit-Switching Concepts

10.4 Softswitch Architecture

10.5 Packet-Switching Principles

10.6 X.25

10.7 Frame Relay

10.8 Recommended Reading and Web Sites

10.9 Key Terms, Review Questions, and Problems

He got into a District Line train at Wimbledon Park, changed on to the Victoria Line at Victoria and on to the Jubilee Line at Green Park for West Hampstead. It was a long and awkward journey but he enjoyed it.

—*King Solomon's Carpet*, Barbara Vine (Ruth Rendell)

KEY POINTS

- 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.
- Packet switching was designed to provide a more efficient facility than circuit switching for bursty data traffic. With packet switching, a station transmits data in small blocks, called packets. Each packet contains some portion of the user data plus control information needed for proper functioning of the network.
- A key distinguishing element of packet-switching networks is whether the internal operation is datagram or virtual circuit. With internal virtual circuits, a route is defined between two endpoints and all packets for that virtual circuit follow the same route. With internal datagrams, each packet is treated independently, and packets intended for the same destination may follow different routes.
- X.25 is the standard protocol for the interface between an end system and a packet-switching network.
- Frame relay is a form of packet switching that provides a streamlined interface compared to X.25, with improved performance.

Part Two describes how information can be encoded and transmitted over a communications link. We now turn to the broader discussion of networks, which can be used to interconnect many devices. The chapter begins with a general discussion of switched communications networks. The remainder of the chapter focuses on wide area networks and, in particular, on traditional approaches to wide area network design: circuit switching and packet switching.

Since the invention of the telephone, circuit switching has been the dominant technology for voice communications, and it has remained so well into the digital era. This chapter looks at the key characteristics of a circuit-switching network.

Around 1970, research began on a new form of architecture for long-distance digital data communications: packet switching. Although the technology of packet switching has evolved substantially since that time, it is remarkable that (1) the basic technology of packet switching is fundamentally the same today as it was in the early 1970s networks, and (2) packet switching remains one of the few effective technologies for long-distance data communications.

This chapter provides an overview of packet-switching technology. We will see, in this chapter and later in this part, that many of the advantages of packet switching (flexibility, resource sharing, robustness, responsiveness) come with a cost. The packet-switching network is a distributed collection of packet-switching nodes. Ideally, all packet-switching nodes would always know the state of the entire network. Unfortunately, because the nodes are distributed, there is a time delay between a change in status in one portion of the network and knowledge of that change elsewhere. Furthermore, there is overhead involved in communicating status information. As a result, a packet-switching network can never perform "perfectly," and elaborate algorithms are used to cope with the time delay and overhead penalties of network operation. These same issues will appear again when we discuss internetworking in Part Five.

Finally, this chapter provides an overview of a popular form of packet switching known as frame relay.

10.1 SWITCHED COMMUNICATIONS 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 typically 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 10.1 illustrates a simple network. The devices attached to the network may be referred to as *stations*. The stations may be computers, terminals, telephones, or other communicating devices. We refer to the switching devices whose purpose is to provide communication as *nodes*. 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*.

In a *switched communication network*, data entering the network from a station are routed to the destination by being switched from node to node. For

[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).

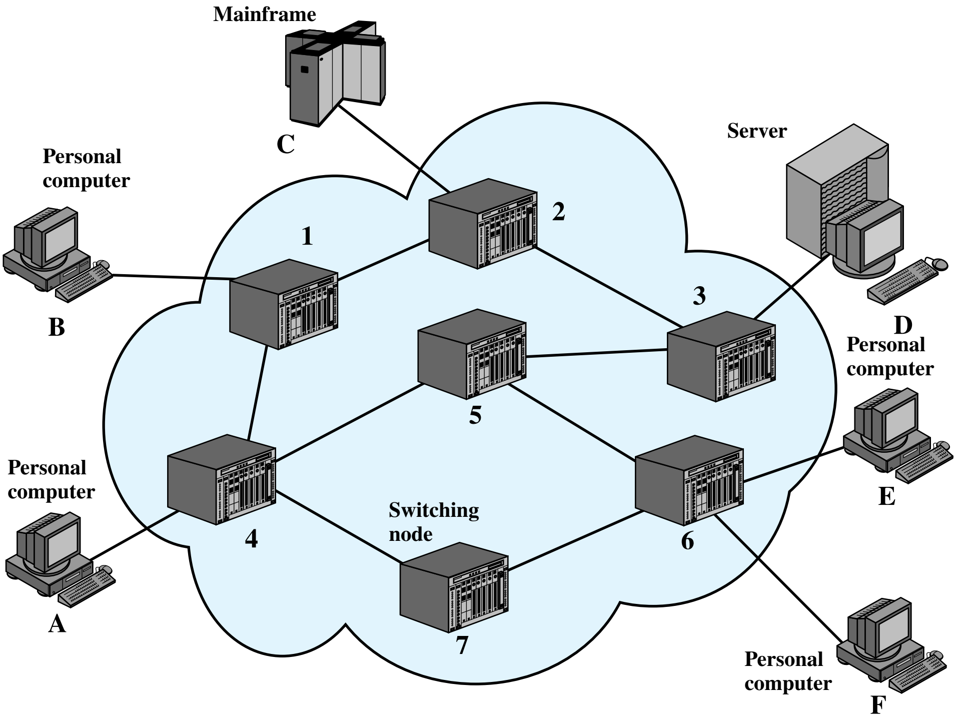


Figure 10.1 Simple Switching Network

example, in Figure 10.1, 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-station links are generally dedicated point-to-point links. 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 different technologies are used in wide area switched networks: circuit switching and packet switching. These two technologies differ in the way the nodes switch information from one link to another on the way from source to destination.

10.2 CIRCUIT-SWITCHING NETWORKS

Communication via circuit switching implies that there is a dedicated communication path between two stations. That path is a connected sequence of links between network nodes. On each physical link, a logical channel is dedicated to the connection. Communication via circuit switching involves three phases, which can be explained with reference to Figure 10.1.

1. **Circuit establishment.** Before any signals can be transmitted, an end-to-end (station-to-station) circuit must be established. For example, station A sends a request to node 4 requesting a connection to station E. Typically, the link from A to 4 is a dedicated line, so that part of the connection already exists. Node 4 must find the next leg in a route leading to E. Based on routing information and measures of availability and perhaps cost, node 4 selects the link to node 5, allocates a free channel (using FDM or TDM) on that link, and sends a message requesting connection to E. So far, a dedicated path has been established from A through 4 to 5. Because a number of stations may attach to 4, it must be able to establish internal paths from multiple stations to multiple nodes. How this is done is discussed later in this section. The remainder of the process proceeds similarly. Node 5 allocates a channel to node 6 and internally ties that channel to the channel from node 4. Node 6 completes the connection to E. In completing the connection, a test is made to determine if E is busy or is prepared to accept the connection.
2. **Data transfer.** Data can now be transmitted from A through the network to E. The transmission may be analog or digital, depending on the nature of the network. As the carriers evolve to fully integrated digital networks, the use of digital (binary) transmission for both voice and data is becoming the dominant method. The path is A-4 link, internal switching through 4, 4-5 channel, internal switching through 5, 5-6 channel, internal switching through 6, 6-E link. Generally, the connection is full duplex.
3. **Circuit disconnect.** After some period of data transfer, the connection is terminated, usually by the action of one of the two stations. Signals must be propagated to nodes 4, 5, and 6 to deallocate the dedicated resources.

Note that the connection path is established before data transmission begins. Thus, channel capacity must be reserved between each pair of nodes in the path, and each node must have available internal switching capacity to handle the requested connection. The switches must have the intelligence to make these allocations and to devise a route through the network.

Circuit switching can be rather inefficient. Channel capacity is dedicated for the duration of a connection, even if no data are being transferred. For a voice connection, utilization may be rather high, but it still does not approach 100%. For a client/server or terminal-to-computer connection, the capacity may be idle during most of the time of the connection. In terms of performance, there is a delay prior to signal transfer for call establishment. However, once the circuit is established, the network is effectively transparent to the users. Information is transmitted at a fixed data rate with no delay other than the propagation delay through the transmission links. The delay at each node is negligible.

