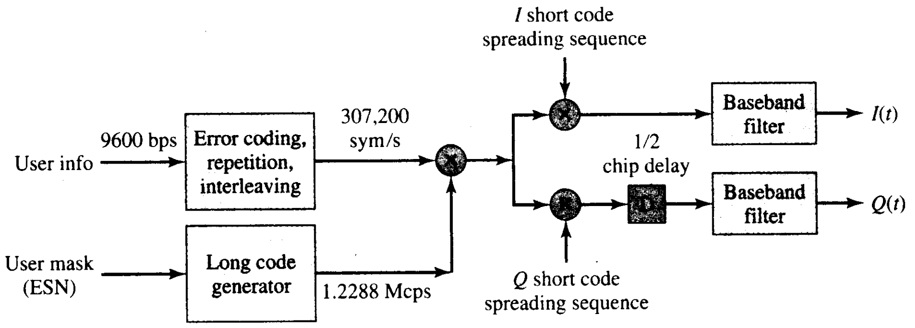


FIGURE 6.39 IS-95 modulator for reverse channel.

The all-zeros Walsh sequence is not used as a traffic channel but is reserved to produce a pilot channel as shown in Figure 6.38b. The pilot signal is received by all mobile stations, and it is used to recover the timing information required to demodulate the received signal. Note in particular that the pilot signal enables the receiver to synchronize to the short code sequence of the signal that arrives from a base station. A mobile station can also detect the pilot signal from more than one base station and then can compare these signals and decide to initiate a **soft handoff** procedure during which the mobile station can receive and transmit to two base stations simultaneously while moving from one cell to another.

Once a mobile station has synchronized to the short code spreading sequence, the station can synchronize to the phase of the carrier to recover the Walsh spread sequence. The correlator detector introduced in the Section 6.4.3 then produces the scrambled information sequence, which is then descrambled to produce the original 9600 bps information signal. IS-95 can accommodate user bit rates of 4800, 2400, and 1200 bps by simply repeating a user information bit several times. For example, a 4800 bps rate is handled by repeating each information bit twice and feeding the resulting 9600 bps stream into the system. An option for bit rates in the set {14400, 7200, 3600, and 1800} bps is available by changing the type of error-correction coding.

In the forward channel the pilot signal is "affordable" because the synchronization required for orthogonal spreading (and channelization) is possible at the base station and because it greatly simplifies the job of the mobile receivers. The situation is different in the reverse channel. Here it is not feasible to synchronize the transmissions of the many mobile stations, so orthogonal spreading is not possible. Consequently, the more conventional spread spectrum transmission technique based on nonorthogonal spreading sequences is implemented in the reverse channel.

As shown in Figure 6.39 the transmitter in the mobile station takes a basic 9600 bps user information sequence and applies error-correction coding, interleaving, and modulation to produce a 307,200 symbol/second sequence that consists of +1s and −1s.[10] This sequence is spread by a factor of 4 by multiplying it by the 1.2288 Msymbol/second sequence produced by the station's long code spreading sequence. This sequence

[10] A confusing point here is that the coding/modulation uses a code that maps blocks of six binary symbols into strings of 64-symbol Walsh sequences. The role of the Walsh sequences here is to facilitate "noncoherent" detection, *not* to produce orthogonal channels. See [Viterbi 1995, Section 4.5].

is subsequently multiplied by the short code sequence that is common to all stations in the cell and then modulated using QPSK. The base station detects the spread spectrum signals from its various mobile stations in the usual manner. Note that, in principle, the reverse channel can produce up to $2^{42} - 1$ code channels, one for each of its possible phases. In fact at most 63 such channels are active in a cell at any given time.

The CDMA approach to cellular communications is clearly very different from either TDMA or FDMA. It did not fit the conventional mode of doing things, and so not surprisingly the calculation of spectrum efficiency proved to be quite controversial. The first major difference is that CDMA can operate with a frequency reuse factor of 1. Recall that TDMA and FDMA operate with a reuse factor of 7; that is, only one-seventh of the channels can be used in a given cell. This reuse factor is required to control the amount of interference between stations in different cells. The use of spread spectrum transmission greatly reduces the severity of intercell interference. The signals that arrive at any base station, whether from mobile stations from its cell or elsewhere, are uncorrelated because the associated transmitters use different long code spreading sequences. The signals that arrive at a mobile station from different base stations are also uncorrelated, since they all use the same short code sequence but with different phase. These features lead to the frequency reuse factor of 1.

An additional factor that contributes to the efficiency of CDMA is its ability to exploit variations in the activity of the users. For example, silence intervals in speech are exploited by reducing the bit rate, say, from 9600 bps to 1200 bps. In effect this step increases the spreading factor G by a factor of 8, so the transmitted power can be reduced. The lower power in turn reduces the interference that is caused to other receivers and thus makes it possible to handle more calls. [Goodman 1997] develops a simplified analysis of spectrum efficiency and arrives at the following bounds:

$$12.1 \text{ calls/cell/MHz} < \text{spectrum capacity of IS-95} < 45.1 \text{ calls/cell/MHz} \qquad (6.29)$$

Even the lower bound is substantially larger than the spectrum efficiencies of IS-54 or GSM. IS-95, IS-54, and GSM are considered examples of second-generation cellular systems. The third-generation cellular system will provide higher bit rates and support a broader range of services. CDMA has been selected as the technology for third-generation systems.

1G, 2G, 3G, 2.5G . . .

Cellular telephone networks evolved quickly from first-generation systems (AMPS) to the current second-generation systems such as GSM, IS-136, and IS-95. The auctioning of new spectrum for wireless services in Europe and the rapid growth of Internet-based services led to unrealistic expectations for a rapid emergence of 3G systems. It has become apparent that the transition to 3G will be gradual and that the ideal situation of a single global standard will not materialize. Nevertheless certain trends in technology and service evolution have become quite clear.

First, the wild success of Short Message Service (SMS) made it amply clear that wireless services need to cater more closely to the requirements of data transfer. SMS is the capability of digital cell phones to transfer short text messages up to 160 characters in length. Both inexpensive and convenient, SMS took off first in Asia and Europe and later in North America. The proliferation of PDAs and laptops drive new demand for wireless access to the Internet. In terms of technology, these trends indicate that wireless access must abandon its circuit-switched origins and become *packet-based* to provide the desired degree of flexibility. Wireless access will continue to support voice service and so the next generation of medium access controls will need to provide quality of service to support not only voice and data but also the soon/eventually-to-follow multimedia services.

Second, significant new spectrum has become available for 3G services but this spectrum has been obtained by service providers at very high cost. Spectrum will remain expensive and wireless systems will continue to place a premium on capacity and bandwidth efficiency. It is no surprise then that all of the contenders for 3G wireless access standards are based on CDMA. In North America and Korea, the cdma2000 standard uses CDMA on multiple 1.25 MHz carriers and is backward compatible with IS-95. In the rest of the world, two wideband CDMA (W-CDMA) standards operate on 5 MHz carriers. One standard uses direct sequence spreading and operates in FDD mode. The other standard uses TDD mode to allocate slots individually in either the uplink or downlink thus providing more flexibility in allocating the bandwidth to users. A synchronous version of the latter standard is being promoted in China. Clearly we can expect multiple standards to continue to coexist.

◆ 6.5 DELAY PERFORMANCE OF MAC AND CHANNELIZATION SCHEMES[11]

In Chapter 5 we introduced performance models for assessing the delay and throughput performance of statistical multiplexers that allow multiple streams of packets to share a common transmission line. In the previous sections we have introduced a number of MAC protocols and channelization schemes that are used to share a broadcast medium among a community of users. A MAC protocol and a statistical multiplexer are similar in that the purpose of both is to share a transmission resource. A MAC protocol, however, is fundamentally different in the need to coordinate the transmissions into the shared medium by physically separate stations. We have already discussed the impact of delay-bandwidth product on the maximum throughput that can be achieved by a MAC protocol and by a channelization scheme. In this section we present performance models for assessing the frame transfer delay performance.

[11] This section builds on the packet multiplexing material presented in Section 5.7.

We use the modeling framework that was introduced in Section 5.7. In particular we assume that frames arrive according to a Poisson arrival process with rate λ frames/second. We usually assume that frames are of constant length with transmission time X seconds/frame. The performance of a statistical multiplexer for this situation is given by the M/D/1 model, which we will use as a benchmark for the performance of the various MAC and channelization schemes.

6.5.1 Performance of Channelization Techniques with Bursty Traffic

First we compare the delay performance of FDMA, TDMA, and CDMA in the direction from remote stations to a central site. We show that channelization techniques are not effective in dealing with bursty traffic.

We suppose that each of the M stations that is connected to the transmission medium has its own buffer and is modeled as a separate multiplexer. We assume that frames arrive at each station with exponential interarrival times with mean λ/M frames/second. We also assume that the frames are always L bits long and that the time to transmit a frame at the full rate of the medium R bps is $X = L/R$ seconds. We assume that for FDMA the transmission rate available to one station is R/M, and so the transmission time of one frame is MX seconds. For TDMA a station gets access to the entire bandwidth $1/M$ of the time, so its average transmission rate is R/M time units/second. We assume that CDMA transmits at a constant rate and does not exploit variations in the activity of the information. In this case the multiplexer in the CDMA behaves in the same way as the FDMA station. Therefore, we focus on FDMA and TDMA after this point.

Figure 6.40 shows the sequence of transmissions as observed by a single station in

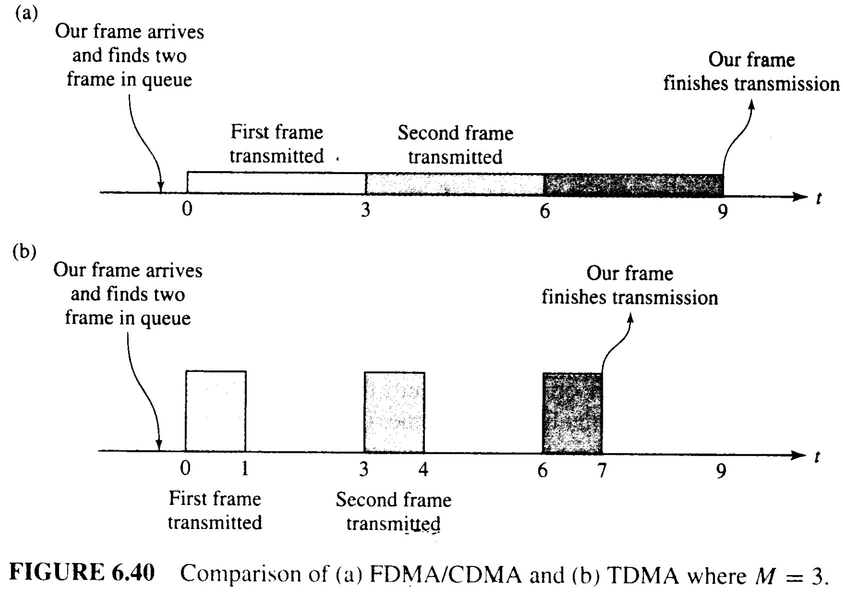

(a)
Our frame arrives and finds two frame in queue
Our frame finishes transmission
First frame transmitted
Second frame transmitted
0
3
6
9
t
(b)
Our frame arrives and finds two frame in queue
Our frame finishes transmission
0
1
3
4
6
7
9
t
First frame transmitted
Second frame transmitted

FIGURE 6.40 Comparison of (a) FDMA/CDMA and (b) TDMA where $M = 3$.

a system with $M = 3$ stations and assuming $X = 1$. In the FDMA system we assume that the transmissions in each channel are slotted. Therefore, the transmissions consist of a sequence of slots that are three time units long, as shown in the figure. When a frame arrives at a given station, the frame must wait for the transmission of all frames in the queue. If a frame arrives to an empty system, the frame must still wait until the beginning of the next slot. Thus our frame arrival in Figure 6.40a must wait till the beginning of the next time slot at $t = 0$ and then for the transmission of the two frames in the queue. Finally, at time $t = 6$ our frame begins transmission that is completed at time $t = 9$.

Figure 6.40b also shows the transmissions as viewed by a station in the TDMA case. Here the station transmits at the full rate for one slot out of the M slots in the frame. A frame that arrives at a station must wait for the transmission of all frames in the queue. Note that each such frame found in queue implies M time units of waiting time to the arriving frame. Thus a frame that arrives at the same time as in part (a) would have to wait until the beginning of the next cycle at $t = 0$ and then for the transmission of the two frames it found in queue. At time $t = 6$ our frame enters service. In TDMA the frame is transmitted at *full speed,* so the frame finishes transmission at $t = 7$. The last frame transmission time is the main difference between the TDMA and FDMA systems.

This example shows that the TDMA and FDMA systems have the same time T_{access} from when a frame arrives at a station to when the frame begins transmission. The access time T_{access} has two components in delay: (1) the time τ_0 until the beginning of the next cycle and (2) the time W waiting for the frames found in queue upon arrival. In Appendix A, Equation (A.61), we show that the average access time is given by

$$\frac{E[T_{access}]}{X} = \frac{M}{2} + \frac{\rho M}{2(1-\rho)} \tag{6.30}$$

where the first term is $E[\tau_0]$ and the next term is $E[W]$. The term ρ is called the *load* of a station and is defined by ρ = arrival rate at a station × transmission time = $(\lambda/M)(MX) = \lambda X$.

The total frame delay in each station is obtained by adding the frame transmission time to the average access time. For FDMA the frame transmission time is MX, and so the total normalized frame delay is

$$\frac{E[T_{FDMA}]}{X} = \frac{\rho M}{2(1-\rho)} + \frac{M}{2} + M \tag{6.31}$$

For TDMA the frame transmission time is X, and so the average total frame delay is

$$\frac{E[T_{TDMA}]}{X} = \frac{\rho M}{2(1-\rho)} + \frac{M}{2} + 1 \tag{6.32}$$

Thus we see that TDMA outperforms FDMA because of the faster frame transmission time. In particular, for low values of ρ the FDMA delay is larger by $M - 1$ frame times. However, both TDMA and FDMA have the undesirable feature that the average total frame delay grows with the number of stations M. As ρ approaches 1, both TDMA and FDMA have delays that are proportional to M. In this respect both TDMA and FDMA compare poorly to an ideal system that would combine all the traffic from the stations into one multiplexer and transmit always at the full rate R. Such a system corresponds to the $M = 1$ case. The total load in such a combined system would be $\rho = \lambda X$, and so the average delay in such an ideal system would be

$$\frac{E[T_{TDM}]}{X} = \frac{\rho}{2(1-\rho)} + \frac{1}{2} + 1 \tag{6.33}$$

Figure 6.26 shows the average delay for TDMA as the number of stations is varied. The $M = 1$ case gives the performance of an ideal statistically multiplexed system and the poorer delay performance with increasing M is clearly evident. Next we will consider MAC scheduling protocols and we will see how they deal with the scalability problem of increasing M.

6.5.2 Performance of Polling and Token Ring Systems

Consider the polling system in Figure 6.22. The total delay incurred by a frame from the instant when it arrives at a station to when its transmission is completed has the following components: (1) The frame must first wait for the transmission of all frames that it finds ahead of it in queue. (2) The frame must also wait for the time that must elapse from when it arrives at the station to when the station is polled. (3) The frame must be transmitted, requiring X seconds. (4) Finally, the frame must propagate from its station to the receiving station. If we assume frame arrivals have exponential interarrival times and constant length, then the total frame delay is

$$E[T] = \frac{\rho}{2(1-\rho)}X + \frac{\tau'(1-\rho/M)}{2(1-\rho)} + X + \tau_{average} \tag{6.34}$$

where each term in the equation corresponds to the delay component in the preceding list and where $\tau_{average}$ is the average time required for a frame to propagate from the source station to the destination station [Bertsekas 1992, p. 201]. The total frame delay normalized to X is then

$$\frac{E[T]}{X} = \frac{\rho}{2(1-\rho)} + \frac{a'(1-\rho/M)}{2(1-\rho)} + 1 + \frac{\tau_{average}}{X} \tag{6.35}$$

In the preceding expression, a' is the ratio of the total walk time to the service time. Neither the average waiting time (the first term) or the transmission time (the third term) is proportional to M. The only dependence on M is through $a' = Mt'/X$ and ρ/M. Figure 6.41 shows the average frame delay for a polling system with $M = 32$ stations and for $a' = 0, 0.5, 1, 5, 10$. As long as a' is less than 1, the average frame delay does not differ significantly from that of an ideal statistical multiplexer system, which in this case corresponds to the M/D/1 system described in Section 5.5.1. This is a clear improvement over channelization schemes.

Next we compare the average waiting time incurred by frames in several variations of token ring. The mean frame transfer delay is obtained by adding a frame transmission time to the average waiting time. These waiting time results are taken from [Bertsekas 1992]. The following expressions assume that the frame transmission time X includes the time required to transmit the token. The normalized mean waiting time for a token

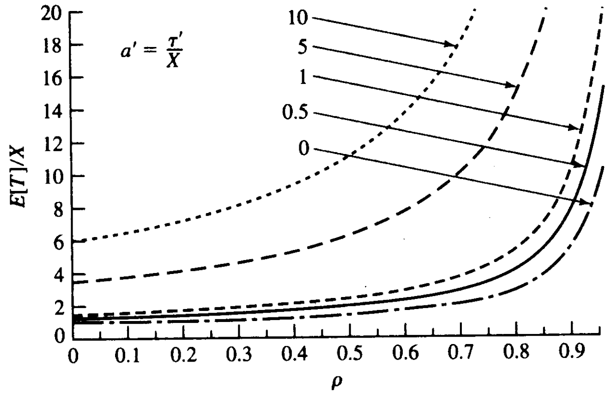


FIGURE 6.41 Average frame delay for polling, $M = 32$ stations.

ring in which there is no limit on the number of frame transmissions/token is given by

$$\frac{E[W]}{X} = \frac{\rho}{2(1-\rho)} + \frac{a'(1-\rho/M)}{2(1-\rho)} \tag{6.36}$$

Note that this corresponds to the average frame delay expression presented earlier for polling systems.

Figure 6.42 shows the mean waiting time for the system with $M = 32$ stations and unlimited service/token. It can be seen that when the normalized ring latency a' is less than 1, the system performance that does not differ significantly from that of an ideal statistical multiplexer system. As a' becomes much larger than 1, the average waiting time can be seen to increase significantly.

Now consider a token ring in which there is a limit of one frame/token and in which token reinsertion is done according to the multitoken operation. The normalized mean waiting time for the token ring is given by [Bertsekas 1992, p. 201]:

$$\frac{E[W]}{X} = \frac{\rho + a'(1+\rho/M)}{2\left(1-\left(1+\frac{a'}{M}\right)\rho\right)} \tag{6.37}$$

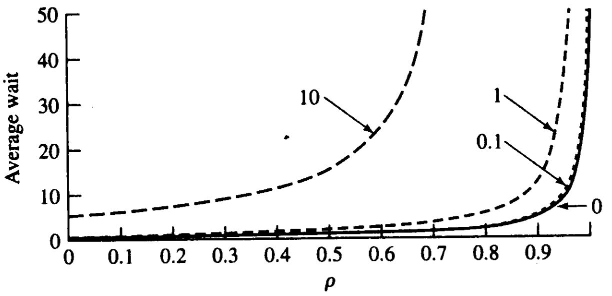


FIGURE 6.42 Mean waiting time for token ring, $M = 32$ stations, unlimited service/token.

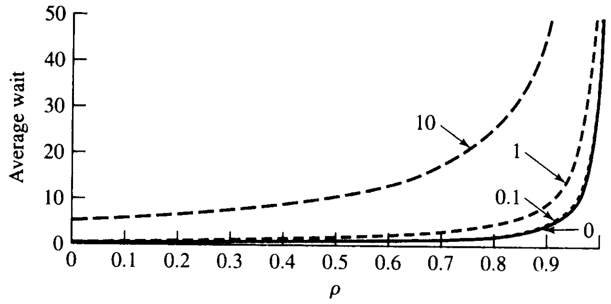


FIGURE 6.43 Mean waiting time for multitoken ring, $M = 32$, one frame/token.

Note that the mean waiting time grows without bound as ρ approaches $1/(1 + a'/M)$, which agrees with our previous result for the maximum normalized throughput ρ_{max} in Equation (6.15).

Figure 6.43 shows the mean waiting time for a system that places a limit of one frame transmission/token and uses multitoken operation. The number of stations is assumed to be $M = 32$. Recall that the multitoken operation reinserts the free token in the minimum time possible. When the normalized ring latency is less than 0.1, the mean waiting time does not differ significantly from that of an ideal statistical multiplexer system. However, as a' increases, the maximum throughput decreases. For example, when $a' = 10$, the maximum throughput is 0.76. The figure shows that this results in increased waiting times at lighter loads.

Finally, consider a token ring in which there is a limit of one frame/token and in which the token reinsertion is done according to the single-frame operation. The mean waiting time is [Bertsekas 1992, p. 202]:

$$\frac{E[W]}{X} = \frac{\rho(1 + 2a' + a'^2) + a'(1 + \frac{\rho}{M}(1 + a'))}{2(1 - (1 + a'(1 + \frac{1}{M}))\rho)} \qquad (6.38)$$

Note again that the mean waiting time grows without bound as ρ approaches $1/(1 + a'(1 + 1/M))$, in agreement with our previous result for ρ_{max} in Equation (6.17).

Figure 6.44 shows the waiting time for a token ring with a limit of one frame transmission/ token and single-frame operation. In this case the free token is not reinserted until after the entire frame is received back at a station. Again the number of stations is $M = 32$. It can be seen that the mean waiting time for this system is much more sensitive to the normalized ring latency. Performance comparable to an ideal statistical multiplexer system is possible only when $a' < 0.01$. The maximum throughput decreases rapidly with increasing a' so that by the time $a' = 1$, the maximum throughput is only 0.47. A comparison of Figure 6.43 and Figure 6.44 shows how multitoken operation is essential when the normalized ring latency becomes much larger than 1.

The preceding results for mean waiting time do not include the case of single-token operation. From Figure 6.25 we know that for $a' < 1$, the mean waiting times are the same as for a system with multitoken operation. On the other hand, as a' becomes much larger than 1, the system behaves like a system with single-frame operation.

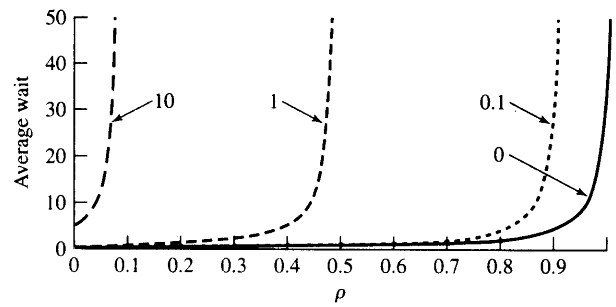


FIGURE 6.44 Mean waiting time for single-frame token ring, $M = 32$.

6.5.3 Random Access and CSMA-CD

The analysis of the delay performance of random access systems is quite involved because of the complex interactions between users through collisions in the channel. Each analysis of a random access system must first deal with modifications to stabilize the behavior of the system. We refer the reader to [Bertsekas] for detailed analyses of random access systems.

We use the following complicated expression from [Schwartz 1987] for the average delay in a CSMA-CD system for the case of constant frame lengths

$$\frac{E[T]}{X} = \rho\frac{1 + (4e+2)a + 5a^2 + 4e(2e-1)a^2}{2\{1 - \rho(1 + (2e+1)a)\}} + 1 + 2ea - \frac{(1 - e^{-2a\rho})\left(\frac{2}{\rho} + 2ae^{-1} - 6a\right)}{2(e^{-\rho}e^{-\rho a - 1} - 1 + e^{-2\rho a})} + \frac{a}{2} \tag{6.39}$$

Equation 6.39 is used to demonstrate the impact of a on the delay performance in CSMA-CD shown in Figure 6.51. As expected, the transfer delay performance of CSMA-CD is good as long as a is much less than 1.

PART II: Local Area Networks

6.6 LAN PROTOCOLS

In Chapter 1 we noted that the development of LANs was motivated by the need to share resources and information among workstations in a department or workgroup. We also noted that the requirements for LANs are different than those in a wide area or in a public network. The short distances between computers imply that low-cost, high-speed, reliable communications is possible. The emphasis on low cost implies a broadcast network approach that does not use equipment that switches information between stations. Instead, the stations cooperate by executing a MAC protocol that minimizes the incidence of collisions in a shared medium (e.g., a wire).

In this section we discuss general aspects of LAN standards. Most LAN standards have been developed by the IEEE 802 committee of the Institute of Electrical and Electronic Engineers (IEEE), which has been accredited in the area of LANs by the American National Standards Institute (ANSI). The set of standards includes the CSMA-CD (Ethernet) LAN and the token-passing ring LAN. It also includes the definition of the logical link control, which places LANs within the data link layer of the OSI reference model.

6.6.1 LAN Structure

The structure of a typical LAN is shown in Figure 6.45a. A number of computers and network devices such as printers are interconnected by a shared transmission medium, typically a cabling system, which is arranged in a bus, ring, or star topology. The cabling system may use twisted-pair cable, coaxial cable, or optical fiber transmission media. In some cases the cabling system is replaced by wireless transmission based on radio or infrared signals. The Ethernet bus topology using coaxial cable is shown in Figure 6.45a. LAN standards define physical layer protocols that specify the physical properties of the cabling or wireless system, for example, connectors and maximum cable lengths, as well as the digital transmission system, for example, modulation, line code, and transmission speed.

The computers and network devices are connected to the cabling system through a **network interface card (NIC)** or **LAN adapter card** (Figure 6.45b). For desktop computers the NIC is inserted into an expansion slot or built into the system. Laptop

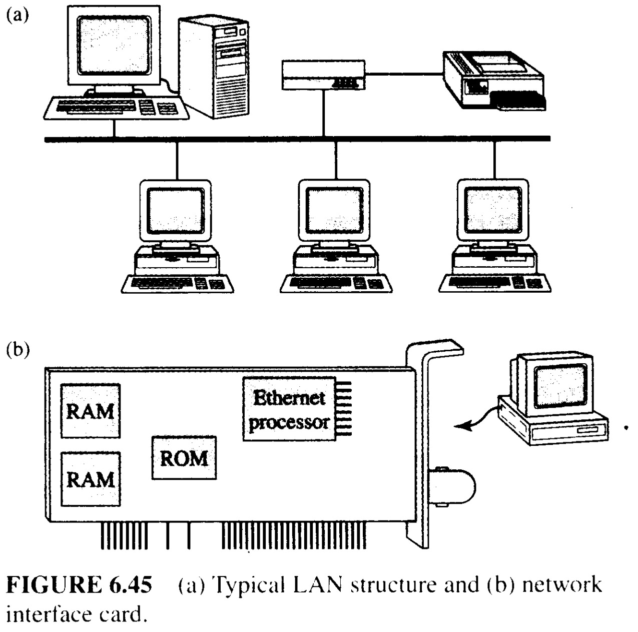


FIGURE 6.45 (a) Typical LAN structure and (b) network interface card.

computers typically use the smaller PCMCIA card, which is inserted into a slot that can also be used by a modem or other device.

The NIC card coordinates the transfer of information between the computer and the network. The NIC card transfers information in parallel format to and from main memory (RAM) in the computer. On the other hand, the NIC card transfers information in serial format to and from the network, so parallel-to-serial conversion is one of the NIC's functions. The speed of the network and the computer are not matched, so the NIC card must also buffer data.

The NIC card has a port that meets the connector and transmission specifications of physical layer standards. The NIC card includes read-only memory (ROM) containing firmware that allows the NIC to implement the MAC protocol of a LAN standard. This process involves taking network layer packets, encapsulating them inside MAC frames, and transferring the frames by using the MAC protocol, as well as receiving MAC frames and delivering the network layer packets to the computer.

Each NIC card is assigned a *unique MAC or physical address* that is burned into the ROM. Typically, the first three bytes of the address specify the NIC vendor, and the remaining bytes specify a unique number for that vendor. The NIC card contains hardware that allows it to recognize its physical address, as well as the broadcast address. The hardware can also be set to recognize multicast addresses that direct frames to groups of stations. The NIC card can also be set to run in "promiscuous" mode where it listens to all transmissions. This mode is used by system administrators to troubleshoot the network. It is also used by hackers to intercept unencrypted passwords and other information that can facilitate unauthorized access to computers in the LAN.[12]

6.6.2 The Medium Access Control Sublayer

The layered model in Figure 6.46 shows how the LAN functions are placed within the two lower layers of the OSI reference model. The data link layer is divided into two sublayers: the logical link control (LLC) sublayer and the medium access control (MAC) sublayer. The MAC sublayer deals with the problem of coordinating the access to the shared physical medium. Figure 6.46 shows that the IEEE has defined several MAC standards, including IEEE 802.3 (Ethernet) and IEEE 802.5 (token ring). Each MAC standard has an associated set of physical layers over which it can operate.

The MAC layer provides for the connectionless transfer of datagrams. Because transmissions in LANs are relatively error free, the MAC protocols usually do not include procedures for error control. The MAC entity accepts a block of data from the LLC sublayer or directly from the network layer. This entity constructs a PDU that includes source and destination MAC addresses as well as a *frame check sequence (FCS)*, which is simply a CRC checksum. The MAC addresses specify the physical connections of the workstations to the LAN. The main task of the MAC entities is to

[12] LANs were developed to operate in a private environment where an element of trust among users could be assumed. This assumption is no longer valid, so network security protocols such as those presented in Chapter 11 are now required.

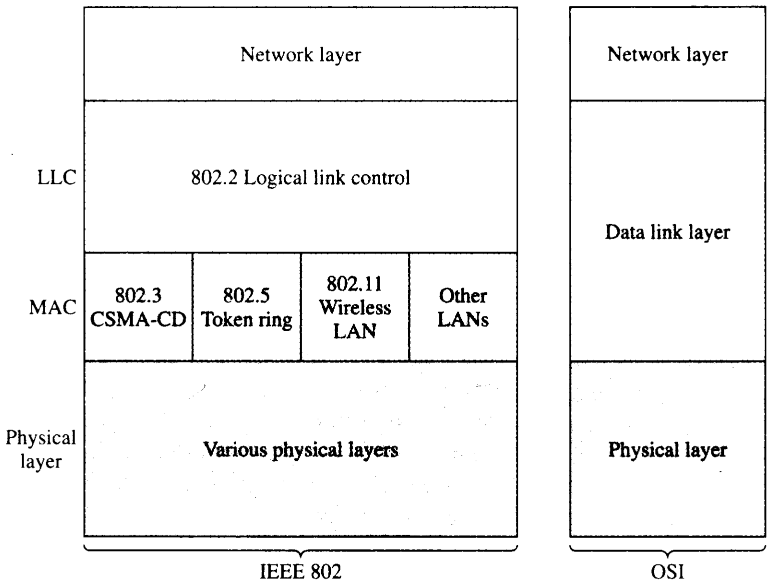


FIGURE 6.46 IEEE 802 LAN standards.

execute the MAC protocol that directs when they should transmit the frames into the shared medium.

In Figure 6.47 we show the protocol stacks of three workstations interconnected through a LAN. Note how all three MAC entities must cooperate to provide the datagram transfer service to the LLC sublayer. In other words, the interaction between MAC entities is not between pairs of peers, but rather all entities must monitor all frames that are transmitted onto the shared medium. We defer the discussion of the specific MAC protocols to the sections on individual LAN standards.

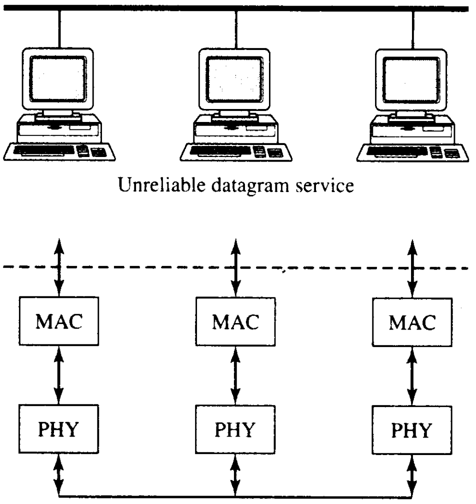


FIGURE 6.47 The MAC sublayer provides unreliable datagram service.

6.6.3 The Logical Link Control Sublayer

The IEEE 802 committee has also defined a **logical link control (LLC)** sublayer that operates over all MAC standards. The LLC can enhance the datagram service offered by the MAC layer to provide some of the services of HDLC at the data link layer. This approach makes it possible to offer the network layer a standard set of services while hiding the details of the underlying MAC protocols. The LLC also provides a means for exchanging frames between LANs that use different MAC protocols.

The LLC builds on the MAC datagram service to provide three HDLC services. Type 1 LLC service is *unacknowledged connectionless service* that uses unnumbered frames to transfer unsequenced information. Recall from Chapter 5 that the HDLC protocols use unnumbered frames in some of their message exchanges. Type 1 LLC service is by far the most common in LANs. Type 2 LLC service uses information frames and provides *reliable connection-oriented service* in the form of the asynchronous balanced mode of HDLC. A connection setup and release is required, and the connection provides for error control, sequencing, and flow control. Figure 6.48 shows two type 2 LLC entities at stations A and C providing reliable frame transfer service. Type 2 operation is useful when the endsystems do not use a transport layer protocol to provide reliable service. For example, type 2 is used in several proprietary PC LAN software products. Type 3 LLC service provides *acknowledged connectionless service*, that is, connectionless transfer of individual frames with acknowledgments. To provide type 3 LLC service, two additional unnumbered frames were added to the set defined by HDLC.

Additional addressing is provided by the LLC to supplement the addressing provided by the medium access control. A workstation in the LAN has a single MAC

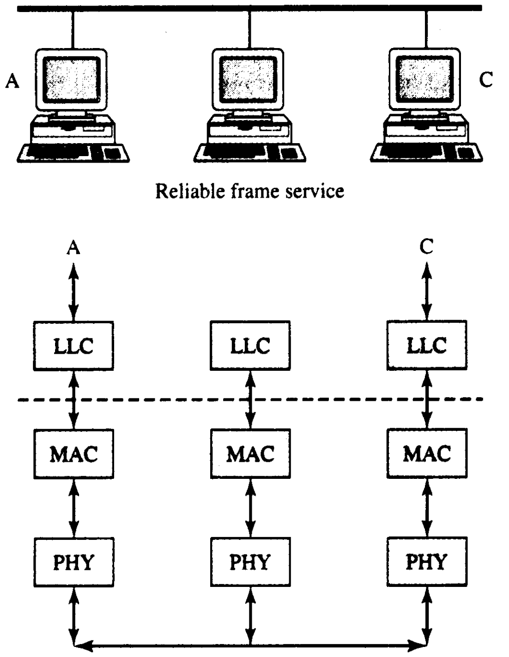


FIGURE 6.48 The LLC can provide reliable frame transfer service.

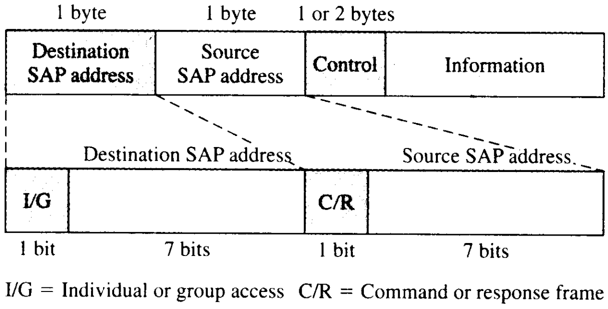


FIGURE 6.49 LLC PDU structure.

(physical) address. However, at any given time such a workstation might simultaneously handle several data exchanges originating from different upper-layer protocols but operating over this same physical connection. These logical connections are distinguished by their *service access point (SAP)* in the LLC, as shown in Figure 6.49. For example, frames that contain IP packets are identified by hexadecimal 06 in the SAP, frames that contain Novell IPX are identified by E0, frames with OSI packets by FE, and frames with SNAP PDUs (discussed below) by AA. In practice, the LLC SAP specifies in which memory buffer the NIC places the frame contents, thus allowing the appropriate higher-layer protocol to retrieve the data.

The top part of Figure 6.49 shows the LLC PDU structure that consists of one byte each for source and destination SAP addresses, one or two bytes for control information, and the information itself, that is, the network layer packet. The bottom part shows the details of the address bytes. In general a seven-bit address is used. The first bit of the destination SAP address byte indicates whether it is an individual or group address. The first bit of the source SAP address byte is not used for addressing and instead is used to indicate whether a frame is a command or response frame. The control field is one byte long if three-bit sequence numbering is used. The control field is two bytes long when extended sequence numbering is used.

The LLC PDU is encapsulated in IEEE MAC frames as shown in Figure 6.50. The MAC adds both a header and a trailer. Note the accumulation of header overhead:

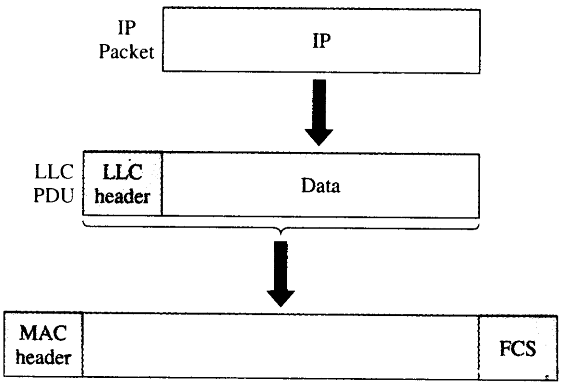


FIGURE 6.50 Packet encapsulation into a MAC frame.

After TCP and IP have added their minimum of 20 bytes of headers, the LLC adds 3 or 4 bytes, and then the MAC adds its header and trailer. In the next several sections we consider MAC protocols. We then consider specific frame formats as we introduce several of the IEEE 802 LAN standards.

BUILDING LARGE LAN NETWORKS

LANs were initially based on sharing the bandwidth of some medium. The physical layer of any medium has a given fixed amount of bandwidth and this fact ultimately limits the number of stations that can be attached to the medium. To increase the number of stations in a LAN one can break the stations into different segments and interconnect these segments using bridges. The function of the bridges is to allow frames to be broadcast only over those segments where it is necessary. Another approach to building larger LANs is to do away with sharing of the medium. Both of these techniques are used in the current generation of LAN switches.

6.7 ETHERNET AND IEEE 802.3 LAN STANDARD

The **Ethernet** LAN protocol was developed in the early 1970s by Robert Metcalfe and his colleagues working at Xerox as a means of connecting workstations. In the early 1980s DEC, Intel, and Xerox completed the "DIX" Ethernet standard for a 10 Mbps LAN based on coaxial cable transmission. This standard formed the basis for the IEEE 802.3 LAN standard that was first issued in 1985 for "thick" coaxial cable. The Ethernet and IEEE 802.3 standards differ primarily in the definition of one header field, which we discuss below. The IEEE 802.3 standard has been revised and expanded every few years.[13] Specifications have been issued for operation using "thin" coaxial cable, twisted-pair wires, and single-mode and multimode optical fiber. Higher-speed versions were approved in 1995 (100 Mbps Fast Ethernet) and in 1998 (1000 Mbps Gigabit Ethernet), and in 2002 (10 Gbps Ethernet).

6.7.1 Ethernet Protocol

The original 802.3 standard was defined for a bus-based coaxial cable LAN in which terminal transmissions are broadcast over the bus medium using Carrier Sensing Multiple Access with Collision Detection (CSMA-CD) for the MAC protocol. A station with a frame to transmit waits until the channel is silent. Ethernet adopts the 1-persistent mode so that when the channel goes silent, the station transmits immediately. During transmission, the station continues to listen for collisions that can occur if other stations also begin to transmit. If a collision occurs, the station aborts the transmission and schedules a later random time when it will reattempt to transmit its frame. If a collision

[13] The 2002 edition of IEEE Standard 802.3 is over 1500 pages with more than 40 annexes!

does not occur within two propagation delay times, then the station knows that it has captured the channel, as the station's transmission will have reached all stations and so they will refrain from transmitting until the first station is done. A **minislot time** defines a time duration that is at least as big as two propagation delays.

The critical parameter in the CSMA-CD system is the minislot time that forms the basis for the contention resolution that is required for a station to seize control of the channel. The original 802.3 was designed to operate at 10 Mbps over a maximum distance of 2500 meters. When allowances are made for four repeaters, the delay translates into a maximum end-to-end propagation delay of 51.2 microseconds. At 10 Mbps this propagation delay equals 512 bits, or 64 bytes, which was selected as the minimum frame length or minislot.

The IEEE 802.3 standard specifies that the rescheduling of retransmission attempts after a collision uses a *truncated binary exponential backoff algorithm.* If a frame is about to undergo its nth retransmission attempt, then its retransmission time is determined by selecting an integer equally likely in the range between 0 and $2^k - 1$, where $k = \min(n, 10)$. That is, the first retransmission time involves zero or one minislot times; the second retransmission time involves 0, 1, 2, or 3 minislot times; and each additional slot retransmission extends the range by a power of 2 until the maximum range of $2^{10} - 1$. The increased retransmission range after each collision is intended to increase the likelihood that retransmissions will succeed. Up to 16 retransmissions will be attempted, after which the system gives up.

The typical activity in the Ethernet channel consists of idle periods, contention periods during which stations attempt to capture the channel, and successful frame transmission times. When the channel approaches saturation, there are few idle periods and mostly frame transmissions alternate with contention periods. Therefore, at or near saturation the time axis consists of the following three subintervals: a period L/R seconds long during which frames are transmitted, a period t_{prop} seconds long during which all the other stations find out about the end of the transmission, and a contention period consisting of an integer number of minislot times, each of duration $2t_{prop}$ seconds. In Section 6.2.4 we showed that the average number of minislots in a contention period is approximately $e = 2.71$. Therefore, the fraction of time that the channel is busy transmitting frames is

$$\frac{L/R}{L/R + t_{prop} + 2et_{prop}} = \frac{1}{1 + (1 + 2e)t_{prop}R/L} = \frac{1}{1 + (1 + 2e)a} = \frac{1}{1 + 6.44a} \tag{6.40}$$

where $a = t_{prop}R/L$.

EXAMPLE Effect of a on Ethernet Performance

Let us assess the impact of the parameter a on the performance of an Ethernet LAN. Suppose that $a = 0.01, 0.1$, and 0.2. The corresponding maximum possible normalized throughputs are then 0.94, 0.61, and 0.44. Thus we see that a has a dramatic impact on the throughput that can be achieved.