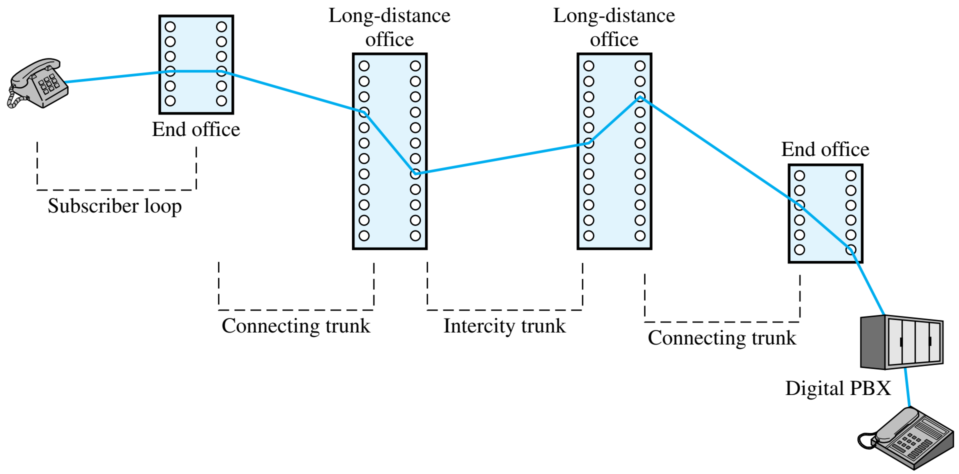


Figure 10.2 Example Connection Over a Public Circuit-Switching Network

Circuit switching was developed to handle voice traffic but is now also used for data traffic. The best-known example of a circuit-switching network is the public telephone network (Figure 10.2). This is actually a collection of national networks interconnected to form the international service. Although originally designed and implemented to service analog telephone subscribers, it handles substantial data traffic via modem and is gradually being converted to a digital network. Another well-known application of circuit switching is the private branch exchange (PBX), used to interconnect telephones within a building or office. Circuit switching is also used in private networks. Typically, such a network is set up by a corporation or other large organization to interconnect its various sites. Such a network usually consists of PBX systems at each site interconnected by dedicated, leased lines obtained from one of the carriers, such as AT&T. A final common example of the application of circuit switching is the data switch. The data switch is similar to the PBX but is designed to interconnect digital data processing devices, such as terminals and computers.

A public telecommunications network can be described using four generic architectural components:

- **Subscribers:** The devices that attach to the network. It is still the case that most subscriber devices to public telecommunications networks are telephones, but the percentage of data traffic increases year by year.
- **Subscriber line:** The link between the subscriber and the network, also referred to as the *subscriber loop* or *local loop.* Almost all local loop connections use twisted-pair wire. The length of a local loop is typically in a range from a few kilometers to a few tens of kilometers.
- **Exchanges:** The switching centers in the network. A switching center that directly supports subscribers is known as an end office. Typically, an end office will support many thousands of subscribers in a localized area. There are over 19,000 end offices in the United States, so it is clearly impractical for each end

office to have a direct link to each of the other end offices; this would require on the order of 2×10^8 links. Rather, intermediate switching nodes are used.

- **Trunks:** The branches between exchanges. Trunks carry multiple voice-frequency circuits using either FDM or synchronous TDM. We referred to these as carrier systems in Chapter 8.

Subscribers connect directly to an end office, which switches traffic between subscribers and between a subscriber and other exchanges. The other exchanges are responsible for routing and switching traffic between end offices. This distinction is shown in Figure 10.3. To connect two subscribers attached to the same end office, a circuit is set up between them in the same fashion as described before. If two subscribers connect to different end offices, a circuit between them consists of a chain of circuits through one or more intermediate offices. In the figure, a connection is established between lines a and b by simply setting up the connection through the end office. The connection between c and d is more complex. In c's end office, a connection is established between line c and one channel on a TDM trunk to the intermediate switch. In the intermediate switch, that channel is connected to a channel on a TDM trunk to d's end office. In that end office, the channel is connected to line d.

Circuit-switching technology has been driven by those applications that handle voice traffic. One of the key requirements for voice traffic is that there must be virtually no transmission delay and certainly no variation in delay. A constant signal transmission rate must be maintained, because transmission and reception occur at the same signal rate. These requirements are necessary to allow normal human conversation. Further, the quality of the received signal must be sufficiently high to provide, at a minimum, intelligibility.

Circuit switching achieved its widespread, dominant position because it is well suited to the analog transmission of voice signals. In today's digital world, its inefficiencies are more apparent. However, despite the inefficiency, circuit switching will

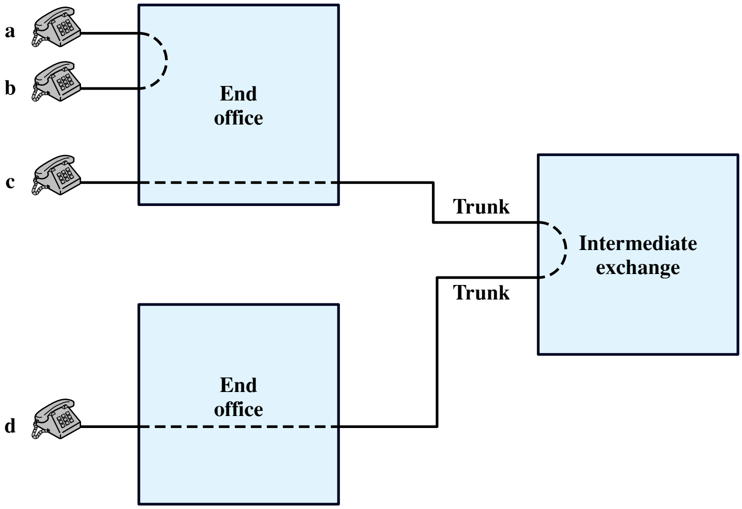


Figure 10.3 Circuit Establishment

remain an attractive choice for both local area and wide area networking. One of its key strengths is that it is transparent. Once a circuit is established, it appears as a direct connection to the two attached stations; no special networking logic is needed at the station.

10.3 CIRCUIT-SWITCHING CONCEPTS

The technology of circuit switching is best approached by examining the operation of a single circuit-switching node. A network built around a single circuit-switching node consists of a collection of stations attached to a central switching unit. The central switch establishes a dedicated path between any two devices that wish to communicate. Figure 10.4 depicts the major elements of such a one-node network. The dotted lines inside the switch symbolize the connections that are currently active.

The heart of a modern system is a **digital switch**. The function of the digital switch is to provide a transparent signal path between any pair of attached devices. The path is transparent in that it appears to the attached pair of devices that there is

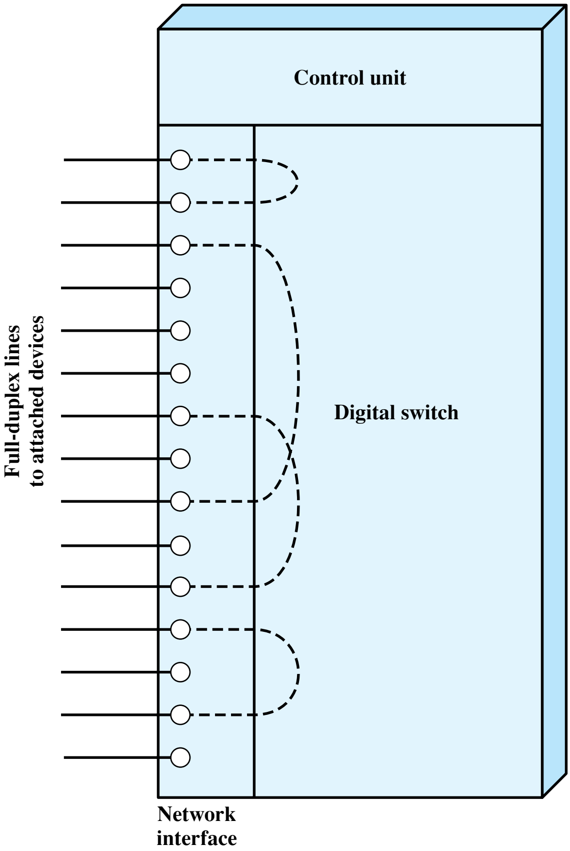


Figure 10.4 Elements of a Circuit-Switch Node

a direct connection between them. Typically, the connection must allow full-duplex transmission.

The **network interface** element represents the functions and hardware needed to connect digital devices, such as data processing devices and digital telephones, to the network. Analog telephones can also be attached if the network interface contains the logic for converting to digital signals. Trunks to other digital switches carry TDM signals and provide the links for constructing multiple-node networks.

The **control unit** performs three general tasks. First, it establishes connections. This is generally done on demand, that is, at the request of an attached device. To establish the connection, the control unit must handle and acknowledge the request, determine if the intended destination is free, and construct a path through the switch. Second, the control unit must maintain the connection. Because the digital switch uses time division principles, this may require ongoing manipulation of the switching elements. However, the bits of the communication are transferred transparently (from the point of view of the attached devices). Third, the control unit must tear down the connection, either in response to a request from one of the parties or for its own reasons.

An important characteristic of a circuit-switching device is whether it is blocking or nonblocking. Blocking occurs when the network is unable to connect two stations because all possible paths between them are already in use. A blocking network is one in which such blocking is possible. Hence a nonblocking network permits all stations to be connected (in pairs) at once and grants all possible connection requests as long as the called party is free. When a network is supporting only voice traffic, a blocking configuration is generally acceptable, because it is expected that most phone calls are of short duration and that therefore only a fraction of the telephones will be engaged at any time. However, when data processing devices are involved, these assumptions may be invalid. For example, for a data entry application, a terminal may be continuously connected to a computer for hours at a time. Hence, for data applications, there is a requirement for a nonblocking or “nearly nonblocking” (very low probability of blocking) configuration.

We turn now to an examination of the switching techniques internal to a single circuit-switching node.