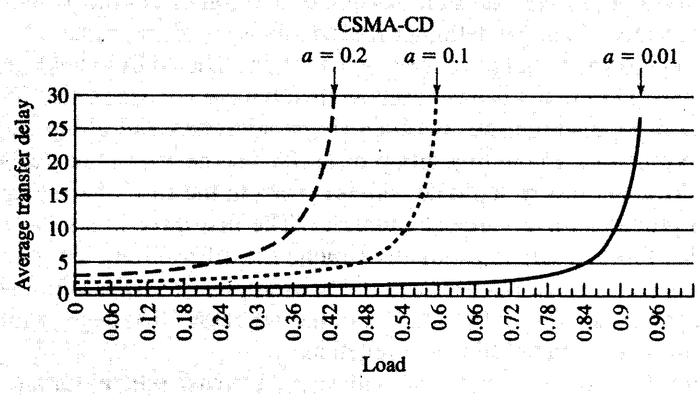
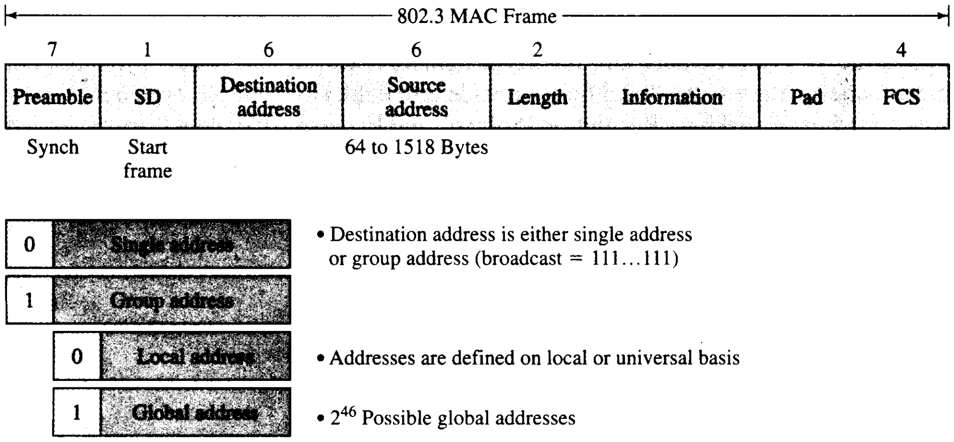


FIGURE 6.51 Frame transfer delay for Ethernet example: as the load reaches its maximum, the transfer delay grows very large.

In Section 6.5 we presented an expression for the average frame transfer delay in Ethernet under the following assumptions: Frame arrival times are independent of each other; frame interarrival times have an exponential distribution; all frames are of the same length. Figure 6.51 shows the average transfer delays for $a = 0.01, 0.1$, and 0.2. It can be seen that the transfer delays grow very large as the load approaches the maximum possible value for the given value of a.

6.7.2 Frame Structure

Figure 6.52 shows the MAC frame structure for the IEEE 802.3. The frame begins with a seven-octet preamble that repeats the octet 10101010. This pattern produces a square wave that allows the receivers to synchronize to the transmitter's bit stream. The



- Destination address is either single address or group address (broadcast = 111...111)
- Addresses are defined on local or universal basis
- 2^{46} Possible global addresses

FIGURE 6.52 IEEE 802.3 MAC frame.

preamble is followed by the start frame delimiter that consists of the pattern 10101011. The two consecutive 1s in the delimiter indicate the start of the frame.

The destination and source address fields follow. The address fields are six bytes long. (Two-byte address fields have been defined but are not used). The first bit of the destination address distinguishes between single addresses and group addresses that are used to multicast a frame to a group of users. The next bit indicates whether the address is a local address or a global address. Thus in the case of six-byte addresses, the standard provides for 2^{46} global addresses. The first three bytes specify the NIC vendor and is called the Organizationally Unique Identifier (OUI). The OUI allows up to $2^{24} = 16{,}777{,}215$ addresses per vendor. For example, Cisco has addresses in which the first three bytes are 00-00-0C and 3Com has addresses that begin with 02-60-8C, where the numbers are in hexadecimal notation.

There are three types of physical addresses. **Unicast addresses** are the unique address permanently assigned to a NIC card. The card normally matches transmissions against this address to identify frames destined to it. **Multicast addresses** identify a group of stations that are to receive a given frame. NIC cards are set by their host computer to accept specific multicast addresses. Multicasting is an efficient way of distributing information in situations where multiple entities or processes require a piece of information as, for example, in the spanning tree algorithm (discussed in Section 6.11.1 on bridges). The **broadcast address**, indicated by the all 1s physical address, indicates that all stations are to receive a given packet.

The length field indicates the number of bytes in the information field. The longest allowable 802.3 frame is 1518 bytes, including the 18-byte overhead but excluding the preamble and SD. The pad field ensures that the frame size is always at least 64 bytes long. The maximum information field size of 1500 bytes translates into the hexadecimal code 05DC.

The FCS field is the CCITT 32-bit CRC check discussed in Chapter 3. The CRC field covers the address, length information, and pad fields. Upon receiving a frame, the NIC card checks to see that the frame is of an acceptable length and then checks the received CRC for errors. If errors are detected, the frame is discarded and not passed to the network layer.

Figure 6.53 shows the frame structure for the Ethernet (DIX) standard, also known as Ethernet II. The Ethernet frame has a type field that identifies the upper-layer protocol in the same location as the 802.3 field has its length field. For example, type field values are defined for IP, Address Resolution Protocol, and Reverse ARP (which are discussed in Chapter 8). The Ethernet standard assigns type field values starting at 0600. Recall that the length field in IEEE 802.3 never takes on values larger than 05DC. Thus the

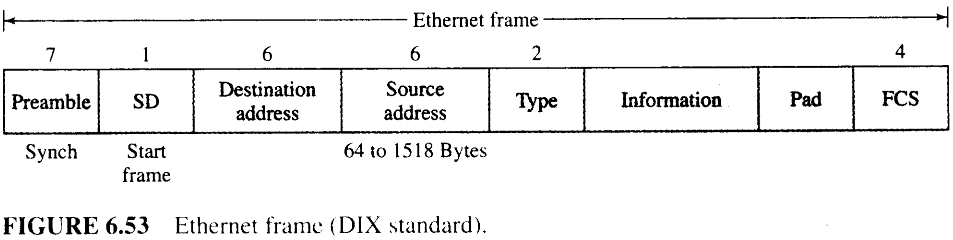


FIGURE 6.53 Ethernet frame (DIX standard).

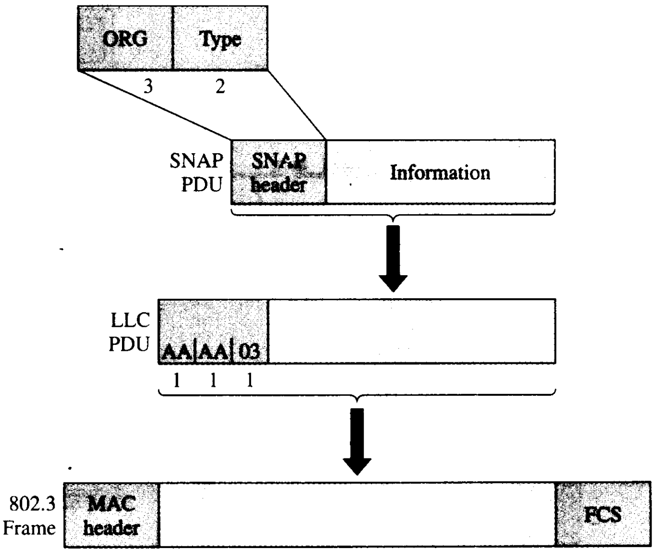


FIGURE 6.54 LLC-SNAP header for encapsulating DIX Ethernet frames in 802.3 frames.

value of this field can tell an Ethernet controller whether it is handling an Ethernet frame or an IEEE 802.3 frame.

Nevertheless, the IEEE standard assumes that the LLC is always used, which provides the upper-layer protocol indication through the SAP field in the LLC header as shown in Figure 6.54. Upper-layer software programs developed to work with DIX Ethernet expect a "Type" field. To allow Ethernet-standard software to work with IEEE 802.3 frames, the **Subnetwork Access Protocol (SNAP)** provides a way of encapsulating Ethernet-standard frames inside a Type 1 LLC PDU. The DSAP and SSAP fields in the LLC header (see Figure 6.49) are set to AA to notify the LLC layer that an Ethernet frame is enclosed and should be processed accordingly. The value 03 in the control field indicates Type 1 service. The SNAP header consists of a three-byte vendor code (usually set to 0) and the two-byte type field required for compatibility.

6.7.3 Physical Layers

Table 6.2 shows the various physical layers that have been defined for use with IEEE 802.3. Each of these medium alternatives are designated by three parameters

TABLE 6.2 IEEE 802 3 10 Mbps medium alternatives.

	10Base5	10Base2	10BaseT	10BaseF
Medium	Thick coax	Thin coax	Twisted pair	Optical fiber
Maximum segment length	500 m	185 m	100 m	2 km
Topology	Bus	Bus	Star	Point-to-point link

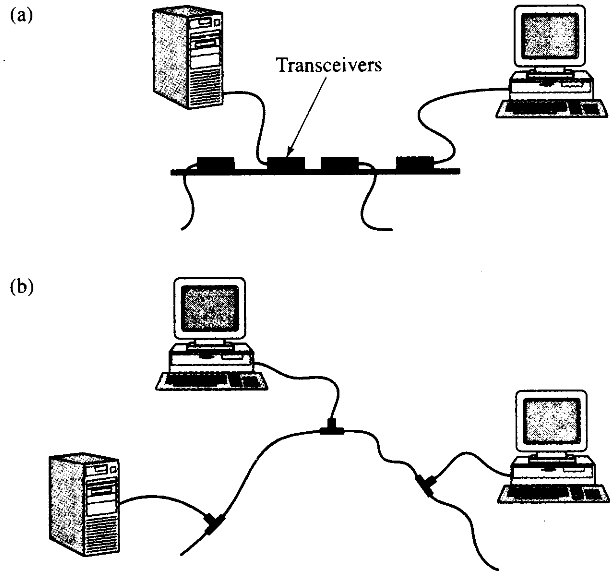


FIGURE 6.55 Ethernet cabling using (a) thick and (b) thin coaxial cable. (Note: The T junction typically attaches to the NIC.)

that specify the bit rate, the transmission technique, and the maximum segment length. For example, the original standard specified 10Base5, which made use of thick (10 mm) coaxial cable operating at a data rate of *10* Mbps, using *base*band transmission and with a maximum segment length of *5*00 meters. The transmission uses Manchester coding, which is discussed in Chapter 3. This cabling system required the use of a *transceiver* to attach the NIC card to the coaxial cable. The thick coaxial cable Ethernet was typically deployed along the ceilings in building hallways, and a connection from a workstation in an office would tap onto the cable as shown in Figure 6.55a. Thick coaxial cable is awkward to handle and install. The 10Base2 standard uses thin (5 mm) coaxial cable operating at 10 Mbps and with a maximum segment of 185 meters. The cheaper and easier-to-handle thin coaxial cable makes use of T-shaped BNC junctions as shown in Figure 6.55b. 10Base5 and 10Base2 segments can be combined through the use of a *repeater* that forwards the signals from one segment to the other.

The 10BaseT standard involves the use of two unshielded twisted pairs (UTPs) of copper wires (diameter of 0.4 mm to 0.6 mm) operating at 10 Mbps and connected to a **hub** as shown in Figure 6.56a. The T designates the use of twisted pair. The advantage of twisted pair is low cost and its prevalence in existing office wiring where it is used for telephones. Existing wiring arrangements allow the hubs to be placed in telephone wiring closets. The use of the 10BaseT standard also involves a move toward a star topology in which the stations connect the twisted pair to a hub where the collisions take place.

The star topology of 10BaseT provides three approaches to operating the LAN. In all three approaches the stations implement the CDMA-CD protocol. The difference

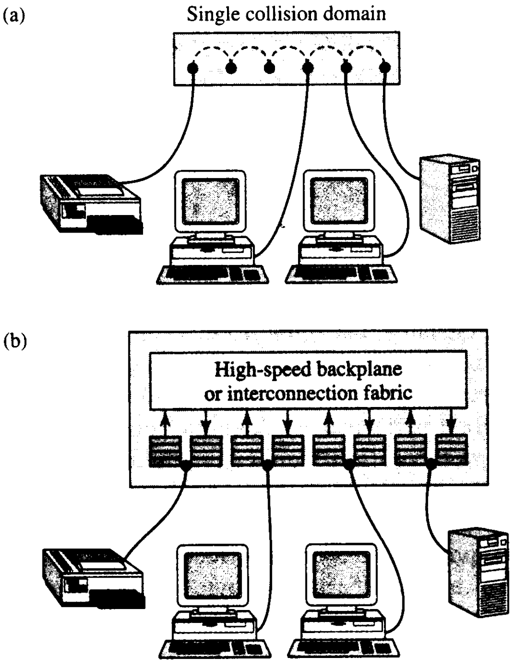


FIGURE 6.56 Ethernet (a) hub and (b) switch topologies using twisted-pair cabling.

is in the operation of the hub at the center of the star. In the first approach, the hub monitors all transmissions from the stations. When there is only one transmission, the hub repeats the transmission on the other lines. If there is a collision, that is, more than one transmission, then the hub sends a jamming signal to all the stations. This action causes the stations to implement the backoff algorithm. In this approach the stations are said to be in the same **collision domain**, that is, a domain where a collision will occur if two or more stations transmit simultaneously.

A second approach involves operating the hub with an **Ethernet switch**, as shown in Figure 6.56b. Ethernet switch is a name commonly used for a product that implements transparent bridging, described in Section 6.11.1. In a switch, each input port buffers incoming transmissions. The incoming frames are examined and transferred to the appropriate outgoing ports. Each output port implements the Ethernet MAC protocol to transmit frames. Collisions will not occur if only a single station is attached to the line. It is possible, however, to have several stations share an input line using another hub, for example. In this case the group of stations attached to the same line will constitute a collision domain.

The number of stations in a LAN cannot be increased indefinitely. Eventually the traffic generated by stations will approach the limit of the shared transmission medium. The introduction of switching LANs provides a means of interconnecting larger numbers of stations without reaching this limit.

A third approach involves having stations transmit in **full-duplex** mode. Normal Ethernet transmission is **half duplex**, because stations transmit data in one direction at a time. In full-duplex mode a station can transmit in both directions simultaneously. Consider the case where each port in the switch has only a single station attached to it.

Introducing a dedicated transmission line for each direction enables transmissions to take place in both directions simultaneously without collisions. Note that the stations can continue to operate the CSMA-CD protocol, but they will never encounter collisions. All three of these approaches have been implemented in LAN products.

6.7.4 Fast Ethernet

The IEEE 802.3u standard was approved in 1995 to provide Ethernet LANs operating at 100 Mbps. We refer to systems that operate under this standard as Fast Ethernet. To maintain compatibility with existing standards, the frame format, interfaces, and procedures have been kept the same. Recall that the performance of the CSMA-CD medium access control is sensitive to the ratio of the round-trip propagation delay and the frame transmission time. To obtain good performance, this ratio must be small. In addition, the correct operation of the protocol itself requires the minimum frame size transmission time to be larger than the round-trip propagation delay. When the transmission speed is increased from 10 Mbps to 100 Mbps, the frame transmission time is reduced by a factor of 10. For the MAC protocol to operate correctly, either the size of the minimum frame must be increased by a factor of 10 to 640 bytes or the maximum length between stations is reduced by a factor of 10 to, say, 250 meters.

The decision in developing the 100 Mbps IEEE 802.3 standard was to keep frame sizes and procedures unchanged and to define a set of physical layers that were entirely based on a hub topology involving twisted pair and optical fiber, as shown in Table 6.3. Coaxial cable was not included in the standard. (Note that the 100BaseF option can extend up to 2000 m because it operates in full-duplex mode and operates with buffered switches only.)

The standard involves stations that use UTP wiring to connect to hubs in a star topology. To obtain a bit rate of 100 Mbps, the 100BaseT4 standard uses four UTP 3 wires (UTP, category 3, that is, ordinary telephone-grade twisted pair). The 100 Mbps transmission is divided among three of the twisted pairs and flows in one direction at a time.

The 100BaseTX uses two UTP 5 wires. The category 5 twisted pair involves more twists per meter than UTP 3, which provides greater robustness with respect to interference thus enabling higher bit rates. One pair of wires is for transmission and one for reception, so 100BaseTX can operate in full-duplex mode.

A 100BaseFX standard has also been provided that uses two strands of multimode optical to provide full-duplex transmission at 100 Mbps in each direction. The 100BaseFX system can reach over longer distances than the twisted pair options, and so it is used in interconnecting wiring closets and buildings in a campus network.

TABLE 6.3 IEEE 802.3 Fast Ethernet medium alternatives.

	100BaseT4	100BaseT	100BaseFX
Medium	Twisted pair category 3 UTP four pairs	Twisted pair category 5 UTP two pairs	Optical fiber multimode two strands
Maximum segment length	100 m	100 m	2 km
Topology	Star	Star	Star

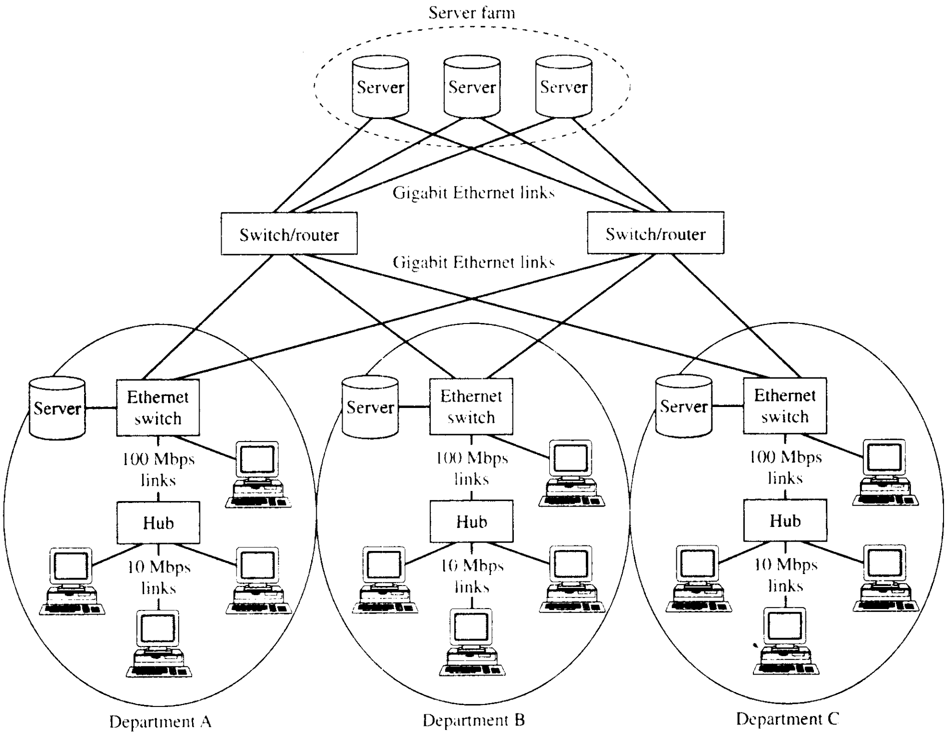


FIGURE 6.57 Deployment of Ethernet in a campus network.

The 100 Mbps IEEE 802.3 standards provide for two modes of operations at the hubs. In the first mode all incoming lines are logically connected into a single collision domain, and the CSMA-CD MAC procedure is applied. In the second mode the incoming frames are buffered and then switched internally within the hub. In the latter approach the CSMA-CD procedure is not used, and instead the IEEE 802.3 standard simply provides a means of accessing the first stage in a LAN that is based on multiplexing and switching.

Fast Ethernet is deployed in departmental networks, as shown in Figure 6.57 where Fast Ethernet switches are used to (1) aggregate traffic from shared 10 Mbps LANs, (2) provide greater bandwidth to a server, and (3) provide greater bandwidth to certain users.

6.7.5 Gigabit Ethernet

The IEEE 802.3z **Gigabit Ethernet** standard was completed in 1998 and established an Ethernet LAN that increased the transmission speed over that of Fast Ethernet by a factor of 10. The goal was to define new physical layers but to again retain the frame structure and procedures of the 10 Mbps IEEE 802.3 standard.

The increase in speed by another factor of 10 put a focus on the limitations of the CSMA-CD MAC protocol. For example, at a 1 Gbps speed, the transmission of a

TABLE 6.4 IEEE 802.3 Gigabit Ethernet medium alternatives.

	1000BaseSX	**1000BaseLX**	**1000BaseCX**	**1000BaseT**
Medium	Optical fiber multimode two strands	Optical fiber single mode two strands	Shielded copper cable	Twisted pair category 5 UTP
Maximum segment length	550 m	5 km	25 m	100 m
Topology	Star	Star	Star	Star

minimum size frame of 64 bytes can result in the transmission being completed before the sending station senses a collision. For this reason, the slot time was extended to 512 bytes. Frames smaller than 512 bytes must be extended with an additional carrier signal, in effect resulting in the same overhead as in padding the frame. In addition, an approach called *frame bursting* was introduced to address this scaling problem. Stations are allowed to transmit a burst of small frames, in effect to improve the key ratio a. Nevertheless, it is clear that with Gigabit Ethernet the CSMA-CD access control reached the limits of efficient operation. In fact, the standard preserves the Ethernet frame structure but operates primarily in a switched mode.

Gigabit Ethernet physical layer standards have been defined for multimode fiber with maximum length of 550 m, single-mode fiber with maximum length of 5 km, and four-pair category 5 UTP at a maximum length of up to 100 m. Table 6.4 lists the different medium alternatives.

Gigabit Ethernet is deployed in campus networks as shown in Figure 6.57. Gigabit Ethernet provides the high bandwidth to connect departmental switches and server farms to campus backbone switches.

6.7.6 10 Gigabit Ethernet

The IEEE 802.3ae 10 Gigabit Ethernet draft supplement to the IEEE 802.3 standard was ratified in 2002 to extend the transmission speed to 10 Gbps. Because the ratio of the round-trip propagation delay and the frame transmission time becomes very small, 10 Gigabit Ethernet is defined only for full-duplex mode providing a point-to-point Ethernet connectivity service with the CSMA-CD algorithm disabled. The standard defines two types of physical layer: the LAN PHY and the WAN PHY. Both types differ in framing but support the same capability in terms of distance. The LAN PHY is primarily intended to support existing Ethernet LAN applications while the WAN PHY allows 10 Gigabit Ethernet terminals to be connected through SONET OC-192c equipment.

The 10 Gigabit specification uses multiple suffixes to indicate the medium as well as the line coding type. The first set of suffixes indicates the line coding type. The suffix X denotes the use of 8B10B code (mBnB code is explained in Chapter 3). The suffix R denotes the use of 64B66B code, resulting in a more efficient code than the 8B10B code but incurring a higher implementation complexity. The suffix W indicates the

TABLE 6.5 IEEE 802.3 10 Gigabit Ethernet medium alternatives.

	10GBaseSR	10GBaseLR	10GBaseEW	10GBaseLX4
Medium	Two optical fibers Multimode at 850 nm	Two optical fibers Single-mode at 1310 nm	Two optical fibers Single-mode at 1550 nm	Two optical fibers Multimode/single-mode with four wavelengths at 1310 nm band
	64B66B code	64B66B	SONET compatibility	8B10B code
Maximum distance	300 m	10 km	40 km	300 m–10 km

encapsulation of 64B66B encoded data into SONET STS-192c payload, thus implementing the WAN PHY.

The second set of suffixes indicates the medium type. The suffix S indicates two multimode optical fibers operating at 850 nm wavelength. The suffixes L and E indicate two single-mode optical fibers operating at 1310 nm and 1550 nm wavelengths, respectively. Finally, the suffix L4 indicates two multimode or single-mode optical fibers operating at 1310 nm WDM band. Table 6.5 lists the different medium alternatives.

ETHERNET EVERYWHERE!

Ethernet is well entrenched as the technology of choice in the LAN. Each new generation of Ethernet has provided the scale to keep up with the demands for increased bandwidth within buildings and in campuses by new generations of workstations and personal computers. Demand for 1 Gbps and 10 Gbps Ethernet LANs has been stimulated by the construction of centralized application hosting and data centers where huge volumes of traffic converge and where very high bandwidths are required to minimize congestion and delay. The demand to connect users to data centers and the requirement to interconnect corporate LANs at different sites has created a need to provide Ethernet connectivity *beyond the LAN* and across metropolitan networks and even wide area networks. A number of standards have been developed to meet these needs.

Until recently, SONET and other TDM networks have provided the connectivity across MANs and WANs. Unfortunately the bit rates of the Ethernet standards and the SONET hierarchy are not well matched. For example, until recently a 1 Gbps Ethernet stream would need to be carried in a 2.5 Gbps OC-48 SONET signal. The GFP framing standard discussed in Chapter 3 combined with new virtual concatenation standards allow arbitrary numbers of STS-1 payloads to be combined to carry a particular stream. For example, these new standards allow 21 of the STS-1s in an OC-48 to carry a 1 Gbps Ethernet stream; the remaining 27 STS-1s are used for other traffic. The ability to carry Ethernet streams efficiently across longer distances makes it feasible to deploy Ethernet switches to direct streams of Ethernet frames in metropolitan networks. The new 10Gbps Ethernet standard can also be used to carry Ethernet streams directly over SONET networks across MANs and WANs. In

many metropolitan areas, "unlit" dark fiber can be leased relatively inexpensively from utility companies to provide connectivity between sites. The new 10 Gbps Ethernet standard can provide low-cost high-speed connectivity over these dark fiber connections.

6.8 TOKEN-RING AND IEEE 802.5 LAN STANDARD

Several versions of token-ring networks were developed primarily by IBM in the 1970s and 1980s. Information flows in one direction along the ring from the source to the destination and back to the source. The key notion is that medium access control is provided via a small frame called a **token** that circulates around a ring-topology network. Only the station that has possession of the token is allowed to transmit at any given time.

The ring topology brings certain advantages to medium access control. The flow of the token along the ring automatically provides each station with a turn to transmit. Thus the ring topology provides for fairness in access and for a fully distributed implementation. The token mechanism also allows for the introduction of access priorities as well as the control of the token circulation time.

The ring topology, however, is seriously flawed when it comes to faults. The entire network will fail if there is a break in any transmission link or a failure in the mechanism that relays a signal from one point-to-point link to the next. This problem is overcome by using a star topology to connect stations to a wiring closet where the wires from the stations can be connected to form a ring as shown in Figure 6.58. Reliability is provided by relays that can bypass the wires of stations that are deemed to have failed. Thus, for example, a failed station E in Figure 6.58 has been bypassed by its own relay circuit, since the power fed by station E is not available. The star topology also has

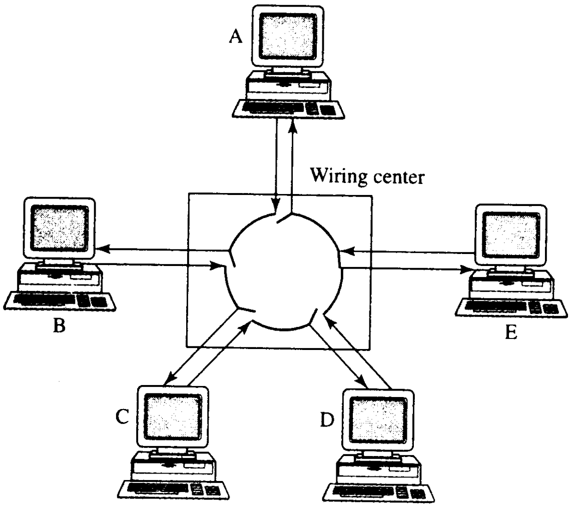


FIGURE 6.58 Token ring with improved reliability through the use of relays in a star topology.

the advantage that it can use existing telephone wiring arrangements that are found in office buildings.

The IEEE 802.5 LAN standard defines token-ring networks operating at 4 Mbps and 16 Mbps transmission. The rings are formed by twisted-pair cables using differential Manchester line coding. The maximum number of stations is set to 250.

6.8.1 Token-Ring Protocol

To transmit a frame, a station must wait for a "free" token to arrive at the interface card. When such a token arrives, the station claims the token by removing it from the ring.[14] The station then proceeds to transmit its frame into its outgoing line. The frame travels along the ring over every point-to-point link and across every interface card. Each station examines the destination address in each passing frame to see whether it matches the station's own address. If not, the frame is forwarded to the next link after a few bits delay. If the frame is intended for the station, the frame is copied to a local buffer, several status bits in the frame are set, and the frame is forwarded along the ring. The sending station has the responsibility of removing the frame from the ring and of reinserting a free token into the ring.

When the traffic on the ring is light, the token spends most of the time circulating around the ring until a station has a frame to transmit. As the traffic becomes heavy, many stations have frames to transmit, and the token mechanism provides stations with a fair round-robin access to the ring.

The approach that is used to reinsert the free token into the ring can have a dramatic effect on the performance when the delay-bandwidth product of the ring is large. To show why this happens, we first have to examine how a frame propagates around the ring. Suppose that the ring has M stations. Each station interface introduces b bits of delay between when the interface receives a frame and forwards it along the outgoing line, so the interfaces introduce Mb bits of delay. A typical value of b is 2.5.[15] If the total length of the links around the ring is d meters, then an additional delay of d/v seconds or dR/v bits is incurred because of propagation delay, where v is the propagation speed in the medium. For example, $v = 2 \times 10^8$ meters/second in twisted-pair wires, or equivalently it takes 5 microseconds to travel 1 kilometer. The *ring latency* defined in Section 6.3.3 can be expressed by

$$\tau' = d/v + Mb/R \text{ seconds} \qquad \text{and} \qquad \tau'R = dR/v + Mb \text{ bits} \qquad (6.41)$$

EXAMPLE Ring Latency and Token Reinsertion

Let us investigate the interplay between ring latency and the token reinsertion method. First suppose that we have a ring that operates at a speed of $R = 4$ Mbps with $M = 20$ stations separated by 100 meters and $b = 2.5$ bits. The ring latency (in bits) is then

[14] In fact, the station "claims" the token by flipping a specific bit from 0 to 1; this process converts the token frame into a data frame.

[15] The half-bit delay is possible because token ring uses Manchester line coding.

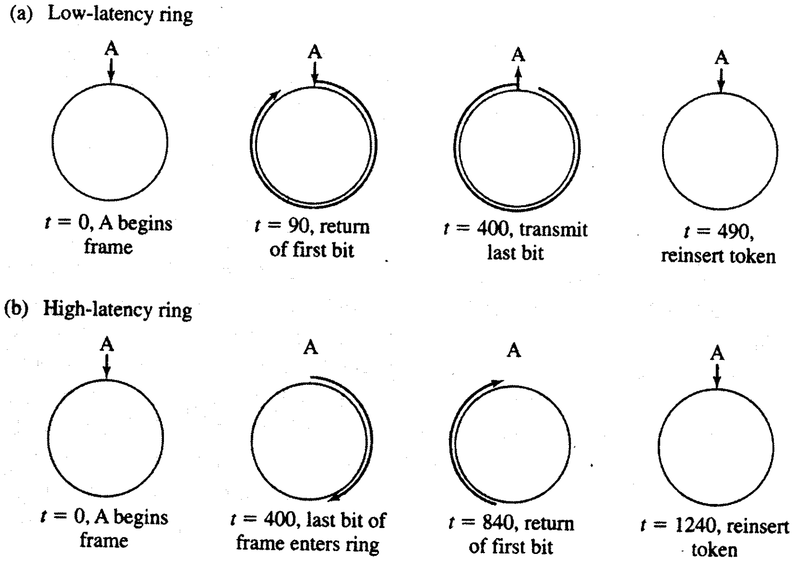


FIGURE 6.59 Ring latency and token reinsertion strategies.

$20 \times 100 \times 4 \times 10^6/(2 \times 10^8) + 20(2.5) = 90$ bits. Thus the first bit in a frame returns to the sending station 90 bit times after being inserted. On the other hand, if the speed of the ring is 16-Mbps and the number of stations is 80, then the ring latency is $80 \times 100 \times 16 \times 10^6/(2 \times 10^8) + 80(2.5) = 840$ bits.

Now suppose that we are transmitting a frame that is $L = 400$ bits long. Suppose that the token reinsertion strategy is to reinsert the token after the frame transmission is completed but not until after the last bit of the frame returns to the sending station (that is, single-frame operation). Figure 6.59a shows that the last bit in the frame returns after 490 bits in the first ring. Thus the sending station must insert an "idle" signal for 90 additional bit times before that station can reinsert the token into the ring. In the second ring the token returns after 1240 bit times, as shown in Figure 6.59b. In this case the sending station has to insert an idle signal for 840 bits times before reinserting the token. Thus we see that this token reinsertion method extends the effective length of each frame by the ring latency. For the first ring the efficiency is $400/490 =$ 82 percent; for the second ring the efficiency drops to $400/1240 = 32$ percent.

Now suppose that the token reinsertion strategy is to reinsert the token after the frame transmission is completed but not until after the header of the frame returns to the sending station (again, single-token operation). Suppose that the header is 15 bytes = 120 bits long. The header returns after $90 + 120 = 210$ bits in the first ring, as shown in Figure 6.60a. The sending station can therefore reinsert the token immediately after transmitting bit 400 of the frame. Figure 6.60b shows that in the second ring the header returns after $840 + 120 = 960$ bits. Consequently, the sending station must send an idle signal for 560 bit times before that station can reinsert the token into the ring. The first ring now operates efficiently, but the second ring has an efficiency of $400/960 =$ 42 percent.

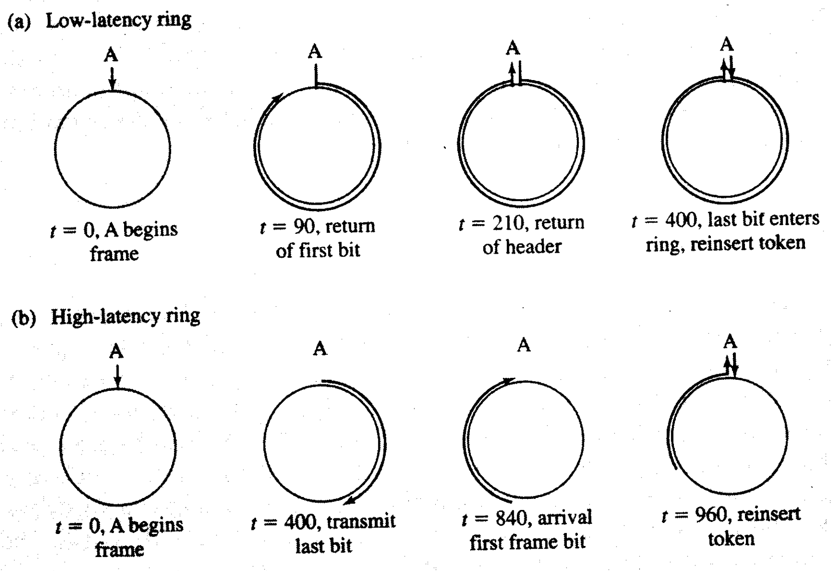

(a) Low-latency ring
A
t = 0, A begins frame
t = 90, return of first bit
t = 210, return of header
t = 400, last bit enters ring, reinsert token
(b) High-latency ring
t = 0, A begins frame
t = 400, transmit last bit
t = 840, arrival first frame bit
t = 960, reinsert token

FIGURE 6.60 Reinsert token after header of frame returns.

Finally suppose that the token reinsertion strategy had been to reinsert the token immediately after the frame transmission is completed (that is, multitoken operation). The need for the idle signal is completely eliminated and so is the associated inefficiency.

All three of the token reinsertion strategies introduced in the preceding example have been incorporated into token-ring LAN standards. The first strategy is part of the MAC protocol of the IEEE 802.5 standard for a 4 Mbps token-ring LAN. The reason for waiting until the last bit in the frame is that the last byte in the frame contains response information from the destination station. The IBM token-ring LAN for 4 Mbps uses the second strategy, where the token is reinserted after the header is returned. Both the IEEE 802.5 standard and the IBM token-ring LAN for 16 Mbps use the third strategy because of its higher efficiency. Each of the token reinsertion strategies has a different maximum achievable throughput leading to dramatic differences in frame transfer delay performance. These differences are discussed in Section 6.3.3.

Once the token has been reinserted into the ring, the token must travel to the next station that has a frame to transmit. The "walk" time that elapses from when the token is inserted to when it is captured by the next active station is also a form of overhead that can affect the maximum achievable throughput.

Finally, we note that different variations of MAC protocols are obtained according to how long a station is allowed to transmit once it captures a free token. One possibility is to allow a station to transmit only a single frame per token. This rather strict rule implies that each frame transmission is extended by a walk time. On the other hand, the rule also guarantees that a token will return to a station after at most M frame transmissions. At the other extreme, a station could be allowed to transmit until it empties its buffers of all frames. This approach is more efficient in that it amortizes the walk-time overhead over several frame transmissions. However, this approach also allows the token return time to grow without bound. An intermediate approach limits the time that a station can hold a token. For example, the IEEE 802.5 standard imposes a maximum token-holding-time limit of 10 ms.

6.8.2 Frame Structure

The structure of the token and data frames for the IEEE 802.5 standard is shown in Figure 6.61. The token frame consists of three bytes. The first and last bytes are the *starting delimiter (SD)* and *ending delimiter (ED)* fields. The standard uses differential Manchester line coding. Recall from Chapter 3 (Figure 3.35) that this line coding has transitions in the middle of each bit time. The SD and ED bytes are characterized by the fact that they contain symbols that violate this pattern: the J symbol begins as a 0 but has no transition in the middle; the K symbol begins as a 1 and has no transition in the middle. The second byte in the token frame is the *access control (AC)* field. The T bit in the access control field is the **token bit**: T = 0 indicates a token frame, and T = 1 indicates a data frame. A station can convert an available token frame (T = 0) into a data frame (T = 1) by simply flipping the T bit. This feature explains why token-ring

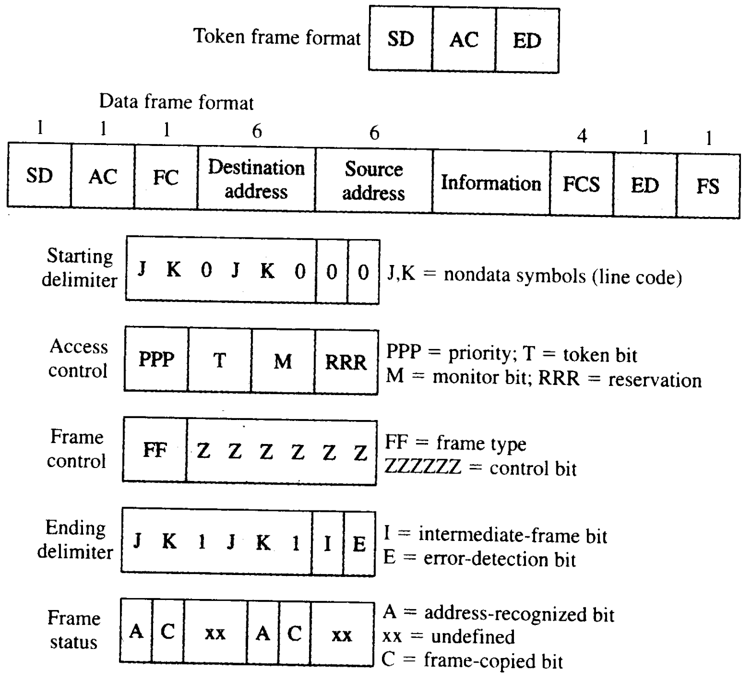


FIGURE 6.61 IEEE 802.5 Token and data frame structure.

interfaces can theoretically pass from an incoming link onto an outgoing link with only a one-bit delay (although a delay of 2.5 bits is usually implemented).

The data frame begins with SD and AC fields. The PPP and RRR bits in the AC field implement eight levels of priority in access to the ring. The monitor M bit is used by a designated monitor station to identify and remove "orphan" frames that are not removed from the ring by their sending station, for example, as a result of a station crash. The *frame control (FC)* field indicates whether a frame contains data or MAC information. Data frames are identified by FF = 01, and the Z bits are then ignored. MAC control frames are identified by FF = 00, and the Z bits then indicate the type of MAC control frame. Using the same format as for IEEE 802.3 Ethernet standard, the IEEE 802.5 standard specifies 48-bit addressing (16-bit addressing is also specified but is not used). The address fields are followed by the information field that is limited in length only by the maximum token holding time. The *frame check sequence (FCS)* field contains a CRC checksum as in IEEE 802.3. The ED field contains an E bit that indicates that a station interface has detected an error such as a line code violation or a frame check sequence error. The I bit indicates the last frame in a sequence of frames exchanged between two stations.

The *frame status (FS)* field in the data frame allows the receiving station to convey transfer status information to the sending station through A and C bits that are repeated within the field. An A = 1 bit indicates that the destination address was recognized by the receiving station. A C = 1 bit indicates that the frame was copied onto the receiving station's buffer. Therefore, an A = 1, C = 1 frame status field indicates that the frame was received by the intended destination station.

The IEEE 802.5 standard allows the token ring to be operated with a priority access mechanism. To transmit a frame of a given priority, a station must wait to capture a token of equal or lower priority. The station can reserve a token of the desired level by setting the RRR field in passing frames to the level of priority of its frame *if* the RRR level is *lower* than the priority the station is seeking. In effect, the RRR field allows stations to bid up the priority of the next token. When the token arrives at a station that has a frame of higher or equal priority, the token is removed and a data frame is inserted into the ring. The RRR field in the data frame is set to 0, and the priority field is kept at the same value as the token frame. When the station is done transmitting its frames, it issues a token at the reserved priority level.

Ring maintenance procedures are necessary to ensure the continued operation of the ring. Problem conditions can lead to the circulation of orphan data frames in the ring, the disappearance of tokens from the ring, the corruption of the frame structure within the ring, or the incidence of breaks in the link between stations. The IEEE 802.5 standard provides a procedure for the selection of a station to become the *active monitor* that is assigned the task of detecting and removing orphan frames, as well as identifying and replacing lost tokens. Additional procedures are defined to deal with other problem conditions. For example a "beacon" MAC control frame can be used by any station to determine whether its incoming link has become broken. Other MAC control frames indicate the presence of an active monitor, elect a new monitor, identify duplicate addresses, clear the ring of all frames, and identify neighbor stations in the ring.

6.9 FDDI

The **Fiber Distributed Data Interface (FDDI)** is a token-based LAN/MAN standard developed by the American National Standards Institute. FDDI uses a ring-topology network in which station interfaces are interconnected by optical fiber transmission links operating at 100 Mbps in a ring that spans up to 200 kilometers and accommodates up to 500 stations. FDDI has found application as a campus backbone network to interconnect various Ethernet LAN subnetworks.

FDDI can operate over multimode or single-mode optical fiber systems. FDDI can also operate over twisted-pair cable at lengths of less than 100 meters. The high bit rate in FDDI precludes the use of the bandwidth-inefficient Manchester line code. Instead FDDI uses a 4B5B binary line code and NRZ-inverted signaling that requires a symbol rate of 125 Msymbols/second. The 4B5B code lacks the self-clocking property of the Manchester code. For this reason FDDI frames begin with a longer preamble that serves to synchronize the receiver to the transmitter clock. Each FDDI station transmits into its outgoing link according to its own local clock. To accommodate differences between the local and incoming clock, each station uses a 10-bit elastic buffer to absorb timing differences. FDDI specifies that all clocks must meet a tolerance of 0.005 percent $= 5 \times 10^{-5}$ of 125 MHz. Assuming the maximum clock difference between a fast transmitter and a slow receiver, the worst-case clock difference is 0.01 percent. Symbols accumulate in the buffer at a rate of 125 Msymbols/second $\times 1 \times 10^{-4} = 12.5 \times 10^3$ bits/second. The five-bit buffer will then fill up in 5 bits/$(12.5 \times 10^3$ bps$) = 0.4$ ms. For this reason the maximum allowable FDDI frame length is 125 Msymbols/second $\times$ 0.4 ms $= 50{,}000$ symbols. The 4B5B code implies this is equivalent to 40,000 bits. Because of this timing consideration, the FDDI frame has a maximum size of 4500 bytes $= 36{,}000$ bits.