Space Division Switching

Space division switching was originally developed for the analog environment and has been carried over into the digital realm. The fundamental principles are the same, whether the switch is used to carry analog or digital signals. As its name implies, a space division switch is one in which the signal paths are physically separate from one another (divided in space). Each connection requires the establishment of a physical path through the switch that is dedicated solely to the transfer of signals between the two endpoints. The basic building block of the switch is a metallic crosspoint or semiconductor gate that can be enabled and disabled by a control unit.

Figure 10.5 shows a simple crossbar matrix with 10 full-duplex I/O lines. The matrix has 10 inputs and 10 outputs; each station attaches to the matrix via one input and one output line. Interconnection is possible between any two lines by enabling the appropriate crosspoint. Note that a total of 100 crosspoints is required. The crossbar switch has a number of limitations:

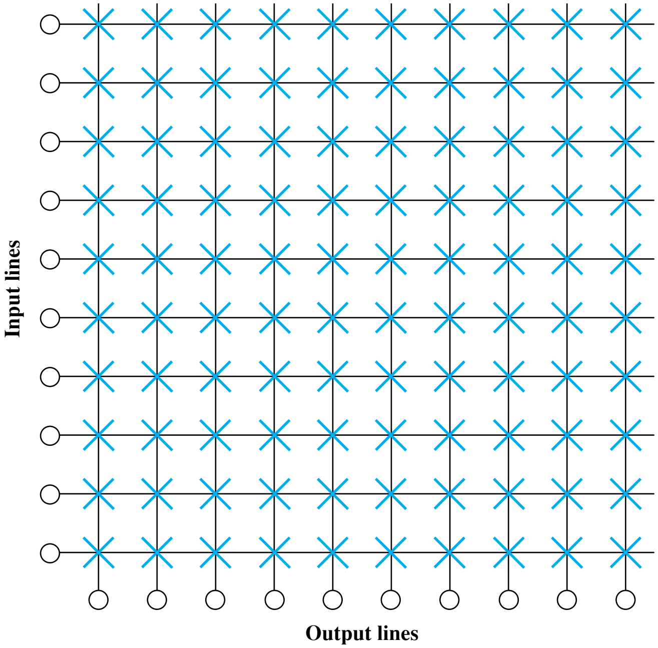


Figure 10.5 Space Division Switch

- The number of crosspoints grows with the square of the number of attached stations. This is costly for a large switch.
- The loss of a crosspoint prevents connection between the two devices whose lines intersect at that crosspoint.
- The crosspoints are inefficiently utilized; even when all of the attached devices are active, only a small fraction of the crosspoints are engaged.

To overcome these limitations, multiple-stage switches are employed. Figure 10.6 is an example of a three-stage switch. This type of arrangement has two advantages over a single-stage crossbar matrix:

- The number of crosspoints is reduced, increasing crossbar utilization. In this example, the total number of crosspoints for 10 stations is reduced from 100 to 48.
- There is more than one path through the network to connect two endpoints, increasing reliability.

Of course, a multistage network requires a more complex control scheme. To establish a path in a single-stage network, it is only necessary to enable a single gate. In a multistage network, a free path through the stages must be determined and the appropriate gates enabled.

A consideration with a multistage space division switch is that it may be blocking. It should be clear from Figure 10.5 that a single-stage crossbar matrix is nonblocking; that is, a path is always available to connect an input to an output. That this may not be the case with a multiple-stage switch can be seen in Figure 10.6. The heavier lines indicate the lines that are already in use. In this state, input line 10, for example, cannot be connected to output line 3, 4, or 5, even though all of these output lines are available. A

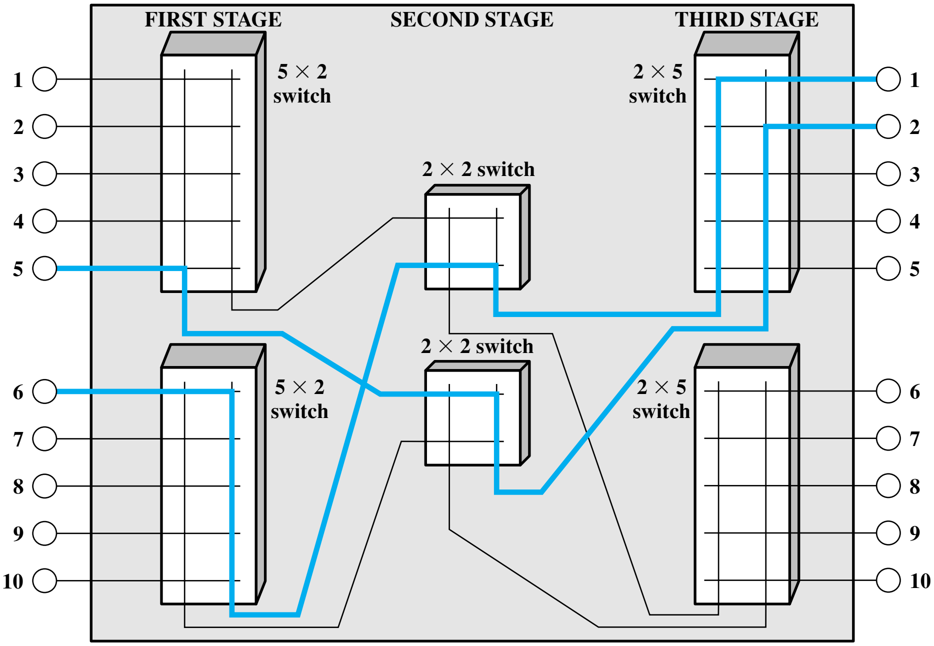


Figure 10.6 Three-Stage Space Division Switch

multiple-stage switch can be made nonblocking by increasing the number or size of the intermediate switches, but of course this increases the cost.

Time Division Switching

The technology of switching has a long history, most of it covering an era when analog signal switching predominated. With the advent of digitized voice and synchronous time division multiplexing techniques, both voice and data can be transmitted via digital signals. This has led to a fundamental change in the design and technology of switching systems. Instead of relatively dumb space division systems, modern digital systems rely on intelligent control of space and time division elements.

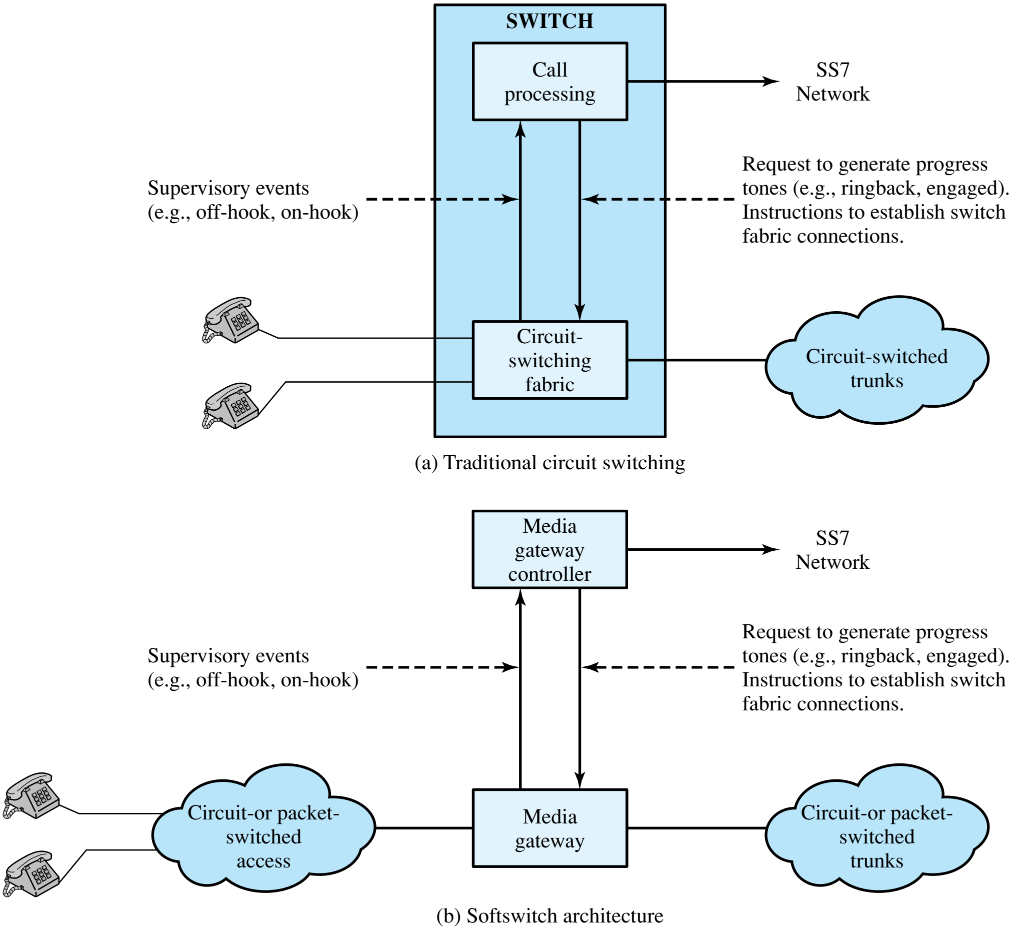
Virtually all modern circuit switches use digital time division techniques for establishing and maintaining "circuits." Time division switching involves the partitioning of a lower-speed bit stream into pieces that share a higher-speed stream with other bit streams. The individual pieces, or slots, are manipulated by control logic to route data from input to output. There are a number of variations on this basic concept, which are beyond the scope of this book

10.4 SOFTSWITCH ARCHITECTURE

The latest trend in the development of circuit-switching technology is generally referred to as the softswitch. In essence, a softswitch is a general-purpose computer running specialized software that turns it into a smart phone switch. Softswitches cost significantly less than traditional circuit switches and can provide more functionality. In particular, in addition to handling the traditional circuit-switching functions, a softswitch can convert a stream of digitized voice bits into packets. This opens up a number of options for transmission, including the increasingly popular voice over IP (Internet Protocol) approach.