To provide reliability with respect to link breakages and interface failure, a dual-ring arrangement is used, as shown in Figure 6.62. A break in the ring is handled by

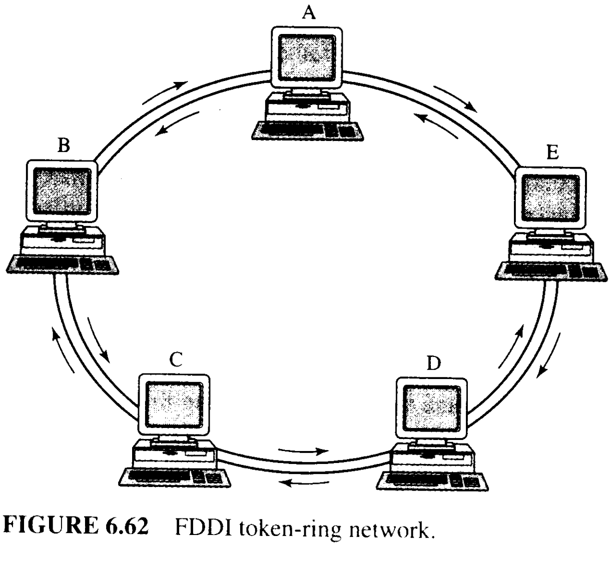


FIGURE 6.62 FDDI token-ring network.

Token frame format

Pre	SD	FC	ED

Data frame format

8	1	1	6	6		4	1	1
Pre	SD	FC	Destination address	Source address	Information	FCS	ED	FS

Preamble

Frame control CLFFZZZZ C = synch/asynch
L = address length (16 or 48 bits)
FF = LLC/MAC control/reserved frame type

FIGURE 6.63 FDDI frame structure.

redirecting the flow in the opposite direction at the last station before the break. This action has the effect of converting the dual ring into a single ring.

From Figure 6.63 we can see that the FDDI frame structure is very similar to that of IEEE 802.5. The frame begins with 16 or more idle control signals that generate a square wave signal that serves to synchronize the receiver. The SD and ED fields contain distinct signal violations that help identify them. The FDDI frame does not contain an AC field or a token bit. Instead the FC field is used to indicate the presence of a token and to provide information about the type of frame. A token frame is indicated by either 10000000 or 11000000 in the FC field. The capture of the token is done, not by flipping a bit, but by removing the token transmission from the ring and replacing it with a data frame. The other remaining fields function as in IEEE 802.5.

The FDDI medium access control was designed to operate in a high-ring-latency environment. If we assume 500 stations each introducing a latency of 10 bits and a maximum length 200 km ring, then the ring latency is $500 \times 10 + 100$ Mbps $\times (200 \times 10^3$ m$)/(2 \times 10^8$ m/sec$) = 5000 + 100{,}000 = 105{,}000$ bits. Thus FDDI can have a very high ring latency, making it essential that the token reinsertion strategy be immediate insertion after completion of each frame transmission. Even with this strategy, however, a maximum length ring can hold more than two maximum length frames! For this reason FDDI also provides an option of having more than one token circulating the ring at a given time.

The FDDI MAC protocol can handle two types of traffic: synchronous traffic that has a tight transfer delay requirement, such as voice or video, and asynchronous traffic that has a greater delay tolerance as in many types of data traffic. To meet the timing requirements of synchronous traffic, the medium access control uses a timed-token mechanism that ensures that the token rotation value is less than some value. In particular, all the stations in an FDDI ring must agree to operate according to a given **target token rotation time (TTRT)**. Each station i is allocated a certain amount of time, S_i seconds, that specifies the maximum duration the station is allowed to send synchronous traffic each time it has captured the token. If the sum of S_i times is smaller than the TTRT, then the operation of the FDDI medium access control guarantees that the token will return to every node in less than 2 TTRT seconds (1 TTRT for the transmission of all synchronous traffic and 1 TTRT for the token to travel around the ring back to a given node). This property allows FDDI to meet the delay requirements of synchronous traffic.

Each station maintains a **token rotation timer (TRT)** that measures the time that has elapsed since the station last received a token. When a station receives a token, the station first calculates the **token holding time (THT)**, defined by THT = TTRT − TRT, which is a measure of the degree of activity in the ring. When traffic is light, the token will rotate quickly around the ring and the TRT will be much smaller than the TTRT. In this case the medium access control need not be restrictive in providing access to the ring. As the traffic becomes heavy, the TRT will approach the TTRT, indicating that the medium access control must begin restricting access to the ring. The FDDI medium access control implements these notions as follows:

If THT > 0, then the station can transmit all its synchronous traffic S_i. In addition, if the THT timer has not expired after S_i seconds, the station is allowed to transmit asynchronous traffic for the balance of its THT time, that is, up to THT $- S_i$ seconds. The station must then release the token.

If THT < 0, then the station is allowed to transmit only its synchronous traffic S_i and must then release the token.

This timed-token mechanism throttles the asynchronous traffic when the ring becomes congested. In combination with the globally agreed upon TTRT, this mechanism assures the timely delivery of synchronous traffic. The timed-token mechanism also ensures fairness in access to the ring. Because the TRT is reset upon *arrival* of a token at a station, a station that has a lot of traffic to send will find it has a large TRT the next time it receives the token. Consequently, the THT will be small, and that station will be prevented from transmitting for an excessive period of time. Thus the station will be forced to relinquish the token sooner so other stations can have an opportunity to transmit.

6.10 WIRELESS LANS AND IEEE 802.11 STANDARD[16]

The case for wireless LANs is quite compelling. All you have to do is look under the desks in a typical small business or home office. You will find a rat's nest of wires: in addition to a variety of power cords and adapters, you have cables for a telephone modem, a printer, a scanner, a mouse, and a keyboard. In addition, there is the need for communications to synchronize files with laptop computers and personal organizers. And, of course, there is still the need to connect to other computers in the office. Wireless technology, in the form of digital radio and infrared transmission, can eliminate many of these wires and, in the process, simplify the installation and movement of equipment, as well as provide connectivity between computers. Wireless technology, however, must overcome significant challenges:

[16] The IEEE 802.11 is a complex standard, and so its explanation requires more space than the previous LAN standards. The discussion on physical layers for 802.11 can be skipped if necessary.

- Radio and infrared transmission is susceptible to noise and interference, so such transmission is not very reliable.
- The strength of a radio transmission varies in time and in space because of fading effects that result from multipath propagation and from uneven propagation due to physical barriers and geographic topology, and so coverage is inconsistent and unpredictable.
- In the case of radio, the transmitted signal cannot easily be contained to a specific area, so signals can be intercepted by eavesdroppers.
- The spectrum is finite and must be shared with other users (your neighbor's wireless LAN), and devices (e.g., microwave ovens and florescent lamps!).
- In the case of radio, the limited spectrum also makes it difficult to provide the high transmission speeds that are easily attained using wired media.
- Radio spectrum has traditionally been regulated differently by different government administrations, so it can be difficult to design products for a global market.

The most compelling reason for wireless networks, however, is that they enable *user mobility*. Many of us have already been conditioned to the convenience of television remote controls and cordless telephones in the "local" area of the home, as well as to the convenience of cellular phones over a "wider" area. In the context of wireless LANs, user mobility is particularly significant in situations where users carry portable computers or devices that need to communicate to a server or with each other. One example is a doctor or nurse in a hospital accessing up-to-date information on a patient, even as the patient is wheeled down a corridor. In this case the hospital may have an infrastructure of *wireless access points* that the portable devices can communicate with to access a backbone (wired) network. Another example is a meeting where the participants can create a temporary *ad hoc LAN* simply by turning on their laptop computers. To provide mobility, further challenges must be overcome:

- Mobile devices operate on batteries, so the MAC protocols must incorporate power management procedures.
- Protocols need to be developed that enable a station to discover neighbors in the local network and to provide seamless connections even as users roam from one coverage area to another.

The development of wireless LAN products was initially stimulated mildly by the allocation of spectrum in the industrial, scientific, and medical (ISM) bands of 902 to 928 MHz, 2400 to 2483.5 MHz, and 5725 to 5850 MHz in the United States under part 15 of the FCC rules. These rules, however, require that users accept interference from other users already using these frequencies, such as microwave ovens. Furthermore, users of the ISM band were required to use spread spectrum transmission techniques that limit the bit rates that can be attained. A stronger stimulus for the development of wireless LANs was the development of a HIPERLAN standard in Europe for a 20 Mbps wireless LAN operating in the 5 GHz band. This development was reinforced in the United States by the FCC's designation of 300 MHz of spectrum in the 5 GHz band

CSMA-CA? WHY NOT WIRELESS ETHERNET?

Given the dominance of the Ethernet standards in wired LANs, an obvious question is, Why not use wireless Ethernet? After all, Ethernet was designed for broadcast networks, and wireless networks are certainly broadcast in nature. There are several reasons why CSMA-CD cannot be used. The first reason is that it is difficult to detect collisions in a radio environment. Because the transmitted power would overwhelm the received power at the same station, it is not possible to abort transmissions that collide. A second reason is that the radio environment is not as well controlled as a wired broadcast medium, and transmissions from users in other LANs can interfere with the operation of CSMA-CD. A third reason is that radio LANs are subject to the *hidden-station problem* that occurs when two stations, say, A and C, attempt to transmit to a station that is located between them, say, B, as shown in Figure 6.64. The two stations may be sufficiently distant from each other that they cannot hear each other's transmission. Consequently, when they sense the channel, they may detect it as idle even as the other station is transmitting. This condition will result in the transmissions from the two stations proceeding and colliding at the intermediate station. The Carrier-Sense Multiple Access with Collision Avoidance (CSMA-CA) medium access control was developed to prevent this type of collision. CSMA-CA is incorporated in IEEE 802.11 and is described in the subsection on medium access control later in this section.

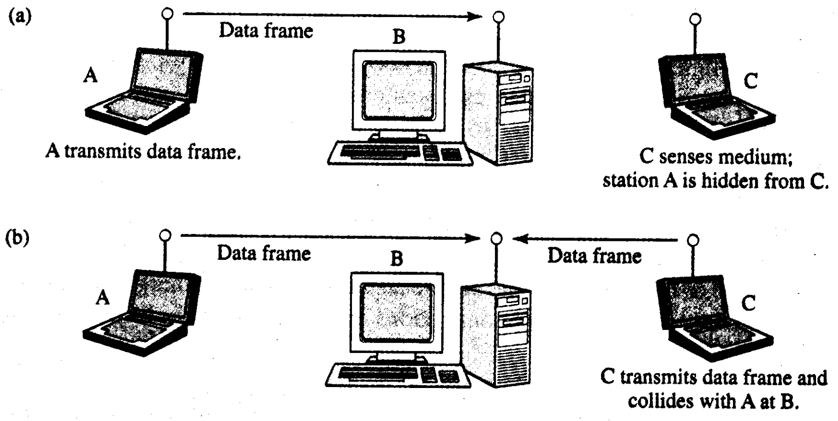

(a)
Data frame
B
A
C
A transmits data frame.
C senses medium; station A is hidden from C.
(b)
Data frame
B
Data frame
A
C
C transmits data frame and collides with A at B.

FIGURE 6.64 The hidden-station problem.

for the development of unlicensed LAN applications operating at speeds of 20 Mbps or higher. In addition, the Infrared Data Association (IrDA) has been promoting the development of MAC and physical layer standards for high-speed infrared systems for interconnecting computers to other devices at short range.

In this section we focus on the IEEE 802.11 LAN standard that specifies a MAC layer that is designed to operate over a number of physical layers. We show that the standard is quite complex. The standard's complexity is rooted in the challenges indicated above. In particular, we show that in addition to the basic issue of coordinating access, the standard must incorporate error control to overcome the inherent unreliability of the channel, modified addressing and association procedures to deal with station portability and mobility, and interconnection procedures to extend the reach of a wireless stations as well as to accommodate users who move while communicating.

6.10.1 Ad hoc and Infrastructure Networks

The **basic service set (BSS)** is the basic building block of the IEEE 802.11 architecture. A BSS is defined as a group of stations that coordinate their access to the medium under a given instance of the medium access control. The geographical area covered by the BSS is known as the *basic service area (BSA)*, which is analogous to a cell in a cellular communications network. A BSA may extend over an area with a diameter of tens of meters. Conceptually, all stations in a BSS can communicate directly with all other stations in a BSS. Note that two unrelated BSSs may be colocated. IEEE 802.11 provides a means for these BSSs to coexist.

A single BSS can be used to form an **ad hoc network**. An ad hoc network consists of a group of stations within range of each other. Ad hoc networks are typically temporary in nature. They can be formed spontaneously anywhere and be disbanded after a limited period of time. Figure 6.65 is an illustration of an ad hoc network. Two stations can make an ad hoc network.

In 802.11 a set of BSSs can be interconnected by a **distribution system (DS)** to form an **extended service set (ESS)** as shown in Figure 6.66. The BSSs are like cells in a cellular ncıwork. Each BSS has an **access point (AP)** that has station functionality and provides access to the DS. The AP is analogous to the base station in a cellular communications network. An ESS can also provide gateway access for wireless users into a wired network such as the Internet. This access is accomplished via a device

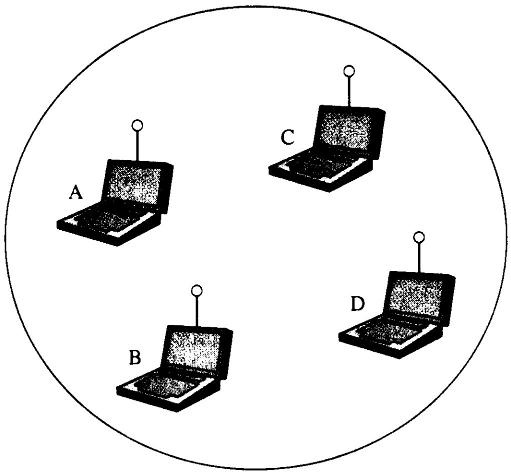


FIGURE 6.65 Ad hoc network.

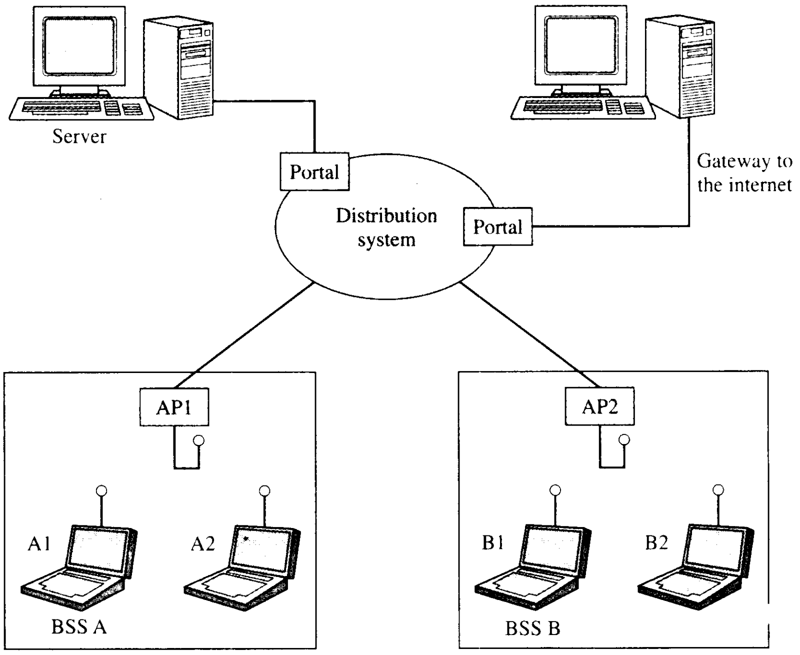


FIGURE 6.66 Infrastructure network and extended service set.

known as a **portal**. The term **infrastructure network** is used informally to refer to the combination of BSSs, a DS, and portals.

The distribution system provides the *distribution service,* which is

1. The transfer of MAC SDUs (MSDUs) between APs of BSSs within the ESS.
2. The transfer of MSDUs between portals and BSSs within the ESS.
3. The transport of MSDUs between stations in the same BSS when either the MSDU has a multicast or broadcast address or the sending station chooses to use the distribution service.

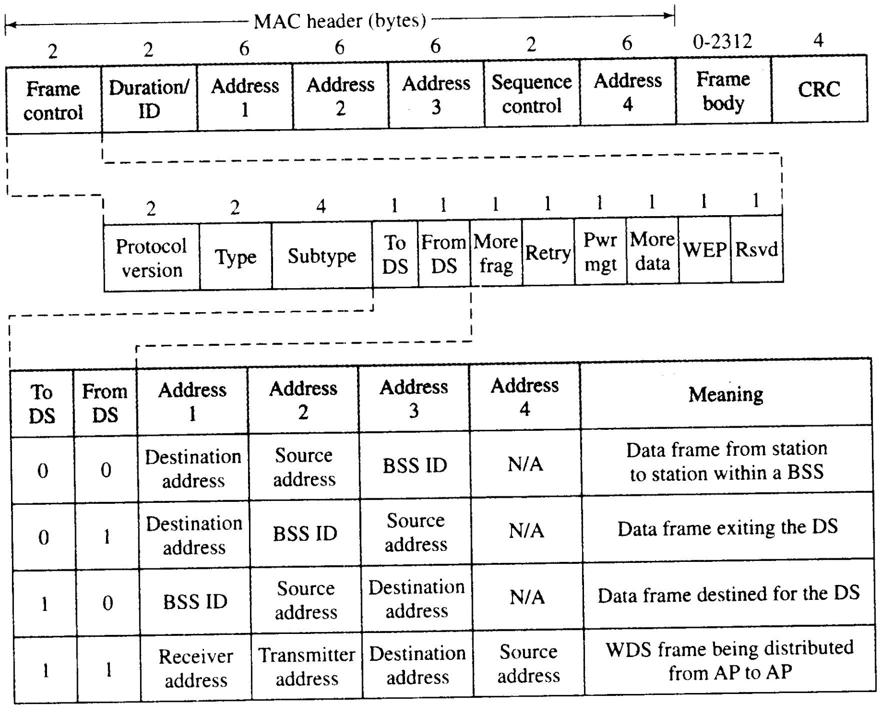
The role of the distribution service is to make the ESS appear as a single BSS to the LLC that operates above the medium access control in any of the stations in the ESS. IEEE 802.11 defines the distribution service but not the distribution system. The distribution system can be implemented by using wired or wireless networks.

To join an infrastructure BSS, a station must select an AP and establish an *association* with it, which is a mapping between the station and the AP that can be provided to the distribution system. The station can then send and receive data messages via the AP. A *reassociation* service allows a station with an established association to *move* its association from one AP to another AP. The *dissociation* service is used to terminate an existing association. Stations have the option of using an *authentication* service to establish the identity of other stations. Stations also have the option of using a *privacy* service that prevents the contents of messages from being read by anyone other than the intended recipient(s).

The dynamic nature of the LAN topologies under the scope of the IEEE 802.11 implies several fundamental differences between wireless and wired LANs. In wired LANs the MAC address specifies the physical location of a station, since users are stationary. In wireless LANs, the MAC address identifies the station but *not* the location, since the standard assumes that stations can be portable or mobile. A station is *portable* if it can move from one location to another but remains fixed while in use. A *mobile* station moves while in use. The 802.11 MAC sublayer is required to present the same set of standard services that other IEEE 802 LANs present to the LLC. This requirement implies that mobility has to be handled within the MAC sublayer.

6.10.2 Frame Structure and Addressing

IEEE 802.11 supports three types of frames: management frames, control frames, and data frames. The management frames are used for station association and disassociation with the AP, timing and synchronization, and authentication and deauthentication. Control frames are used for handshaking and for positive acknowledgments during the data exchange. Data frames are used for the transmission of data. The MAC header provides information on frame control, duration, addressing, and sequence control. Figure 6.67



2	2	6	6	6	2	6	0-2312	4
Frame control	Duration/ ID	Address 1	Address 2	Address 3	Sequence control	Address 4	Frame body	CRC

2	2	4	1	1	1	1	1	1	1	1
Protocol version	Type	Subtype	To DS	From DS	More frag	Retry	Pwr mgt	More data	WEP	Rsvd

To DS	From DS	Address 1	Address 2	Address 3	Address 4	Meaning
0	0	Destination address	Source address	BSS ID	N/A	Data frame from station to station within a BSS
0	1	Destination address	BSS ID	Source address	N/A	Data frame exiting the DS
1	0	BSS ID	Source address	Destination address	N/A	Data frame destined for the DS
1	1	Receiver address	Transmitter address	Destination address	Source address	WDS frame being distributed from AP to AP

DS = distribution system AP = access point

FIGURE 6.67 IEEE 802.11 frame structure.

shows that the format of the MAC frame consists of a MAC header, a frame body, and a CRC checksum.

The *frame control field* in the MAC header is 16 bits long, and it specifies the following items:

- The 802.11 protocol version (the current version is 0).
- The type of frame, that is, management (00), control (01), or data (10).
- The subtype within a frame type, for example, type = "management," subtype = "association request" or type = "control," subtype = "ACK."
- The To DS field is set to 1 in Data type frames destined for the DS, including Data type frames from a station associated with the AP that have broadcast or multicast addresses.
- The From DS field is set to 1 in Data type frames exiting the distribution system.
- The More Fragments field is set to 1 in frames that have another fragment of the current MSDU to follow.
- The Retry field is set to 1 in Data or Management type frames that are retransmissions of an earlier frame; this helps the receiver deal with duplicate frames.
- The Power Management bit is set to indicate the power management mode of a station.
- The More Data field is set to 1 to indicate to a station in power save mode that more MSDUs are buffered for it at the AP.
- The Wired Equivalent Privacy (WEP) field is set to 1 if the frame body field contains information that has been processed by the cryptographic algorithm.

The *Duration/ID field* in the MAC header is 16 bits long and is used in two ways. It usually contains a duration value (net allocation vector) that is used in the MAC protocol. The only exception is in Control type frames of subtype PS-Poll, where this field carries the ID of the station that transmitted the frame.

The use of the four *Address fields* is specified by the To DS and From DS fields in the Frame Control field as shown in Figure 6.67. Addresses are 48-bit-long IEEE 802 MAC addresses and can be individual or group (multicast/broadcast). The Address 1 field in this case contains the destination address. The *BSS identifier* (BSS ID) is a 48-bit field of the same format as IEEE 802 MAC addresses, uniquely identifies a BSS, and is given by the MAC address of the station in the AP of the BSS. The *destination address* is an IEEE MAC individual or group address that specifies the MAC entity that is the final recipient of the MSDU that is contained in the Frame Body field. The *source address* is a MAC individual address that identifies the MAC entity from which the MSDU originated. The *receiver address* is a MAC address that identifies the intended immediate recipient station for the MAC PDU (MPDU) in the Frame Body field. The *transmitter address* is a MAC individual address that identifies the station that transmitted the MPDU contained in the frame body field. This description is confusing so let's consider the four cases shown in the figure.

- To DS = 0, From DS = 0. This case corresponds to the transfer of a frame from one station in the BSS to another station in the same BSS. The stations in the BSS look at the Address 1 field to see whether the frame is intended for them. The Address 2 field contains the address that the ACK frame is to be sent to. The Address 3 field specifies the BSS ID.

- To DS = 0, From DS = 1. This case corresponds to the transfer of a frame from the DS to a station in the BSS. The stations in the BSS look at the Address 1 field to see whether the frame is intended for them. The Address 2 field contains the address that the ACK frame is to be addressed to, in this case the AP. The Address 3 field specifies the source address.
- To DS = 1, From DS = 0. This case corresponds to the transfer of a frame from a station in the BSS to the DS. The stations in the BSS, including the AP, look at the Address 1 field to see whether the frame is intended for them. The Address 2 field contains the address that the ACK frame is to be addressed to, in this case the source address. The Address 3 field specifies the destination address that the distribution system is to deliver the frame to.
- To DS = 1, From DS = 1. This special case applies when we have a *wireless distribution system* (WDS) transferring frames between BSSs. The Address 1 field contains the receiver address of the station in the AP in the WDS that is the next immediate intended recipient of the frame. The Address 2 field contains the destination address of the station in the AP in the WDS that is transmitting the frame and should receive the ACK. The Address 3 field specifies the destination address of the station in the ESS that is to receive the frame, and the Address 4 field specifies the source address of the station in the ESS that originated the frame.

The *Sequence Control field* is 16 bits long, and it provides 4 bits to indicate the number of each fragment of an MSDU and 12 bits of sequence numbering for a sequence number space of 4096. The *Frame Body field* contains information of the type and subtype specified in the Frame Control field. For Data type frames, the Frame Body field contains an MSDU or a fragment of an MSDU. Finally, the *CRC field* contains the 32-bit cyclic redundancy check calculated over the MAC header and Frame Body field.

6.10.3 Medium Access Control

The MAC sublayer is responsible for the channel access procedures, protocol data unit (PDU) addressing, frame formatting, error checking, and fragmentation and reassembly of MSDUs. The MAC layer also provides options to support security services through authentication and privacy mechanisms. MAC management services are also defined to support roaming within an ESS and to assist stations in power management.

The IEEE 802.11 MAC protocol is specified in terms of *coordination functions* that determine when a station in a BSS is allowed to transmit and when it may be able to receive PDUs over the wireless medium. The distributed coordination function (DCF) provides support for asynchronous data transfer of MSDUs on a best-effort basis. Under this function, the transmission medium operates in the *contention mode* exclusively, requiring all stations to contend for the channel for each packet transmitted. IEEE 802.11 also defines an optional point coordination function (PCF), which may be implemented by an AP, to support connection-oriented time-bounded transfer of MSDUs. Under this function, the medium can alternate between the **contention period (CP)**, during which the medium uses contention mode, and a **contention-free period (CFP)**. During the CFP, the medium usage is controlled by the AP, thereby eliminating the need for stations to contend for channel access.

DISTRIBUTED COORDINATION FUNCTION

The **distributed coordination function (DCF)** is the basic access method used to support asynchronous data transfer on a best-effort basis. All stations are required to support the DCF. The access control in ad hoc networks uses only the DCF. Infrastructure networks can operate using just the DCF or a coexistence of the DCF and PCF. The 802.11 MAC architecture is depicted in Figure 6.68, which shows that the DCF sits directly on top of the physical layer and supports contention services. Contention services imply that each station with an MSDU queued for transmission must contend for the channel and, once the given MSDU is transmitted, must recontend for the channel for all subsequent frames. Contention services are designed to promote fair access to the channel for all stations.

The DCF is based on the **carrier sensing multiple access with collision avoidance (CSMA-CA)** protocol. Carrier sensing involves monitoring the channel to determine whether the medium is idle or busy. If the medium is busy, it makes no sense for a station to transmit its frame end cause a collision and waste bandwidth. Instead the station should wait until the channel becomes idle. When this happens, there is another problem: Other stations may have also been waiting for the channel to become idle. If the protocol is to transmit immediately after the channel becomes idle, then collisions are likely to occur; and because collision detection is not possible, the channel will be wasted for an entire frame duration. A solution to this problem is to randomize the times at which the contending stations attempt to seize the channel. This approach reduces the likelihood of simultaneous attempts and hence the likelihood that a station can seize the channel.

Figure 6.69 shows the basic CSMA-CA operation. *All* stations are obliged to remain quiet for a certain minimum period after a transmission has been completed, called the **interframe space (IFS)**. The length of the IFS depends on the type of frame that the station is about to transmit. High-priority frames must only wait the **short IFS (SIFS)** period before they contend for the channel. Frame types that use SIFS include

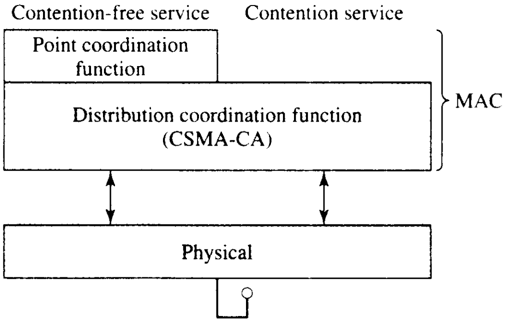

Contention-free service
Contention service
Point coordination function
Distribution coordination function (CSMA-CA)
MAC
Physical

FIGURE 6.68 IEEE 802.11 MAC architecture.

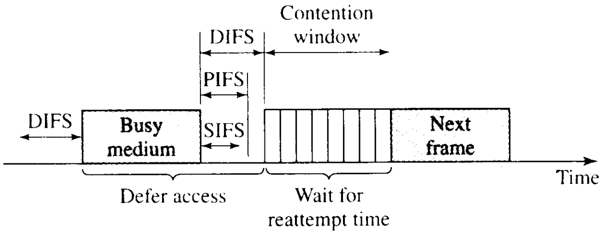


FIGURE 6.69 Basic CSMA-CA operation.

ACK frames, CTS frames, data frames of a segmented MSDU, frames from stations that are responding to a poll from an AP, and any frame from an AP during the CFP. All of these frame types complete frame exchanges that are already in progress. The **PCF interframe space (PIFS)** is intermediate in duration and is used by the PCF to gain priority access to the medium at the start of a CFP. The **DCF interframe space (DIFS)** is used by the DCF to transmit data and management MDPUs.

A station is allowed to transmit an *initial* MPDU under the DCF method if the station detects the medium idle for a period DIFS or greater. However, if the station detects the medium busy, then it must calculate a random backoff time to schedule a reattempt. A station that has scheduled a reattempt monitors the medium and decrements a counter each time an idle contention slot transpires. The station is allowed to transmit when its backoff timer expires during the contention period. If another station transmits during the contention period before the given station, then the backoff procedure is suspended and resumed the next time a contention period takes place. When a station has successfully completed a frame transmission and has another frame to transmit, the station must first execute the backoff procedure. Stations that had already been contending for the channel tend to have smaller remaining backoff times when their timers are resumed, so they tend to access the medium sooner than stations with new frames to transmit. This behavior introduces a degree of fairness in accessing the channel.

A handshake procedure was developed to operate with CSMA-CA when there is a hidden-station problem. Figure 6.70 shows that if a station, say, A, wants to send a data frame to station B, station A first sends a **request-to-send (RTS)** frame. If station B receives the RTS frame, then B issues a **clear-to-send (CTS)** frame. *All* stations within range of B receive the CTS frame and are aware that A has been given permission to send, so they remain quiet while station A proceeds with its data frame transmission. If the data frame arrives without error, station B responds with an ACK. In this manner CSMA-CA coordinates stations, even in the presence of hidden stations, so that collisions are avoided. It is still possible for two stations to send RTS frames at the same time so that they collide at B. In this case the stations must execute a backoff to schedule a later attempt. Note that having RTS frames collide is preferable to having data frames collide, since RTS frames are much shorter than data frames. For example, RTS is 20 bytes and CTS is 14 bytes, whereas an MPDU can be 2300 bytes long.

In IEEE 802.11, carrier sensing is performed at both the air interface, referred to as *physical carrier sensing*, and at the MAC sublayer, referred to as *virtual carrier sensing*. Physical carrier sensing detects the presence of other IEEE 802.11 stations by analyzing all detected frames and also detects activity in the channel via relative signal strength from other sources. Virtual carrier sensing is used by a source station to inform

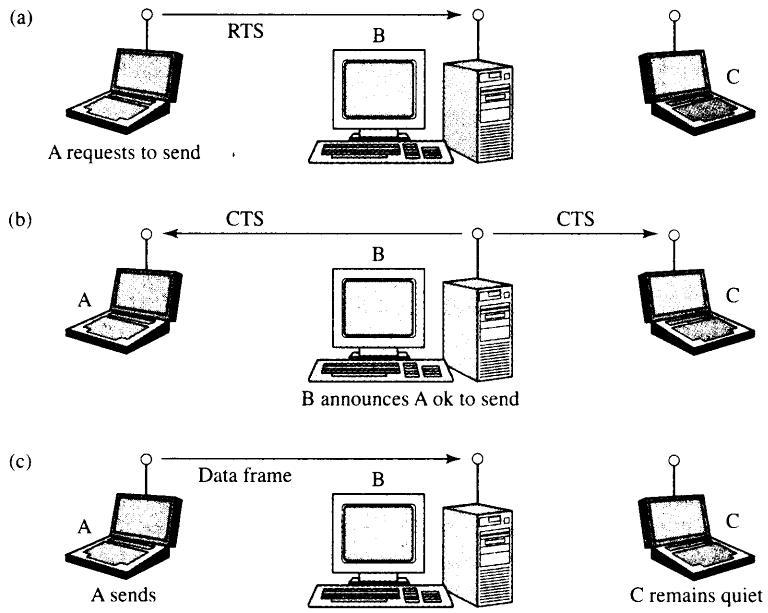


FIGURE 6.70 CSMA-CA.

all other stations in the BSS of how long the channel will be utilized for the successful transmission of a MPDU. The source stations set the *duration field* in the MAC header of data frames or in RTS and CTS control frames. The duration field indicates the amount of time (in microseconds) after the end of the present frame that the channel will be utilized to complete the successful transmission of the data or management frame. Stations detecting a duration field in a transmitted MSDU adjust their **network allocation vector (NAV)**, which indicates the amount of time that must elapse until the current transmission is complete and the channel can be sampled again for idle status. The channel is marked busy if either the physical or virtual carrier-sensing mechanism indicates that the channel is busy.

Figure 6.71 is a timing diagram that illustrates the successful transmission of a data frame. When the data frame is transmitted, the duration field of the frame lets all stations in the BSS know how long the medium will be busy. All stations hearing the data frame adjust their NAV based on the duration field value, which includes the SIFS interval and the acknowledgment frame following the data frame.

Figure 6.72 illustrates the transmission of an MPDU using the RTS/CTS mechanism. Stations can choose to never use RTS/CTS, to use RTS/CTS whenever the MSDU exceeds the value of RTS_Threshold (which is a manageable parameter), or to always use RTS/CTS. If a collision occurs with an RTS or CTS MPDU, far less bandwidth is wasted in comparison to a large data MPDU. However, for a lightly loaded medium the overhead of the RTS/CTS frame transmissions imposes additional delay.

Wireless channels cannot handle very long transmissions due to their relatively large error rates. Large MSDUs handed down from the LLC to the medium access control may require fragmentation to increase transmission reliability. To determine

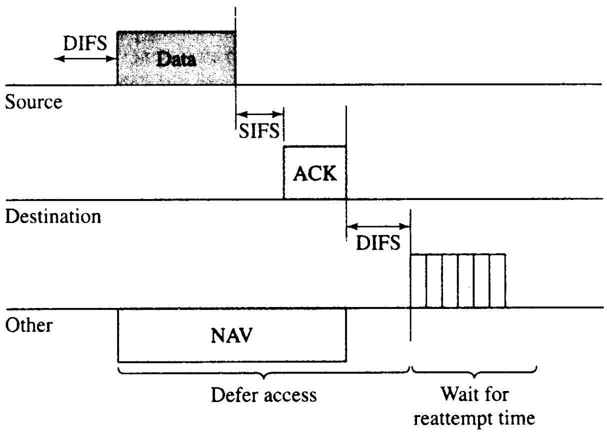


FIGURE 6.71 Transmission of MPDU without RTS/CTS.

whether to perform fragmentation, MDPUs are compared to the manageable parameter, Fragmentation_Threshold. If the MPDU size exceeds the value of Fragmentation_Threshold, then the MSDU is broken into multiple fragments.

The collision avoidance portion of CSMA-CA is performed through a random backoff procedure. If a station with a frame to transmit initially senses the channel to be busy, then the station waits until the channel becomes idle for a DIFS period and then computes a random backoff time. For IEEE 802.11 time is slotted in time periods that correspond to a Slot_Time. The Slot_Time used in IEEE 802.11 is much smaller than an MPDU and is used to define the IFS intervals and to determine the backoff time for stations in the CP. The random backoff time is an integer value that corresponds to a number of time slots. Initially, the station computes a backoff time uniformly in the range 0 to 7. When the medium becomes idle after a DIFS period, stations decrement their backoff timer until either the medium becomes busy again or the timer reaches zero. If the timer has not reached zero and the medium becomes busy, the station freezes

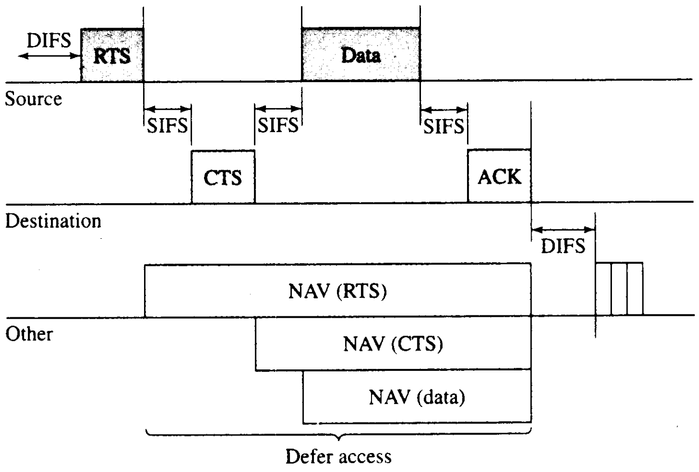


FIGURE 6.72 Transmission of MPDU with RTS/CTS.

its timer. When the timer is finally decremented to zero, the station transmits its frame. If two or more stations decrement to zero at the same time, then a collision will occur and each station will have to generate a new backoff time in the range 0 to 15. For each retransmission attempt, the number of available backoff slots grows exponentially as 2^{2+i}. The idle period after a DIFS period is referred to as the contention window (CW).

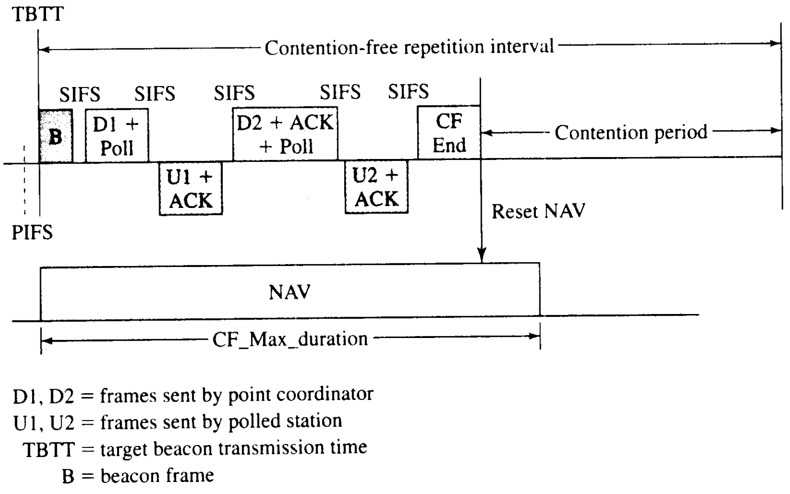
The operation of the DCF includes mechanisms for dealing with lost or errored frames. Receiving stations are required to transmit an ACK frame if the CRC of the frame they receive is correct. The sending station expects an ACK frame and interprets the failure to receive such a frame as an indication of loss of the frame. Note, however, that the lack of an ACK frame may also be due to loss of the ACK frame itself, not the original data frame. The sending station maintains an ACK_Timeout, equal to an ACK frame time plus a SIFS, for each data frame. If the ACK is not received, the station executes the backoff procedure to schedule a reattempt time. Receiver stations use the sequence numbers in the frame to detect duplicate frames. 802.11 does not provide MAC-level recovery for the broadcast of multicast frames except when these frames are sent with the To_DS bit set.

POINT COORDINATION FUNCTION

The **point coordination function (PCF)** is an optional capability that can be used to provide connection-oriented, contention-free services by enabling polled stations to transmit without contending for the channel. The PCF function is performed by the *point coordinator* (PC) in the AP within a BSS. Stations within the BSS that are capable of operating in the CFP are known as *CF-aware stations*. The method by which polling tables are maintained and the polling sequence is determined by the PC is left to the implementor.

The PCF is required to coexist with the DCF and logically sits on top of the DCF (see Figure 6.68). The *CFP repetition interval* (CFP_Rate) determines the frequency with which the PCF occurs. Within a repetition interval, a portion of the time is allotted to contention-free traffic, and the remainder is provided for contention-based traffic. The CFP repetition interval is initiated by a *beacon frame*, where the beacon frame is transmitted by the AP. One of the AP's primary functions is synchronization and timing. The duration of the CFP repetition interval is a manageable parameter that is always an integer-multiple number of beacon frames. Once the CFP_Rate is established, the duration of the CFP is determined. The maximum size of the CFP is determined by the manageable parameter, CFP_Max_Duration. At a minimum, time must be allotted for at least one MPDU to be transmitted during the CP. It is up to the AP to determine how long to operate the CFP during any given repetition interval. If traffic is very light, the AP may shorten the CFP and provide the remainder of the repetition interval for the DCF. The CFP may also be shortened if DCF traffic from the previous repetition interval carries over into the current interval. The maximum amount of delay that can be incurred is the time it takes to transmit an RTS/CTS handshake, maximum MPDU, and an acknowledgment. Figure 6.73 is a sketch of the CFP repetition interval, illustrating the coexistence of the PCF and DCF.

At the nominal beginning of each CFP repetition interval, the so-called target beacon transmission time (TBTT), all stations in the BSS update their NAV to the maximum length of the CFP (i.e., CFP_Max_Duration). During the CFP, stations may



D1, D2 = frames sent by point coordinator
U1, U2 = frames sent by polled station
TBTT = target beacon transmission time
B = beacon frame

FIGURE 6.73 Point coordination frame transfer.

transmit only to respond to a poll from the PC or to transmit an acknowledgment one SIFS interval after receipt of an MPDU. At the nominal start of the CFP, the PC senses the medium. If the medium remains idle for a PIFS interval, the PC transmits a beacon frame to initiate the CFP. In case the CFP is lightly loaded, the PC can foreshorten the CFP and provide the remaining bandwidth to contention-based traffic by issuing a CF-End or CF-End + ACK control frame. This action causes all stations that receive the frame in the BSS to reset their NAV values.

RTS/CTS frames are not used by the point coordinator or by CF-aware stations during the CFP. After the PC issues a poll, the intended CF-aware station may transmit one frame to *any* station as well as piggyback an ACK of a frame received from the PC by using the appropriate subtypes of a Data type frame. When a frame is transmitted to a non-CF-aware station, the station sends its ACK using DCF rules. The PC keeps control of the medium by only waiting the PIFS duration before proceeding with its contention-free transmissions.

◆ 6.10.4 Physical Layers

The IEEE 802.11 LAN has several physical layers defined to operate with its MAC layer. Each physical layer is divided into two sublayers that correspond to two protocol functions as shown in Figure 6.74. The **physical layer convergence procedure (PLCP)** is the upper sublayer, and it provides a convergence function that maps the MPDU into a format suitable for transmission and reception over a given physical medium. The **physical medium dependent (PMD)** sublayer is concerned with the characteristics and methods for transmitting over the wireless medium.