In any telephone network switch, the most complex element is the software that controls call processing. This software performs call routing and implements call-processing logic for hundreds of custom calling features. Typically, this software runs on a proprietary processor that is integrated with the physical circuit-switching hardware. A more flexible approach is to physically separate the call processing function from the hardware switching function. In softswitch terminology, the physical switching function is performed by a **media gateway** (MG) and the call processing logic resides in a **media gateway controller** (MGC).

Figure 10.7 contrasts the architecture of a traditional telephone network circuit switch with the softswitch architecture. In the latter case, the MG and MGC are distinct entities and may be provided by different vendors. To facilitate interoperability, two Internet standards have been issued for a media gateway control protocol between the MG and MGC: RFC 2805 (*Media Gateway Control Protocol Architecture*


SWITCH
Call processing
SS7 Network
Supervisory events (e.g., off-hook, on-hook)
Request to generate progress tones (e.g., ringback, engaged). Instructions to establish switch fabric connections.
Circuit-switching fabric
Circuit-switched trunks
Media gateway controller
SS7 Network
Supervisory events (e.g., off-hook, on-hook)
Request to generate progress tones (e.g., ringback, engaged). Instructions to establish switch fabric connections.
Circuit-or packet-switched access
Media gateway
Circuit-or packet-switched trunks

(a) Traditional circuit switching

(b) Softswitch architecture

Figure 10.7 Comparison between Traditional Circuit Switching and Softswitch

and Requirements) and RFC 3525 (*Gateway Control Protocol Version 1*). Softswitch functionality is also defined in the H series or ITU-T Recommendations, which covers audiovisual and multimedia systems.

10.5 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 10.8). 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 10.1, 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

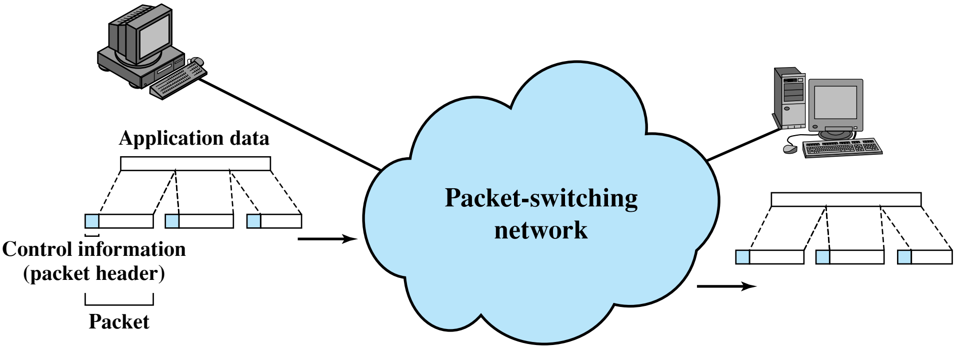


Figure 10.8 The Use of Packets

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:

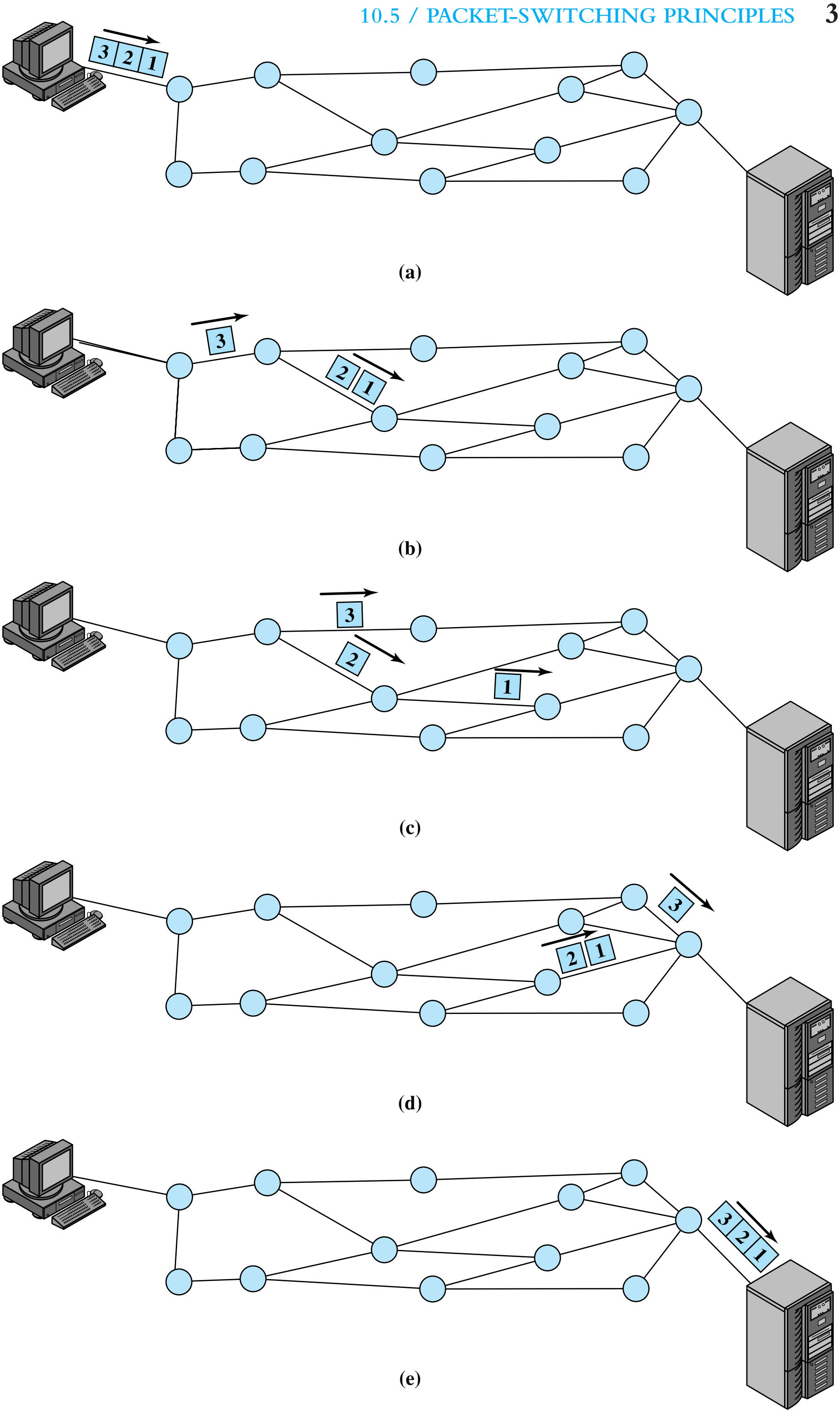
- Line efficiency is greater, because a single node-to-node link can be dynamically shared by many packets over time. The packets are queued up and transmitted as rapidly as possible over the link. By contrast, with circuit switching, time on a node-to-node link is preallocated using synchronous time 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.
- 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. 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. Two approaches 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. This approach is illustrated in Figure 10.9, which shows a time sequence of snapshots of the progress of three packets through the network. Each node chooses the next node on a packet's path, taking into account information received from neighboring nodes on traffic, line failures, and so on. So the packets, each with the same destination address, do not all follow the same route, and they may arrive out of sequence at the exit point. In this example, the exit node restores the packets to their original order before delivering them to the destination. In some datagram networks, it is up to the destination rather than the exit node to do the reordering. 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. Again, it is up to either the exit node or the destination to detect the loss of a packet and decide 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. Once the route is established, all the packets between a pair of communicating parties follow this same route through the network. This is illustrated in Figure 10.10. 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



(a)

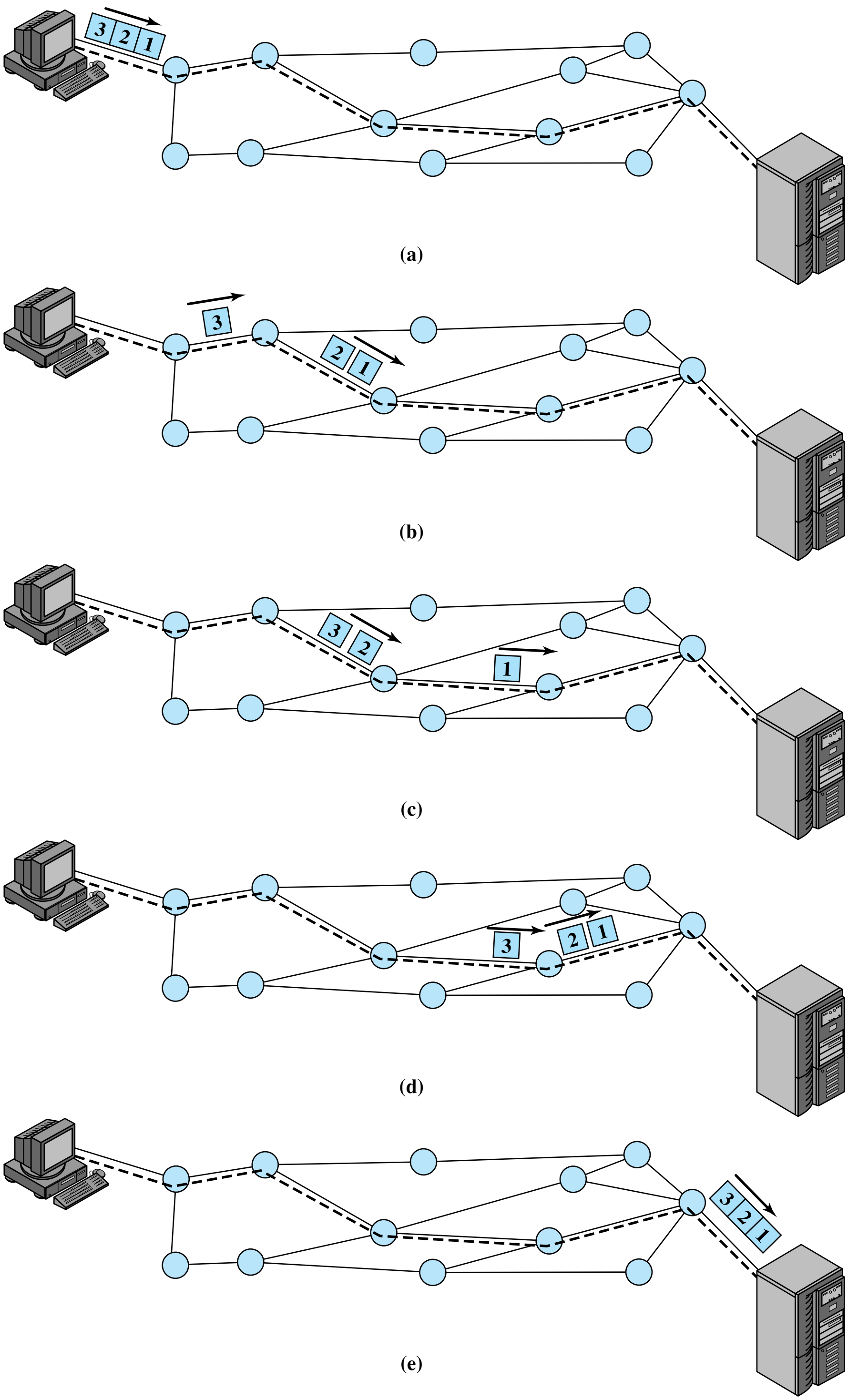
(b)

(c)

(d)

(e)

Figure 10.9 Packet Switching: Datagram Approach



(a)

(b)

(c)

(d)

(e)

Figure 10.10 Packet Switching: Virtual-Circuit Approach

virtual circuit. Each packet contains a virtual circuit identifier as well as data. Each node on the preestablished route knows where to direct such packets; no routing decisions are required. 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 transmitted packet is 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 also 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, 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. 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 10.11. 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 10.11a). 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 × 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.

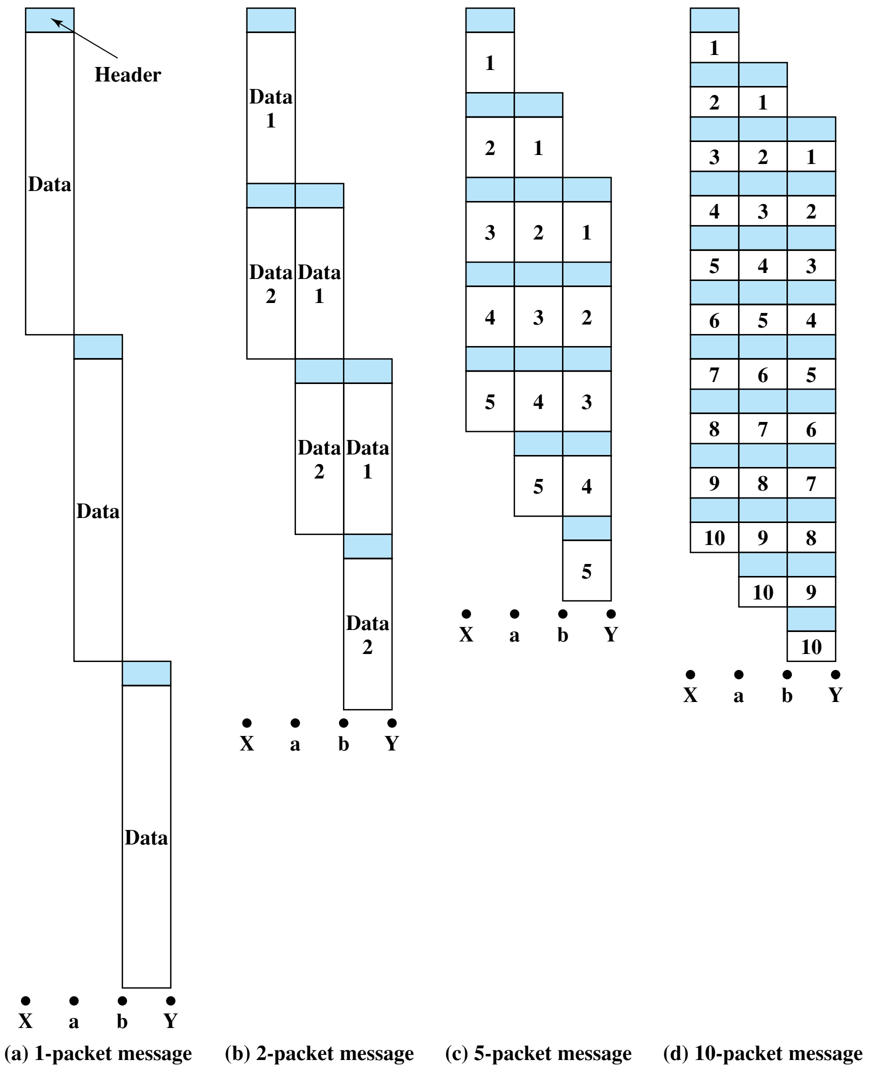


Figure 10.11 Effect of Packet Size on Transmission Time

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 10.11d. 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.

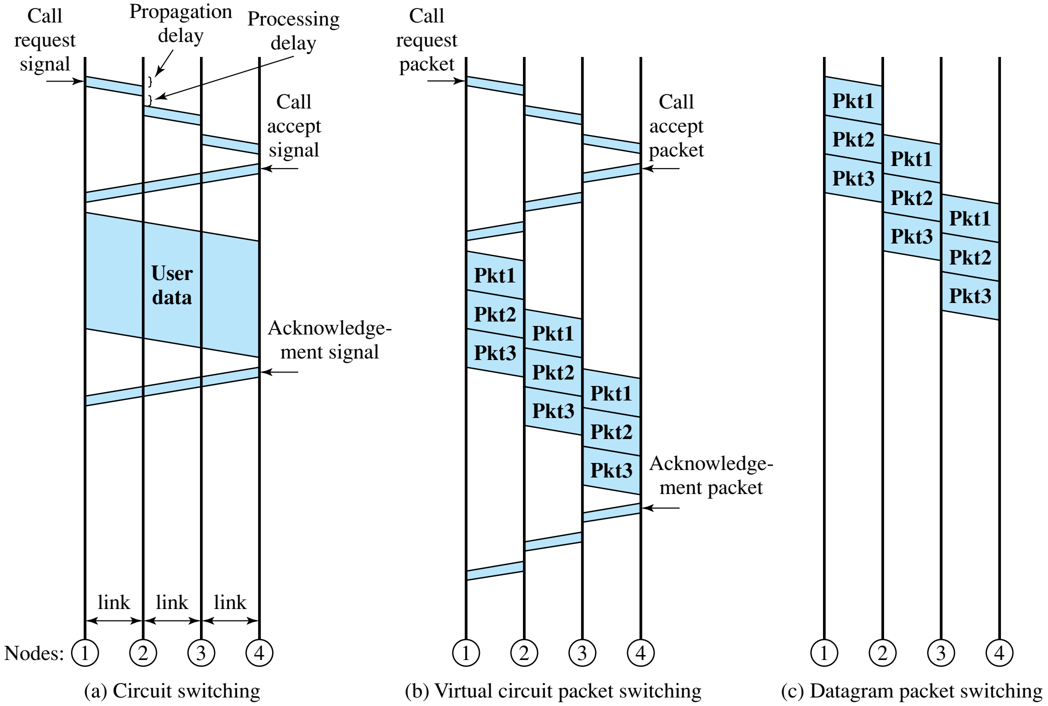
Comparison of Circuit Switching and Packet Switching

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 10.12. The figure depicts the transmission of a message 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.
- **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.