Figure 6.74 shows that the MPDU is mapped into a PLCP frame that consists of three parts. The first part is a preamble that provides synchronization and start-of-frame information. The second part is a PLCP header that provides transmission bit rate and

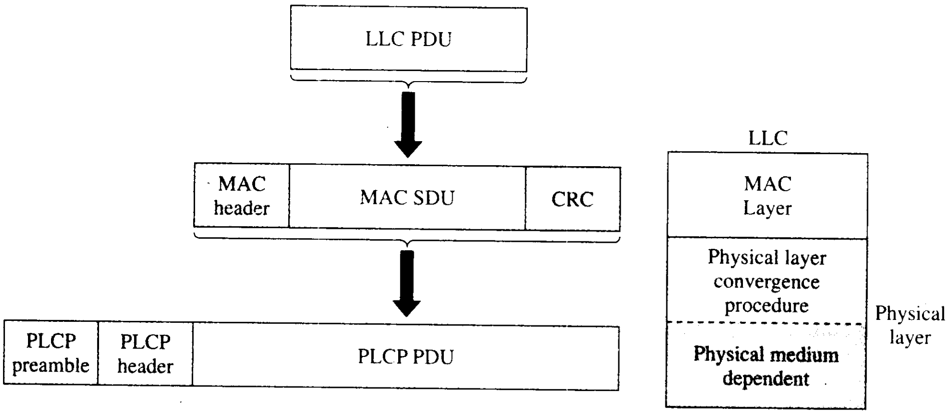


FIGURE 6.74 IEEE 802.11 physical layer has two sublayers, PLCP and PMD.

other initialization information as well as frame-length information and a CRC. The third part consists of the MPDU possibly modified (scrambled) to meet requirements of the transmission system. The specific structure of each PLCP depends on the particular physical layer definition. We next discuss three original physical layers that have been defined for IEEE 802.11, followed by recent extensions.

FREQUENCY-HOPPING SPREAD SPECTRUM FOR THE 2.4 GHZ ISM BAND

Spread spectrum transmission is a form of digital modulation technique that takes a data signal of certain bit rate and modulates it onto a transmitted signal of much larger bandwidth. In essence, spread spectrum systematically spreads the energy of the data signal over a wide frequency band.[17] The spread spectrum receiver uses its knowledge of how the spreading was done to compress the received signal and recover the original data signal. Spread spectrum provides great robustness with respect to interference as well as other transmission impairments such as fading that results from multipath propagation. Frequency hopping is one type of spread spectrum technique.

Frequency hopping involves taking the data signal and modulating it so that the modulated signal occupies different frequency bands as the transmission progresses. It is analogous to transmitting a song over a large number of FM radio channels. To recover the signal, the receiver must know the sequence of channels that it should tune to as well as the "dwell" time in each channel. The obvious question is, Why not give each user its own dedicated channel? One reason is that multipath fading affects narrow frequency bands so some of the channels exhibit very poor transmission. Frequency hopping minimizes the time spent on each channel.

The 802.11 frequency-hopping physical layer standard uses 79 nonoverlapping 1 MHz channels to transmit a 1 Mbps data signal over the 2.4 GHz ISM band. An option provides for transmission at a rate of 2 Mbps. This band occupies the range 2400 to 2483.5 MHz, providing 83.5 MHz of bandwidth. A channel hop occurs every

[17] Spread spectrum technique is used in CDMA, which is discussed in Section 6.4.3.

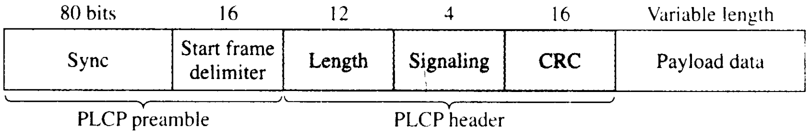


FIGURE 6.75 Frequency-hopping spread spectrum PLCP frame format.

224 microseconds. The standard defines 78 hopping patterns that are divided into three sets of 26 patterns each. Each hopping pattern jumps a minimum of six channels in each hop, and the hopping sequences are derived via a simple modulo 79 calculation. The hopping patterns from each set collide three times on the average and five times in the worst case over a hopping cycle. Each 802.11 network must use a particular hopping pattern. The hopping patterns allow up to 26 networks to be colocated and still operate simultaneously.

Figure 6.75 shows the format of the PLCP frame. The PLCP preamble starts with 80 bits of 0101 synchronization pattern that the receiver uses to detect the presence of a signal and to acquire symbol timing. The preamble ends with a 16-bit start frame delimiter that consists of the pattern 0000 1100 1011 1101. The PLCP header consists of a 12-bit PLCP_PDU length indicator that allows for PLCP total lengths of up to 4095 bytes. The PLCP header also contains a four-bit field in which the first three bits are reserved and the last bit indicates operation at 1 Mbps or 2 Mbps. The last 16 bits of the PLCP header are a 16-bit CRC using the CCITT-16 generator polynomial that covers the preceding 16 bits in the header. The PLCP header is always transmitted at the base rate of 1 Mbps. The PLCP_PDU is formed by scrambling the binary sequence of the MPDU and converting it into the sequence of symbols that are suitable for the frequency shift keying modulation scheme that is used in the frequency hopping.

When we discussed the 802.11 medium access control, we found that the operation depended on the values of certain key time parameters. In Table 6.6 we show the default values of some of the key parameters for the frequency-hopping physical layer.

TABLE 6.6 Default time parameters in IEEE 802.11 frequency-hopping spread spectrum physical layer.

Parameter	Value μsec	Definition
Air propagation time	1	Time for transmitted signal to go from transmitter to receiver.
RxTx turnaround time	20	Time for a station to transmit a symbol after request from MAC.
CCA assessment time	29	Time for the receiver to determine the state of the channel.
Slot time	50	Time used by MAC to determine PIFS and DIFS periods = CCA assessment + RxTx turnaround + air propagation.
SIFS time	28 +2/−3	Time required by MAC and physical sublayers to receive the last symbol of a frame at the air interface, process the frame, and respond with the first symbol of a preamble on the air interface.
Preamble length	96	Time to transmit the PLCP preamble.
PLCP header	32	Time required to transmit the PLCP header.

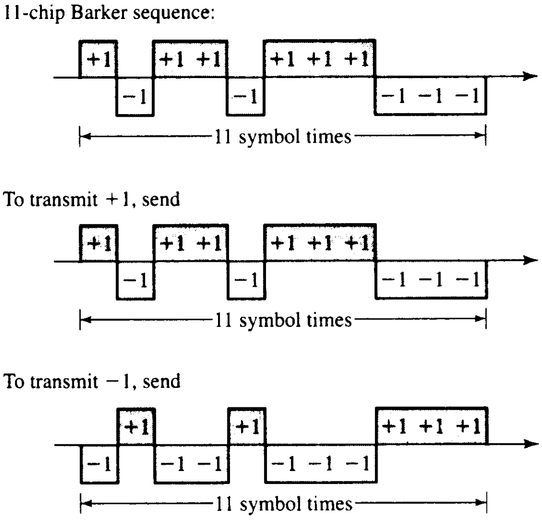


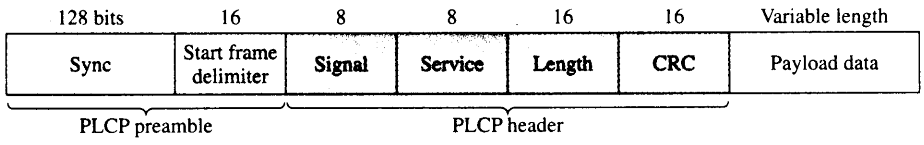
FIGURE 6.76 Direct sequence spread spectrum using 11-chip Barker sequence.

DIRECT SEQUENCE SPREAD SPECTRUM FOR THE 2.4 GHZ ISM BAND

Direct sequence spread spectrum (DSSS) is another method for taking a data signal of a given bit rate and modulating it into a signal that occupies a much larger bandwidth. DSSS represents each data 0 and 1 by the symbols −1 and +1 and then multiplies each symbol by a binary pattern of +1s and −1s to obtain a digital signal that varies more rapidly and hence occupies a larger frequency band. The IEEE 802.11 DSSS physical layer uses a particularly simple form as shown in Figure 6.76: Each binary data bit results in the transmission of plus or minus the polarity of the 11-chip Barker sequence. The term *chip* is used to distinguish the time required to transmit a +1 or −1 signal element from the time required to transmit a data bit (= 11 chip times). The Barker sequence provides good immunity against interference and noise as well as some protection against multipath propagation.

The DSSS transmission system in 802.11 takes the 1 Mbps data signal and converts it into an 11 Mbps signal using binary phase-shift keying (BPSK) modulation. Eleven channels have been defined to operate in the 2.4 GHz ISM band in the United States. Nine channels have been defined to operate in the 2.4 GHz band in Europe. Channels can operate without interfering with each other if their center frequencies are separated by at least 30 MHz. The 802.11 DSSS physical layer also defines an option for 2 Mbps operation using quaternary PSK (QPSK).

Figure 6.77 shows the format of the PLCP frame. The PLCP preamble starts with 128 scrambled bits of synchronization that the receiver uses to detect the presence of



128 bits	16	8	8	16	16	Variable length
Sync	Start frame delimiter	Signal	Service	Length	CRC	Payload data

FIGURE 6.77 Direct sequence spread spectrum PLCP frame format.

TABLE 6.7 Default time parameters in IEEE 802.11 direct sequence spread spectrum physical layer.

Parameter	Value μsec	Definition
RxTx turnaround time	<5	Time for a station to transmit a symbol aftur request from MAC.
CCA assessment time	<15	Time for the receiver to determine the state of the channel.
Slot time	20	Time used by MAC to determine PIFS and DIFS periods = CCA assessment + RxTx turnaround + air propagation.
SIFS time	10	Time required by MAC and physical sublayers to receive the last symbol of a frame at the air interface, process the frame, and respond with the first symbol of a preamble on the air interface.
Preamble length	144 bits	Time to transmit the PLCP preamble.
PLCP header	48 bits	Time required to transmit the PLCP header.

a signal. The preamble ends with a 16-bit start frame delimiter (hF3A0) that is used for bit synchronization. The PLCP header consists of an 8-bit signal field that indicates to the physical layer the modulation that is to be used for transmission and reception of the MPDU (h0A for 1 Mbps BPSK, h14 for 2 Mbps QPSK); an 8-bit service field that is reserved for future use; a 16-bit field that indicates the number of bytes in the MPDU, from 4 to 2^{16}; and a 16-bit CRC using the CCITT-16 generator polynomial. The PLCP header is always transmitted at the base rate of 1 Mbps.

The 802.11 MAC operation depends on the values of certain key time parameters. In Table 6.7 we show the default values of some of the key parameters for the DSSS physical layer.

INFRARED PHYSICAL LAYER

The IEEE 802.11 infrared physical layer operates in the near-visible light range of 850 to 950 nanometers. Diffuse transmission is used so the transmitter and receivers do not have to point to each other. The transmission distance is limited to the range 10 to 20 meters, and the signal is contained by walls and windows. This feature has the advantage of isolating the transmission systems in different rooms. The system cannot operate outdoors.

The transmission system uses pulse-position modulation (PPM) in which the binary data is mapped into symbols that consist of a group of slots. The 1 Mbps data system uses a 16 L-PPM slot that uses "symbols" that consist of 16 time slots and in which only 1 slot can contain a pulse. A slot is 250 nanoseconds in duration. The modulation takes four data bits to determine an integer in the range 1 to 16, which determines the corresponding symbol. The 2 Mbps data system uses a 4 L-PPM in which groups of two data bits are mapped into symbols that consist of four slots.

Figure 6.78 shows the format of the PLCP frame. The PLCP preamble starts with a minimum of 57 and a maximum of 73 L-PPM slots of alternating presence and absence of pulse in consecutive slots, which must terminate in the absence of pulse. The receiver uses this sequence to perform slot synchronization and other optional initialization procedures. The preamble ends with a 4 L-PPM slot start frame delimiter

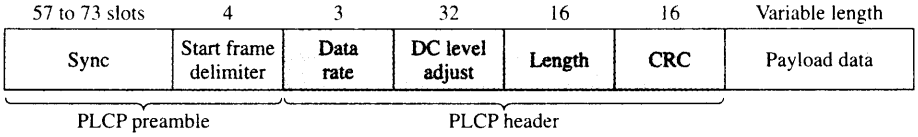


FIGURE 6.78 Infrared PLCP frame format.

(1001) to indicate the start of frame and to perform bit and symbol synchronization. The PLCP header consists of a 3 L-PPM slot data rate field to indicate the data rate (000 for 1 Mbps; 001 for 2 Mbps); a 32 L-PPM slot sequence of pulses that stabilizes the DC level of the received signal; a 16-bit integer (modulated using L-PPM) that indicates the length of the PSDU; and a 16-bit CRC calculated over the length field using the CCITT-16 generator polynomial and modulated using L-PPM. The PSDU field consists of 0 to 2500 octets modulated using the L-PPM format. The PLCP length, CRC, and PSDU fields can be transmitted at either 1 Mbps or 2 Mbps. The fields prior to these are defined in terms of slots, not symbols.

The 802.11 MAC operation depends on the values of certain key time parameters. In Table 6.8 we show the default values of some of the key parameters for the infrared physical layer.

PHYSICAL-LAYER EXTENSIONS

The original physical layer, which provides 1 Mbps and 2 Mbps data rates, was perceived as being too slow for many applications. To support high data rates, the IEEE later defined a physical-layer extension for high-rate DSSS operating at the 2.4 GHz band, which is specified in IEEE 802.11b (also known as "WiFi" in the industry). Instead of using the Barker sequence to encode each data bit into an 11-chip Barker sequence as in the original 802.11, the 802.11b uses *complementary code keying (CCK)* consisting of 64 eight-chip code words. Each of the 64 code words can be used to encode six data bits. The CCK code word can be modulated with 2-Mbps QPSK to support a data rate of 11 Mbps.

TABLE 6.8 Default time parameters in IEEE 802.11 infrared physical layer.

Parameter	Value μsec	Definition
RxTx turnaround time	0	Time for a station to transmit a symbol after request from MAC.
CCA assessment time	5	Time for the receiver to determine the state of the channel.
Slot time	6	Time used by MAC to determine PIFS and DIFS periods = CCA assessment + RxTx turnaround + air propagation
SIFS time	7	Time required by MAC and physical sublayers to receive the last symbol of a frame at the air interface, process the frame, and respond with the first symbol of a preamble on the air interface.

The IEEE also defined another extension, called 802.11a, which operates at the 5 GHz band and supports data rates up to 54 Mbps. The modulation scheme in the 802.11a system uses orthogonal frequency-division multiplexing consisting of eight nonoverlapping 20-MHz channels. Each channel in turn is divided into 52 subcarriers that are modulated using BPSK, QPSK, or QAM. Because the 802.11a uses a completely different modulation scheme than the original 802.11 or the 802.11b, this implies that the 802.11a will not interoperate with the other physical layers.

BLUETOOTH, WIRELESS 802.11 LANS, AND 3G NETWORKS

These three emerging wireless standards cover distinct but overlapping applications. The Bluetooth standard was developed as a low-cost wireless replacement for the wiring that interconnects telephones, computers, peripheral devices, and personal devices such as PDAs, electronic games, and DVD players. The Bluetooth physical layer operates in the 2.4 GHz band and uses frequency-hopping spread spectrum across 79 frequencies that are 1 MHz apart to provide full duplex transmission at speeds up to 700 kbps between devices separated by up to 10 meters. The Bluetooth standards enable multiple devices to be interconnected into a so-called personal area network. For example, a PDA could be continuously connected via Bluetooth to a laptop computer, which in turn may be connected to an 802.11 wireless LAN in airport lounge. Information can flow continuously from PDA to the laptop and on to the Internet, for example, in the exchange of e-mail, electronic calendar transactions, or game moves. When the user transfers to a taxi, the PDA and laptop can continue to be interconnected to the Internet from the laptop to a 3G cellular data network. The connectivity can return to a wireless LAN when the user enters her residence or business. Neat huh?

6.11 LAN BRIDGES AND ETHERNET SWITCHES

There are several ways of interconnecting networks. When two or more networks are interconnected at the physical layer, the type of device is called a **repeater**. When two or more networks are interconnected at the MAC or data link layer, the type of device is called a **bridge**.[18] When two or more netwoıks are interconnected at the network layer, the type of device is called a **router**. Interconnection at higher layers is done less frequently. The device that interconnects networks at a higher level is usually called a **gateway**. Gateways usually perform some protocol conversion and security functions. In this section we focus on bridges.

When range extension is the only problem, repeaters may solve the problem as long as the maximum distance between two stations is not exceeded. Local area net-

[18] The term "LAN bridge" that is found in standards is often referred to as "LAN switch" in the industry. In this book the two are synonymous.

works (LANs) that involve sharing of media, such as Ethernet and token ring, can only handle up to some maximum level of traffic. As the number of stations in the LAN increases, or as the traffic generated per station increases, the amount of activity in the medium increases until it reaches a saturation point. As we saw earlier in this chapter, the point at which saturation occurs depends on the particular MAC protocol as well as the ratio *a* of delay-bandwidth product to frame size. Figure 6.51 (page 429) shows a typical performance curve for a LAN system. Since the collision domain of LANs connected through repeaters is the entire network, repeaters do not solve the LAN saturation problem. An approach to improve the saturation problem is to segment the entire network into multiple collision domains. Each domain consists of a group of stations in a single LAN, and a bridge interconnects multiple LANs to form a **bridged LAN** or an **extended LAN**.

Another typical scenario involves large organizations in which LANs are initially introduced by different departments to meet their particular needs. Eventually, the need arises to interconnect these departmental LANs to enable the exchange of information and the sharing of certain resources. This scenario is frequently complicated by the following factors:

1. The departmental LANs use different network layer protocols that are packaged with the applications that they require.
2. The LANs may be located in different buildings.
3. The LANs differ in type.

These three requirements can be met by bridges. Because bridges exchange frames at the data link layer, the frames can contain any type of network layer PDUs. If necessary, bridges can be connected by point-to-point links. However, we have seen that the MAC PDUs do differ in structure in operation and in the size of the frames they allow, and so at the very least, the third requirement involves some form of frame conversion process. By extending the LAN, bridges provide the plug-and-play convenience of operating a single LAN. However, bridges need to deal with security and broadcast storm concerns.

LANs originally assume an element of trust between the users in the LAN. As a LAN grows, this assumption breaks down, and security concerns become prominent. The fact that most LANs are broadcast in nature implies that eavesdropping can be done easily by operating the NIC in *promiscuous mode* where every frame in the LAN is captured and examined. This behavior opens the door for the various security threats that are discussed in Chapter 11. Bridges can contribute to this problem by extending the reach of users. However, bridges can also help deal with security problems because of their ability to *filter* frames. By examining the contents of frame and packet headers, bridges can control the flow of traffic allowed in and out of a given LAN segment.

LANs inherently involve the broadcasting of frames in the shared medium. A problem in all LANs is that a faulty station may become stuck sending broadcast traffic, and this situation can bring down the entire LAN. The problem becomes more severe with the introduction of bridges that can potentially distribute these broadcast frames over all segments of the LAN.

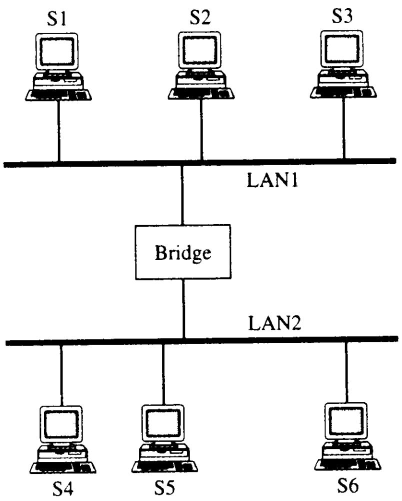


FIGURE 6.79 A bridged LAN.

Consider two LANs connected by a bridge, as shown in Figure 6.79. When station 1 transmits a frame to station 3 (local traffic), the frame is broadcast only on LAN1. The bridge can prevent the signal from propagating to LAN2 because it knows that station 3 is connected to the same LAN as station 1. If station 1 transmits a frame to a remote station, say, station 5, then both LAN1 and LAN2 will be busy during the frame transmission. Thus if most traffic is local, the load on each LAN will be reduced. In contrast, if the bridge is replaced with a repeater, both LANs will be busy when a station is transmitting a frame, independent of where the destination station is.

To have a frame filtering capability, a bridge has to monitor the MAC address of each frame. For this reason, a bridge cannot work with physical layers. On the other hand, a bridge does not perform a routing function, which is why a bridge is a layer 2 relay. Because bridges are mostly used for extending LANs of the same type, they usually operate at the MAC layer as shown in Figure 6.80.

Two types of bridges are widely used: *transparent bridges* and *source routing bridges*. Transparent bridges are typically used in Ethernet LANs, whereas source routing bridges are typically used in token-ring and FDDI networks.

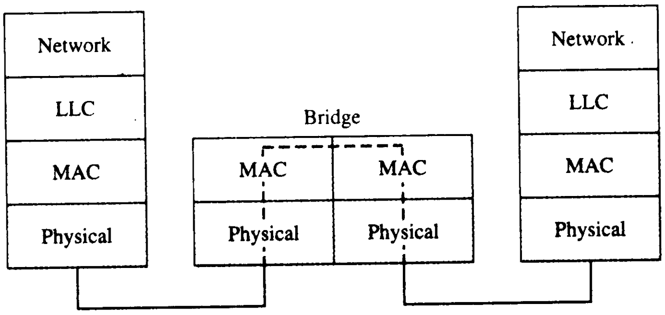


FIGURE 6.80 Interconnection by a bridge.

6.11.1 Transparent Bridges

Transparent bridges were defined by the IEEE 802.1d committee. The term *transparent* refers to the fact that stations are completely unaware of the presence of bridges in the network. Thus introducing a bridge does not require the stations to be reconfigured. A **transparent bridge** performs the following three basic functions:

1. Forwards frames from one LAN to another.
2. Learns where stations are attached to the LAN.
3. Prevents loops in the topology.

A transparent bridge is configured in a promiscuous mode so that each frame, independent of its destination address, can be received by its MAC layer for further processing. Ethernet switches are simply multiport transparent bridges for interconnecting stations using Ethernet links.