(a) Circuit switching (b) Virtual circuit packet switching (c) Datagram packet switching

Figure 10.12 Event Timing for Circuit Switching and Packet Switching

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 circuit 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 10.12 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.

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 nodes	Small switching 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

10.6 X.25

One technical aspect of packet-switching networks remains to be examined: the interface between attached devices and the network. We have seen that a circuit-switching network provides a transparent communications path for attached devices that makes it appear that the two communicating stations have a direct link. However, in the case of packet-switching networks, the attached stations must organize their data into packets for transmission. This requires a certain level of cooperation between the network and the attached stations. This cooperation is embodied in an interface standard. The standard used for traditional packet-switching networks is X.25.

X.25 is an ITU-T standard that specifies an interface between a host system and a packet-switching network. The functionality of X.25 is specified on three levels:

- Physical level
- Link level
- Packet level

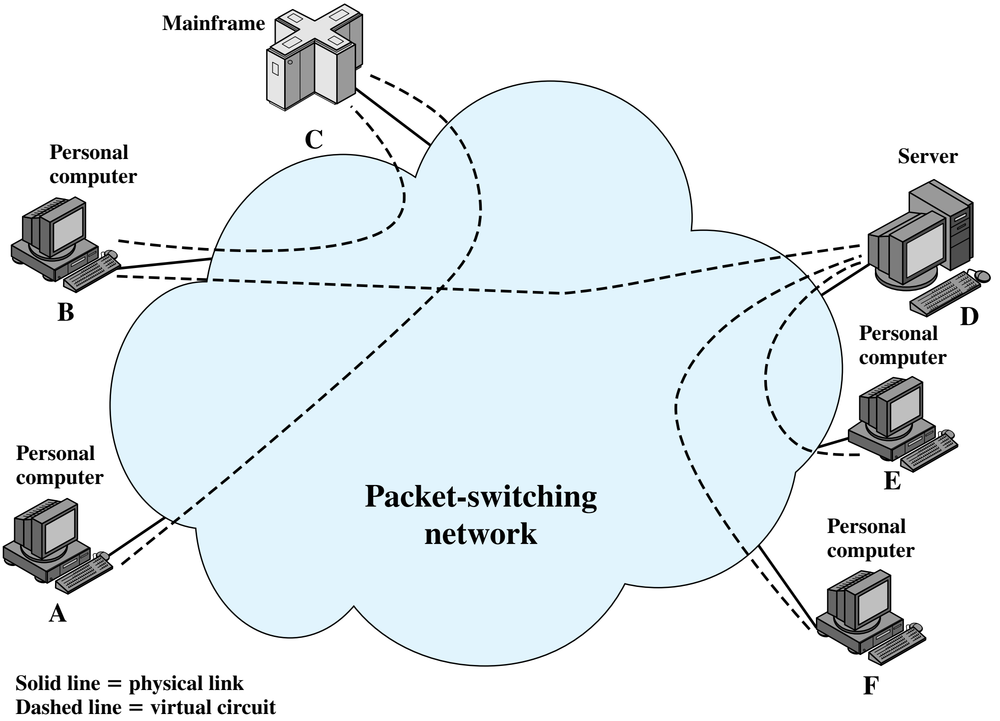


Figure 10.13 The Use of Virtual Circuits

The physical level deals with the physical interface between an attached station (computer, terminal) and the link that attaches that station to the packet-switching node. It makes use of the physical-level specification in a standard known as X.21, but in many cases other standards, such as EIA-232, are substituted. The link level provides for the reliable transfer of data across the physical link, by transmitting the data as a sequence of frames. The link level standard is referred to as LAPB (Link Access Protocol–Balanced). LAPB is a subset of HDLC, which was described in Chapter 7.

The packet level provides a virtual circuit service. This service enables any subscriber to the network to set up logical connections, called virtual circuits, to other subscribers. An example is shown in Figure 10.13 (compare Figure 10.1). In this example, station A has a virtual circuit connection to C; station B has two virtual circuits established, one to C and one to D; and stations E and F each have a virtual circuit connection to D.

In this context, the term *virtual circuit* refers to the logical connection between two stations through the network; this is perhaps best termed an external virtual circuit. Earlier, we used the term *virtual circuit* to refer to a specific preplanned route through the network between two stations; this could be called an internal virtual circuit. Typically, there is a one-to-one relationship between external and internal virtual circuits. However, it is also possible to employ X.25 with a datagram-style network. What is important for an external virtual circuit is that there is a logical relationship, or logical channel, established between two stations, and all of the data

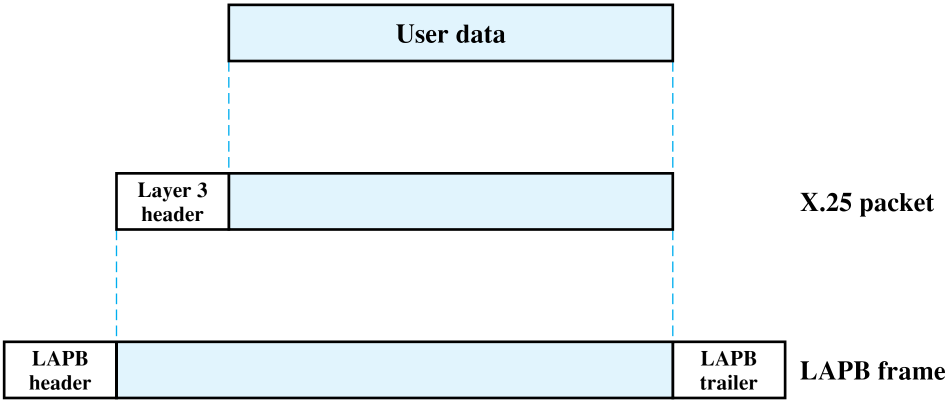


Figure 10.14 User Data and X.25 Protocol Control Information

associated with that logical channel are considered as part of a single stream of data between the two stations. For example, in Figure 10.13, station D keeps track of data packets arriving from three different workstations (B, E, F) on the basis of the virtual circuit number associated with each incoming packet.

Figure 10.14 illustrates the relationship among the levels of X.25. User data are passed down to X.25 level 3, which appends control information as a header, creating a packet. This control information serves several purposes, including

1. Identifying by number a particular virtual circuit with which this data is to be associated
2. Providing sequence numbers that can be used for flow and error control on a virtual circuit basis

The entire X.25 packet is then passed down to the LAPB entity, which appends control information at the front and back of the packet, forming a LAPB frame (see Figure 7.7). Again, the control information in the frame is needed for the operation of the LAPB protocol.

The operation of the X.25 packet level is similar to that of HDLC as described in Chapter 7. Each X.25 data packet includes send and receive sequence numbers. The send sequence number, P(S), is used to number sequentially all outgoing data packets on a particular virtual circuit. The receive sequence number, P(R), is an acknowledgment of packets received on that virtual circuit.

10.7 FRAME RELAY

Frame relay is designed to provide a more efficient transmission scheme than X.25. The standards for frame relay matured earlier than those for ATM, and commercial products also arrived earlier. Accordingly, there is a large installed base of frame relay products. Interest has since shifted to ATM for high-speed data networking, but because of the remaining popularity of frame relay, we provide a survey in this section.

Background

The traditional approach to packet switching makes use of X.25, which not only determines the user-network interface but also influences the internal design of the network. The following are key features of the X.25 approach:

- Call control packets, used for setting up and clearing virtual circuits, are carried on the same channel and same virtual circuit as data packets.
- Multiplexing of virtual circuits takes place at layer 3.
- Both layer 2 and layer 3 include flow control and error control mechanisms.

The X.25 approach results in considerable overhead. At each hop through the network, the data link control protocol involves the exchange of a data frame and an acknowledgment frame. Furthermore, at each intermediate node, state tables must be maintained for each virtual circuit to deal with the call management and flow control/error control aspects of the X.25 protocol. All of this overhead may be justified when there is a significant probability of error on any of the links in the network. This approach is not suitable for modern digital communication facilities. Today's networks employ reliable digital transmission technology over high-quality, reliable transmission links, many of which are optical fiber. In addition, with the use of optical fiber and digital transmission, high data rates can be achieved. In this environment, the overhead of X.25 is not only unnecessary but degrades the effective utilization of the available high data rates.

Frame relay is designed to eliminate much of the overhead that X.25 imposes on end user systems and on the packet-switching network. The key differences between frame relay and a conventional X.25 packet-switching service are as follows:

- Call control signaling, which is information needed to set up and manage a connection, is carried on a separate logical connection from user data. Thus, intermediate nodes need not maintain state tables or process messages relating to call control on an individual per-connection basis.
- Multiplexing and switching of logical connections takes place at layer 2 instead of layer 3, eliminating one entire layer of processing.
- There is no hop-by-hop flow control and error control. End-to-end flow control and error control are the responsibility of a higher layer, if they are employed at all.

Thus, with frame relay, a single user data frame is sent from source to destination, and an acknowledgment, generated at a higher layer, may be carried back in a frame. There are no hop-by-hop exchanges of data frames and acknowledgments.

Let us consider the advantages and disadvantages of this approach. The principal potential disadvantage of frame relay, compared to X.25, is that we have lost the ability to do link-by-link flow and error control. (Although frame relay does not provide end-to-end flow and error control, this is easily provided at a higher layer.) In X.25, multiple virtual circuits are carried on a single physical link, and LAPB is available at the link level for providing reliable transmission from the source to the packet-switching network and from the packet-switching network to the destination. In addition, at each hop through the network, the link control protocol can be

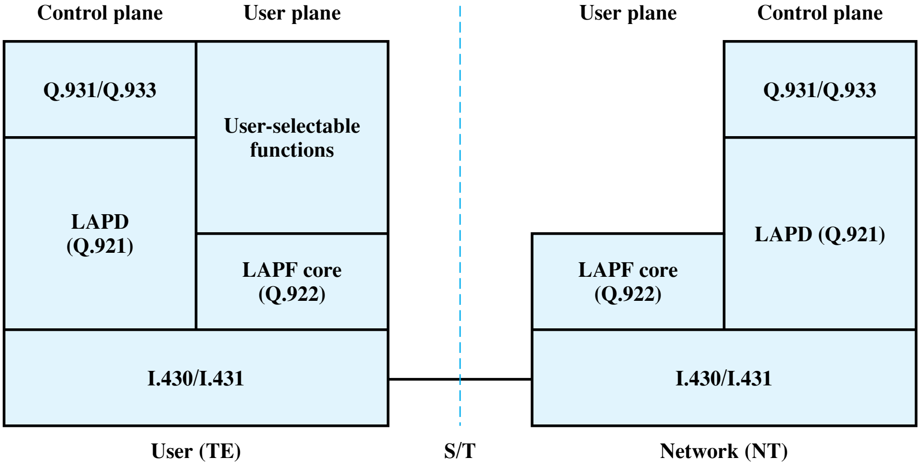


Figure 10.15 Frame Relay User-Network Interface Protocol Architecture

used for reliability. With the use of frame relay, this hop-by-hop link control is lost. However, with the increasing reliability of transmission and switching facilities, this is not a major disadvantage.

The advantage of frame relay is that we have streamlined the communications process. The protocol functionality required at the user-network interface is reduced, as is the internal network processing. As a result, lower delay and higher throughput can be expected. Studies indicate an improvement in throughput using frame relay, compared to X.25, of an order of magnitude or more [HARB92]. The ITU-T Recommendation I.233 indicates that frame relay is to be used at access speeds up to 2 Mbps. However, frame relay service at even higher data rates is now available.

Frame Relay Protocol Architecture

Figure 10.15 depicts the protocol architecture to support the frame mode bearer service. We need to consider two separate planes of operation: a control (C) plane, which is involved in the establishment and termination of logical connections, and a user (U) plane, which is responsible for the transfer of user data between subscribers. Thus, C-plane protocols are between a subscriber and the network, while U-plane protocols provide end-to-end functionality.

Control Plane The control plane for frame mode bearer services is similar to that for common channel signaling for circuit-switching services, in that a separate logical channel is used for control information. At the data link layer, LAPD (Q.921) is used to provide a reliable data link control service, with error control and flow control, between user (TE) and network (NT). This data link service is used for the exchange of Q.933 control signaling messages.

User Plane For the actual transfer of information between end users, the user-plane protocol is LAPF (Link Access Procedure for Frame Mode Bearer Services), which is defined in Q.922. Only the core functions of LAPF are used for frame relay:

- Frame delimiting, alignment, and transparency
- Frame multiplexing/demultiplexing using the address field
- Inspection of the frame to ensure that it consists of an integral number of octets prior to zero bit insertion or following zero bit extraction
- Inspection of the frame to ensure that it is neither too long nor too short
- Detection of transmission errors
- Congestion control functions

The last function listed is new to LAPF. The remaining functions listed are also functions of LAPD.

The core functions of LAPF in the user plane constitute a sublayer of the data link layer. This provides the bare service of transferring data link frames from one subscriber to another, with no flow control or error control. Above this, the user may choose to select additional data link or network-layer end-to-end functions. These are not part of the frame relay service. Based on the core functions, a network offers frame relay as a connection-oriented link layer service with the following properties:

- Preservation of the order of frame transfer from one edge of the network to the other
- A small probability of frame loss

As with X.25, frame relay involves the use of logical connections, in this case called data link connections rather than virtual circuits. The frames transmitted over these data link connections are not protected by a data link control pipe with flow and error control. Another difference between X.25 and frame relay is that the latter devotes a separate data link connection to call control. The setting up and tearing down of data link connections is done over this permanent control-oriented data link connection.

The frame relay architecture significantly reduces the amount of work required of the network. User data are transmitted in frames with virtually no processing by the intermediate network nodes, other than to check for errors and to route based on connection number. A frame in error is simply discarded, leaving error recovery to higher layers.

User Data Transfer

The operation of frame relay for user data transfer is best explained by considering the frame format, illustrated in Figure 10.16a. This is the format defined for the minimum-function LAPF protocol (known as LAPF core protocol). The format is similar to that of LAPD and LAPB with one obvious omission: There is no Control field. This has the following implications:

- There is only one frame type, used for carrying user data. There are no control frames.
- It is not possible to perform all control on the connection; a logical connection can only carry user data.
- It is not possible to perform flow control and error control, because there are no sequence numbers.

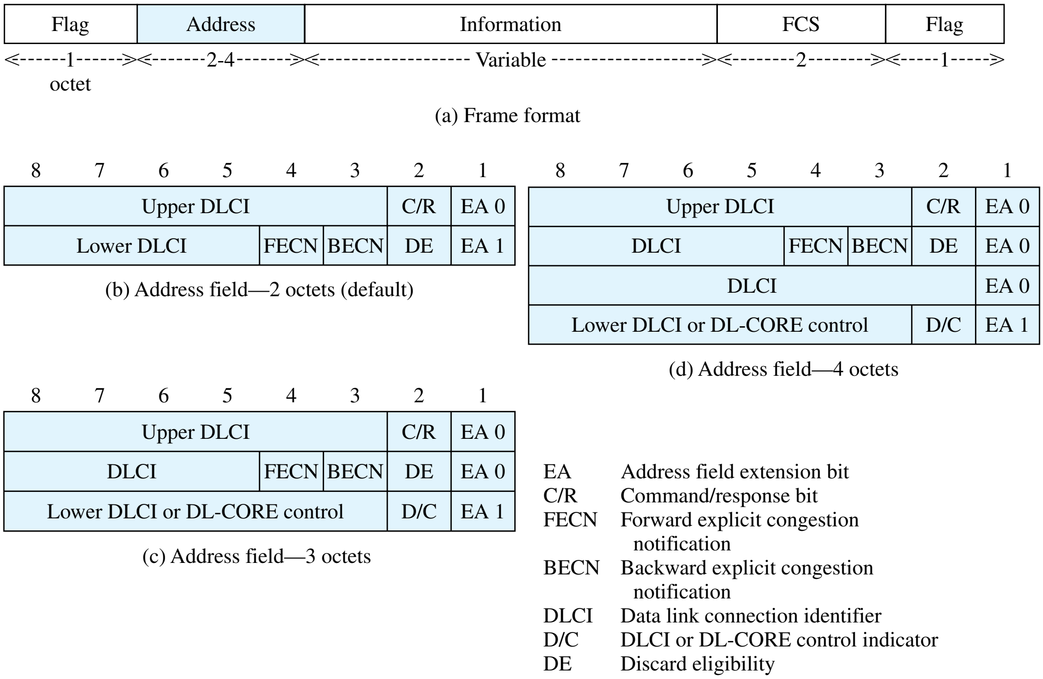

Flag
Address
Information
FCS
Flag
<------1-----><-------2-4------><----------------- Variable -----------------><--------2--------><-----1----->
octet
(a) Frame format
8 7 6 5 4 3 2 1
Upper DLCI
C/R
EA 0
Lower DLCI
FECN
BECN
DE
EA 1
(b) Address field—2 octets (default)
8 7 6 5 4 3 2 1
Upper DLCI
C/R
EA 0
DLCI
FECN
BECN
DE
EA 0
DLCI
EA 0
Lower DLCI or DL-CORE control
D/C
EA 1
(d) Address field—4 octets
8 7 6 5 4 3 2 1
Upper DLCI
C/R
EA 0
DLCI
FECN
BECN
DE
EA 0
Lower DLCI or DL-CORE control
D/C
EA 1
(c) Address field—3 octets
EA Address field extension bit
C/R Command/response bit
FECN Forward explicit congestion notification
BECN Backward explicit congestion notification
DLCI Data link connection identifier
D/C DLCI or DL-CORE control indicator
DE Discard eligibility

Figure 10.16 LAPF-Core Formats

The Flag and Frame Check Sequence (FCS) fields function as in HDLC. The information field carries higher-layer data. If the user selects to implement additional data link control functions end to end, then a data link frame can be carried in this field. Specifically, a common selection will be to use the full LAPF protocol (known as LAPF control protocol), to perform functions above the LAPF core functions. Note that the protocol implemented in this fashion is strictly between the end subscribers and is transparent to the frame relay network.

The address field has a default length of 2 octets and may be extended to 3 or 4 octets. It carries a data link connection identifier (DLCI) of 10, 16, or 23 bits. The DLCI serves the same function as the virtual circuit number in X.25: It allows multiple logical frame relay connections to be multiplexed over a single channel. As in X.25, the connection identifier has only local significance: Each end of the logical connection assigns its own DLCI from the pool of locally unused numbers, and the network must map from one to the other. The alternative, using the same DLCI on both ends, would require some sort of global management of DLCI values.

The length of the Address field, and hence of the DLCI, is determined by the Address field extension (EA) bits. The C/R bit is application specific and not used by the standard frame relay protocol. The remaining bits in the address field have to do with congestion control and are discussed in Chapter 13.

10.8 RECOMMENDED READING AND WEB SITES

As befits its age, circuit switching has inspired a voluminous literature. Two good books on the subject are [BELL00] and [FREE04].

The literature on packet switching is enormous. Books with good treatments of this subject include [BERT92] and [SPRA91]. [ROBE78] is a classic paper on how packet switching technology evolved. [RYBZ80] is a good tutorial on X.25. [BARA02] and [HEGG84] are also interesting.

A more in-depth treatment of frame relay can be found in [STAL99]. An excellent book-length treatment is [BUCK00]. [CHER89] is a good tutorial on frame relay.

BARA02 Baran, P. "The Beginnings of Packet Switching: Some Underlying Concepts." *IEEE Communications Magazine*, July 2002.

BELL00 Bellamy, J. *Digital Telephony.* New York: Wiley, 2000.

BERT92 Bertsekas, D., and Gallager, R. *Data Networks.* Englewood Cliffs, NJ: Prentice Hall, 1992.

BUCK00 Buckwalter, J. *Frame Relay: Technology and Practice.* Reading, MA: Addison-Wesley, 2000.

CHER89 Cherukuri, R., and Derby, R. "Frame Relay: Protocols and Private Network Applications." *Proceedings*, *IEEE INFOCOM '89,* 1989.

FREE04 Freeman, R. *Telecommunication System Engineering.* New York: Wiley, 1996.

HEGG84 Heggestad, H. "An Overview of Packet Switching Communications." *IEEE Communications Magazine*, April 1984.

ROBE78 Roberts, L. "The Evolution of Packet Switching." *Proceedings of the IEEE*, November 1978.

RYBZ80 Rybzzynski, A. "X.25 Interface and End-to-End Virtual Circuit Characteristics." *IEEE Transactions on Communications*, April 1980.

SPRA91 Spragins, J,; Hammond, J.; and Pawlikowski, K. *Telecommunications Protocols and Design.* Reading, MA: Addison-Wesley, 1991.

STAL99 Stallings, W. *ISDN and Broadband ISDN, with Frame Relay and ATM.* Upper Saddle River, NJ: Prentice Hall, 1999.

Recommended Web sites:

- **International Packet Communications Consortium:** News, technical information, and vendor information on softswitch technology and products
- **Media Gateway Control Working Group:** Chartered by IETF to develop the media gateway control protocol and related standards
- **Frame Relay Resource:** Good source of tutorials, service providers, and other links
- **Frame Relay Resource Center:** Good source of information on frame relay

10.9 KEY TERMS, REVIEW QUESTIONS, AND PROBLEMS

Key Terms

circuit switching
circuit-switching network
control signaling
crossbar matrix
datagram
digital switch
exchange
softswitch
space division switching
frame relay
LAPB
LAPF
local loop
media gateway controller
packet switching
subscriber
subscriber line
subscriber loop
time division switching
trunk
virtual circuit
X.25

Review Questions

10.1. Why is it useful to have more than one possible path through a network for each pair of stations?

10.2. What are the four generic architectural components of a public communications network? Define each term.

10.3. What is the principal application that has driven the design of circuit-switching networks?

10.4. What are the advantages of packet switching compared to circuit switching?

10.5. Explain the difference between datagram and virtual circuit operation.

10.6. What is the significance of packet size in a packet-switching network?

10.7. What types of delay are significant in assessing the performance of a packet-switching network?

10.8. How does frame relay differ from X.25?

10.9. What are the relative advantages and disadvantages of frame relay compared to X.25?

Problems

10.1 Consider a simple telephone network consisting of two end offices and one intermediate switch with a 1-MHz full-duplex trunk between each end office and the intermediate switch. Assume a 4-kHz channel for each voice call. The average telephone is used to make four calls per 8-hour workday, with a mean call duration of six minutes. Ten percent of the calls are long distance. What is the maximum number of telephones an end office can support?

10.2 **a.** If a crossbar matrix has n input lines and m output lines, how many crosspoints are required?

b. How many crosspoints would be required if there were no distinction between input and output lines (i.e., if any line could be interconnected to any other line serviced by the crossbar)?

c. Show the minimum configuration.

10.3 Consider a three-stage switch such as Figure 10.6. Assume that there are a total of N input lines and N output lines for the overall three-stage switch. If n is the number of input lines to a stage 1 crossbar and the number of output lines to a stage 3 crossbar, then there are N/n stage 1 crossbars and N/n stage 3 crossbars. Assume each stage 1 crossbar has one output line going to each stage 2 crossbar, and each stage 2 crossbar has one output line going to each stage 3 crossbar. For such a configuration it can be shown that, for the switch to be nonblocking, the number of stage 2 crossbar matrices must equal $2n - 1$.

a. What is the total number of crosspoints among all the crossbar switches?

b. For a given value of *N*, the total number of crosspoints depends on the value of *n*. That is, the value depends on how many crossbars are used in the first stage to handle the total number of input lines. Assuming a large number of input lines to each crossbar (large value of *n*), what is the minimum number of crosspoints for a nonblocking configuration as a function of *n?*

c. For a range of *N* from 10^2 to 10^6, plot the number of crosspoints for a single-stage $N \times N$ switch and an optimum three-stage crossbar switch.

10.4 Explain the flaw in the following reasoning: Packet switching requires control and address bits to be added to each packet. This introduces considerable overhead in packet switching. In circuit switching, a transparent circuit is established. No extra bits are needed. Therefore, there is no overhead in circuit switching. Because there is no overhead in circuit switching, line utilization must be more efficient than in packet switching.

10.5 Define the following parameters for a switching network:

N = number of hops between two given end systems
L = message length in bits
B = data rate, in bits per second (bps), on all links
P = fixed packet size, in bits
H = overhead (header) bits per packet
S = call setup time (circuit switching or virtual circuit) in seconds
D = propagation delay per hop in seconds

a. For $N = 4, L = 3200, B = 9600, P = 1024, H = 16, S = 0.2, D = 0.001$, compute the end-to-end delay for circuit switching, virtual circuit packet switching, and datagram packet switching. Assume that there are no acknowledgments. Ignore processing delay at the nodes.

b. Derive general expressions for the three techniques of part (a), taken two at a time (three expressions in all), showing the conditions under which the delays are equal.

10.6 What value of *P*, as a function of *N*, *L*, and *H*, results in minimum end-to-end delay on a datagram network? Assume that *L* is much larger than *P*, and *D* is zero.

10.7 Assuming no malfunction in any of the stations or nodes of a network, is it possible for a packet to be delivered to the wrong destination?

10.8 Flow-control mechanisms are used at both levels 2 and 3 of X.25. Are both necessary, or is this redundant? Explain.

10.9 There is no error-detection mechanism (frame check sequence) in X.25 level three. Isn't this needed to assure that all of the packets are delivered properly?

10.10 In X.25, why is the virtual circuit number used by one station of two communicating stations different from the virtual circuit number used by the other station? After all, it is the same full-duplex virtual circuit.

10.11 Q.933 recommends a procedure for negotiating the sliding-window flow control window, which may take on a value from 1 to 127. The negotiation makes use of a variable *k* that is calculated from the following parameters:

L_d = data frame size in octets
R_u = throughput in bps
T_{td} = end-to-end transit delay in seconds
k = window size (maximum number of outstanding I frames)

The procedure is described as follows:

> The window size should be negotiated as follows. The originating user should calculate k using the above formula substituting maximum end-to-end transit delay and outgoing maximum frame size for T_{td} and L_d, respectively. The SETUP message shall include the link layer protocol parameters, the link layer core parameters, and the end-to-end transit delay information elements. The destination user should calculate its own k using the above formula substituting cumulative end-to-end transit delay and its own outgoing maximum frame size for T_{td} and L_d, respectively. The CONNECT message shall include the link layer core parameters and the end-to-end transit delay information element so that the originating user can adjust its k based on the information conveyed in these information elements. The originating user should calculate k using the above formula, substituting cumulative end-to-end transit delay and incoming maximum frame size for T_{td} and L_d, respectively.

SETUP and CONNECT are messages exchanged on a control channel during the setup of a frame relay connection. Suggest a formula for calculating k from the other variables and justify the formula.