BRIDGE LEARNING

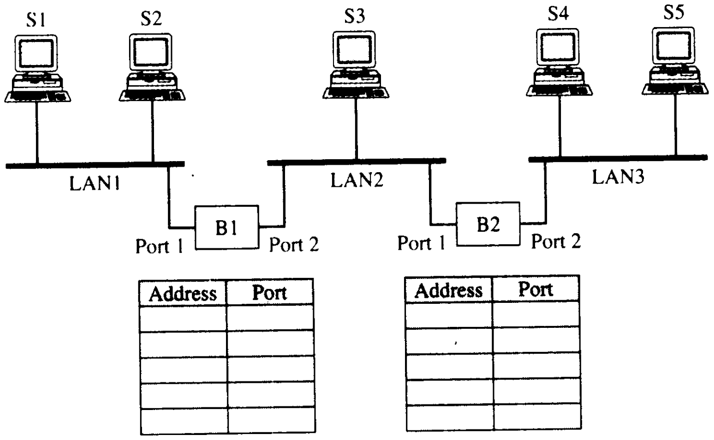
When a frame arrives on one of its ports, the bridge has to decide whether to forward the incoming frame to another port based on the destination address of the frame. To do so, the bridge needs a table to indicate which side of the port the destination station is attached to, whether indirectly or directly. The table is called a *forwarding table*, or *forwarding database*, and associates each station address with a port number, as shown in Table 6.9. In practice, the table can have a few thousand entries to handle an interconnection of multiple LANs.

The question then is how the entries in the forwarding table are filled. One way is to have the network administrator record these entries and load them up during system startup. Although this approach is theoretically possible, it is not desirable in practice, as it requires the system administrator to change the entry manually when a station is moved from one LAN to another and when a new station is added or removed. It turns out that there is a simple and elegant way for a bridge to "learn" the location of the stations as it operates and to build the forwarding table automatically.

We first look at the basic learning process that is used by the bridge. Further improvements are needed to handle the dynamics of the network. The basic process works as follows. When a bridge receives a frame, the bridge first compares the source address of the frame with each entry in the forwarding table. If the bridge does not find a match, it adds to the forwarding table the *source address* together with the port number on which the frame was received. The bridge then compares the *destination address* of the frame with each entry in the forwarding table. If the bridge finds a match, it forwards the frame to the port indicated in the entry; however, if the port is the one on which the frame was received, no forwarding is required and the frame is discarded.

TABLE 6.9 Forwarding table (with no data).

MAC address	Port



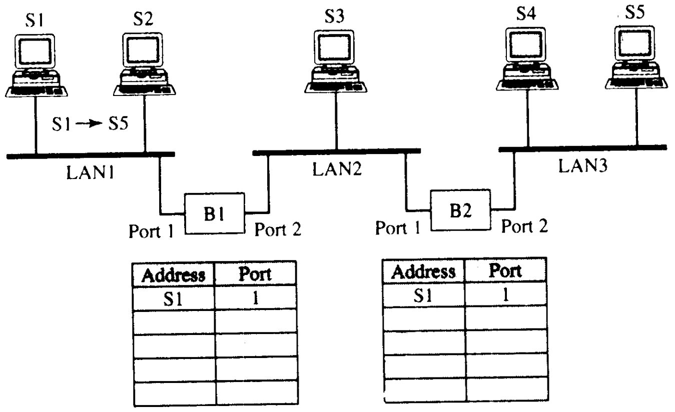
Address	Port

Address	Port

FIGURE 6.81 Bridge learning: initial configuration.

If the bridge does not find a match, the bridge "floods" the frame on all ports except the one on which the frame was received.

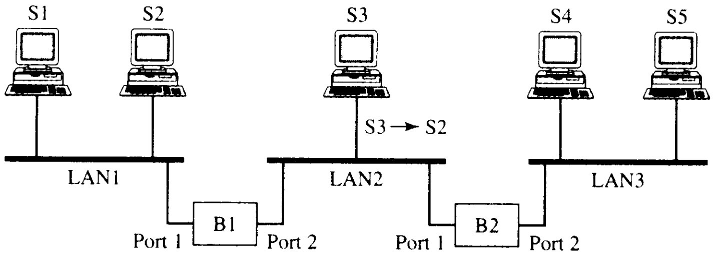
To see how bridges use this procedure to learn station locations, consider an example of a bridged LAN comprising three LANs, as shown in Figure 6.81. Assume that the forwarding tables are initially empty. Suppose now S1 (station 1) sends a frame to S5. The frame carries the MAC address of S5 as the destination address and the MAC address of S1 as the source address. When B1 (bridge 1) receives the frame, it finds the table empty and adds S1's source address and the port number on which the frame arrived (which is port 1). The destination address is also not found in the table, and so the frame is forwarded to port 2 and transmitted on LAN2. When B2 receives the frame, it performs the same process, adding the source address and forwarding the frame to LAN3. S5 eventually receives the frame destined to it. Figure 6.82 shows the current state of both forwarding tables. Both bridges have learned the location of S1.



Address	Port
S1	1

Address	Port
S1	1

FIGURE 6.82 S1 sends a frame to S5 allowing bridges 1 and 2 to 'learn' the location of S1.



Address	Port
S1	1
S3	2

Address	Port
S1	1
S3	1

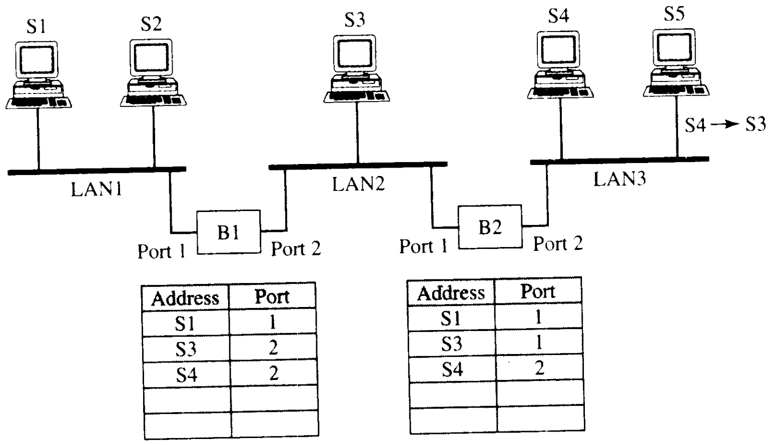
FIGURE 6.83 S3 sends a frame to S2 and another address is added. Frame is sent to all three LANs as B2 still does not know where S2 is.

Next S3 sends a frame to S2. Both B1 and B2 receive the frame, since they are connected to the same LAN as S3. B1 cannot find the address of S3 in its table, so it adds (S3, 2) to its table. It then forwards the frame through port 1, which S2 finally receives. B2 also does not find the source address, adding the new information in its table and forwarding the frame on LAN 3. The traffic on LAN 3 is wasted, since the destination is located on the opposite side. But at this point the bridges are still in the learning process and need to accumulate more information to make intelligent decisions. At the end of this process, the forwarding tables gain one more entry (see Figure 6.83).

Now assume that S4 sends a frame to S3. First B2 records the address of S4 and the port number on which the frame arrived, since the address of S4 is not found. Then B2 checks the destination address of the frame in the forwarding table. The destination address of the frame matches one of the entries, so the bridge forwards the frame to the port indicated in the entry (which is port 1). When B1 receives the frame, it adds the source address and the port number on which the frame arrived into the forwarding table. The bridge, however, finds the destination address. Because the port number in the entry is the same as that on which the frame arrived, the frame is discarded and not transmitted to LAN1. Thus the traffic is confined to LAN2 and LAN3 only. Figure 6.84 shows the forwarding tables after this point.

Now assume that S2 sends a frame to S1. B1 first adds the address of S2 in its forwarding table. Since the bridge has learned the address of S1, it discards the frame after finding out that S1 is also connected to the same port. We see that the traffic now is completely isolated in LAN1. At the same time, transmission can occur on different LANs without interfering with each other, thus increasing the *aggregate* throughput. Note also that because the frame is not transmitted to LAN2, B2 cannot learn the address of S2, as indicated by its forwarding table in Figure 6.85.

It is now obvious that if the learning process continues indefinitely, both tables eventually store the address of each station in the bridged LAN. At this point the bridges stop learning. Unfortunately, nothing stays static in real life. For example,

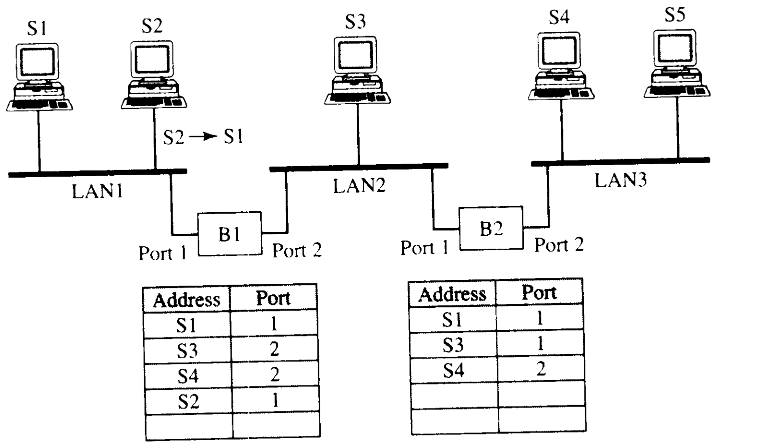


Address	Port
S1	1
S3	2
S4	2

Address	Port
S1	1
S3	1
S4	2

FIGURE 6.84 S4 sends a frame to S3; traffic is confined to LAN2 and LAN3 as S3's address is already learned.

stations may be added to a LAN or moved to another LAN. To have a bridge that can adapt to the dynamics of the network, we need two additional minor changes. First, the bridge adds a timer associated with each entry. When the bridge adds an address to its table, the timer is set to some value (typically on the order of a few minutes). The timer is decremented periodically. When the value reaches zero, the entry is erased so that a station that has been removed from the LAN will eventually have its address removed from the table as well. When the bridge receives a frame and finds that the source address of the frame matches with the one in the table, the corresponding entry is "refreshed" so that the address of an active station will be retained in the table. Second, the bridge could update address changes quickly by performing the following simple task. When the bridge receives a frame and finds a match in the source address but the



Address	Port
S1	1
S3	2
S4	2
S2	1

Address	Port
S1	1
S3	1
S4	2

FIGURE 6.85 S2 sends a frame to S1; as both addresses are now known, the frame is discarded by B1 thus isolating the traffic to LAN1.

port number in the entry is different from the port number on which the frame arrived, the bridge updates the entry with the new port number. Thus a station that has moved to another LAN will be updated as soon as it transmits.

SPANNING TREE ALGORITHM

The learning process just described works as long as the network does not contain any loops, meaning that there is only one path between any two LANs. In practice, however, loops may be created accidentally or intentionally to increase redundancy. Unfortunately, loops can be disastrous during the learning process, as each frame from the flooding triggers the next flood of frames, eventually causing a *broadcast storm* and bringing down the entire network.

To remove loops in a network, the IEEE 802.1 committee specified an algorithm called the *spanning tree algorithm.* If we represent a network with a graph, a spanning tree maintains the connectivity of the graph by including each node in the graph but removing all possible loops. This is done by automatically disabling certain bridges. It is important to understand that these bridges are not physically removed, since a topology change may require a different set of bridges to be disabled, thus reconfiguring the spanning tree dynamically.

The spanning tree algorithm requires that each bridge have a unique bridge ID, each port within a bridge have a unique port ID, and all bridges on a LAN recognize a unique MAC group address. Together, bridges participating in the spanning tree algorithm carry out the following procedure:

1. Select a *root bridge* among all the bridges in the bridged LAN. The root bridge is the bridge with the lowest bridge ID.
2. Determine the *root port* for each bridge except the root bridge in the bridged LAN. The root port is the port with the least-cost path to the root bridge. In case of ties the root port is the one with lowest port ID. Cost is assigned to each LAN according to some criteria. One criterion could be to assign higher costs to lower speed LANs. A path cost is the sum of the costs along the path from one bridge to another.
3. Select a *designated bridge* for each LAN. The designated bridge is the bridge that offers the least-cost path from the LAN to the root bridge. In case of ties the designated bridge is the one with the lowest bridge ID. The port that connects the LAN and the designated bridge is called a *designated port.*

Finally, all root ports and all designated ports are placed into a "forwarding" state. These are the only ports that are allowed to forward frames. The other ports are placed into a "blocking" state.

EXAMPLE Creating a Spanning Tree

To see how a spanning tree is created by the procedure described above, consider the topology of a bridged LAN shown in Figure 6.86. For simplicity the bridges are identified by B1, B2, . . . , B5, and each port ID is indicated by the number in parentheses. Costs assigned to each LAN are assumed to be equal.

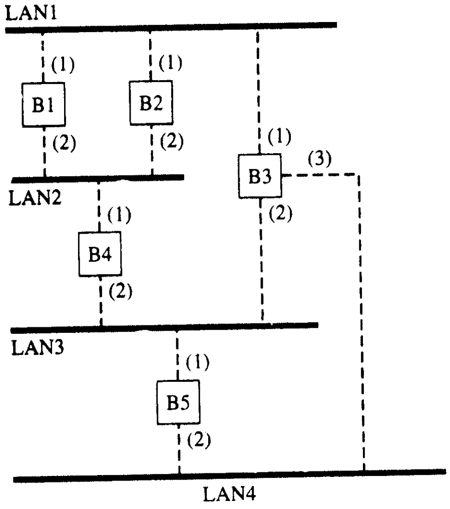


FIGURE 6.86 Sample topology of a bridged LAN.

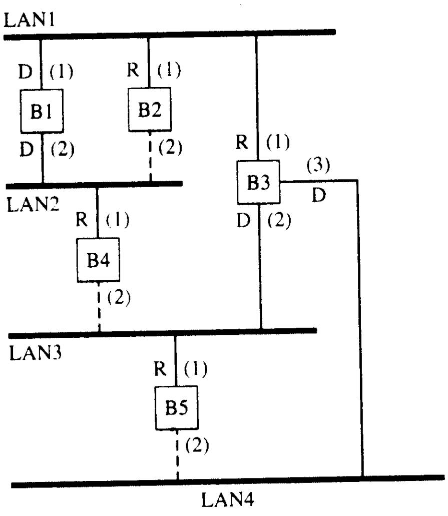


FIGURE 6.87 The corresponding spanning topology.

The resulting spanning tree configuration is shown in Figure 6.87. First bridge 1 is selected as the root bridge, since it has the lowest bridge ID. Next the root port is selected for each bridge except for B1, as indicated by the letter *R*. Then the designated bridge is selected for each LAN, as indicated by the letter *D* on the corresponding designated port. Finally, the root ports and the designated ports are put into a forwarding state, as indicated by the solid lines. The broken lines represent the ports that are in a blocking state. You should verify that the final topology contains no loops.

The procedure to discover a spanning tree can be implemented by using a distributed algorithm. To do so, each bridge exchanges special messages called *configuration bridge protocol data units* (configuration BPDUs). A configuration BPDU contains the bridge ID of the transmitting bridge, the root bridge ID, and the cost of the least-cost path from the transmitting bridge to the root bridge. Each bridge records the best configuration BPDU it has so far. A configuration BPDU is "best" if it has the lowest root bridge ID. If there is a tie, the configuration BPDU is best if it has the least-cost path to the root bridge. If there is still a tie, the configuration BPDU is best if it has the lowest bridge ID of the transmitting bridge.

Initially, each bridge assumes that it is the root bridge and transmits configuration BPDUs periodically on each of its ports. When a bridge receives a configuration BPDU from a port, the bridge adds the path cost to the cost of the LAN that this BPDU was received from. The bridge then compares the configuration BPDU with the one recorded. If the bridge receives a better configuration BPDU, it stops transmitting on that port and saves the new configuration BPDU. Eventually, only one bridge on each LAN (the designated bridge) will be transmitting configuration BPDUs on that LAN, and the algorithm stabilizes.

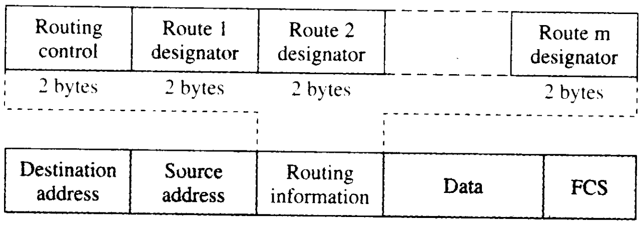


FIGURE 6.88 Frame format for source routing.

To detect a bridge failure after the spanning tree is discovered, each bridge maintains an aging timer for the saved configuration BPDU, which is incremented periodically. The timer is reset when the bridge receives a configuration BPDU. If a designated bridge fails, one or more bridges will not receive the configuration BPDU. When the timer expires, the bridge starts the algorithm again by assuming that it is the root bridge, and the distributed algorithm should eventually configure a new spanning tree.

6.11.2 Source Routing Bridges

Source routing bridges were developed by the IEEE 802.5 committee and are primarily used to interconnect token-ring networks. Unlike transparent bridges that place the implementation complexity in bridges, source routing bridges put the burden more on the end stations. The main idea of source routing is that each station should determine the route to the destination when it wants to send a frame and therefore include the route information in the header of the frame. Thus the problem boils down to finding good routes efficiently.

A source routing frame introduces additional routing information in the frame, as shown in Figure 6.88. The routing information field is inserted only if the two communicating stations are on different LANs. The presence of the routing information field is indicated by the individual/group address (I/G) bit in the source address field. If the routing information field is present, the I/G bit is set to 1.[19] If a frame is sent to a station on the same LAN, that bit is 0. The routing control field defines the type of frame, the length of the routing information field, the direction of the route given by the route designator fields (from left to right or right to left), and the largest frame supported over the path. The route designator field contains a 12-bit LAN number and a 4-bit bridge number.

As an example, if S1 (station 1) wants to send a frame to S2 (Figure 6.89), then a possible route is LAN1→B1→LAN2→B4→LAN4. The bridge number in the final route designator field is not used. Many more routes are available for this source-destination pair, making it possible to share the load among several routes or to choose

[19] The I/G bit was originally defined to indicate a multicast source address. As the bit has never been used for that purpose, the IEEE 802.5 committee decided to use it for indicating the presence of a routing information field.

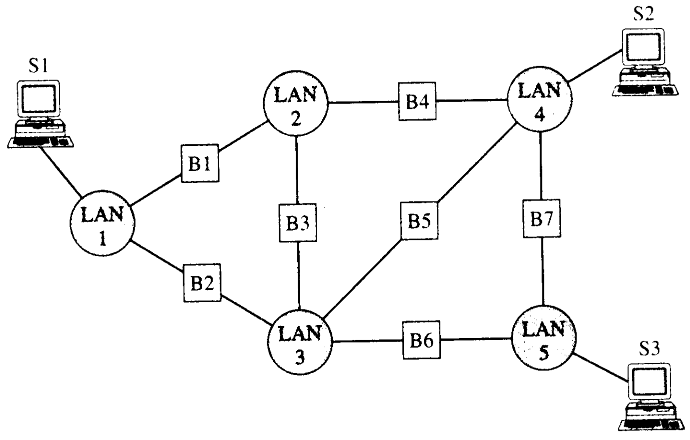


FIGURE 6.89 LAN interconnection with source routing bridges.

an alternative route if one should fail. In general, when a station wants to transmit a frame to another station on a different LAN, the station consults its routing table. If the route to the destination is found, the station simply inserts the routing information into the frame. Otherwise, the station performs a route discovery procedure. Once the route is found, the station adds the route information to its routing table for future use.

The basic idea for a station to discover a route is as follows. First the station broadcasts a special frame, called the *single-route broadcast* frame. The frame visits every LAN in the bridged LAN exactly once, eventually reaching the destination station. Upon receipt of this frame, the destination station responds with another special frame, called the *all-routes broadcast* frame, which generates all possible routes back to the source station. After collecting all routes, the source station chooses the best route and saves it.

The detailed route discovery procedure is as follows. First the source station transmits the single-route broadcast frame on its LAN without any route designator fields. To ensure that this frame appears on each LAN exactly once, selected bridges are configured to form a spanning tree, which can be done manually or automatically. When a selected bridge at the first hop receives a single-route broadcast frame, that bridge inserts an incoming LAN number, its bridge number, and the outgoing LAN number to the routing information field and forwards the frame to the outgoing LAN. When a selected bridge at other hops receives a single-route broadcast frame, the bridge inserts its bridge number and the outgoing LAN number to the routing information field and then forwards the frame to the outgoing LAN. Nonselected bridges simply ignore the single-route broadcast frame. Because the spanning tree maintains the full connectivity of the original topology, one frame should eventually reach the destination station.

Upon receipt of a single-route broadcast frame, the destination station responds with an all-routes broadcast frame containing no route designator fields. When a bridge at the first hop receives an all-route broadcast frame, the bridge inserts an incoming LAN number, its bridge number, and the outgoing LAN number to the routing information field and forwards the frame to the outgoing LAN. Other bridges insert their bridge number and the outgoing LAN number to the routing information field and forward the frame to the outgoing LAN. To prevent all-routes broadcast frames from circulating in the network, a bridge first checks whether the outgoing LAN number is already recorded in the route designator field. The bridge will not forward the frame if the outgoing LAN number is already recorded. The all-routes broadcast received by the source station eventually should list all possible routes to the destination station.

EXAMPLE Determining Routes for Broadcast Frames

Consider a bridged LAN as shown in Figure 6.89. Assume that B1, B3, B4, and B6 are part of the spanning tree. Suppose S1 wants to send a frame to S3 but has not learned the route yet. Sketch the routes followed by the single-route and all-routes broadcast frames during route discovery.

Figure 6.90 shows the routes followed by single-route broadcast frames. Figure 6.91 shows the trajectory of the all-routes broadcast frames that originate from LAN5. Each possible route from S1 to S3 starts from LAN1 and goes back to LAN5. There are a total of seven possible routes.

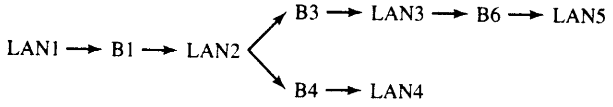


FIGURE 6.90 Routes followed by single-route broadcast frames.

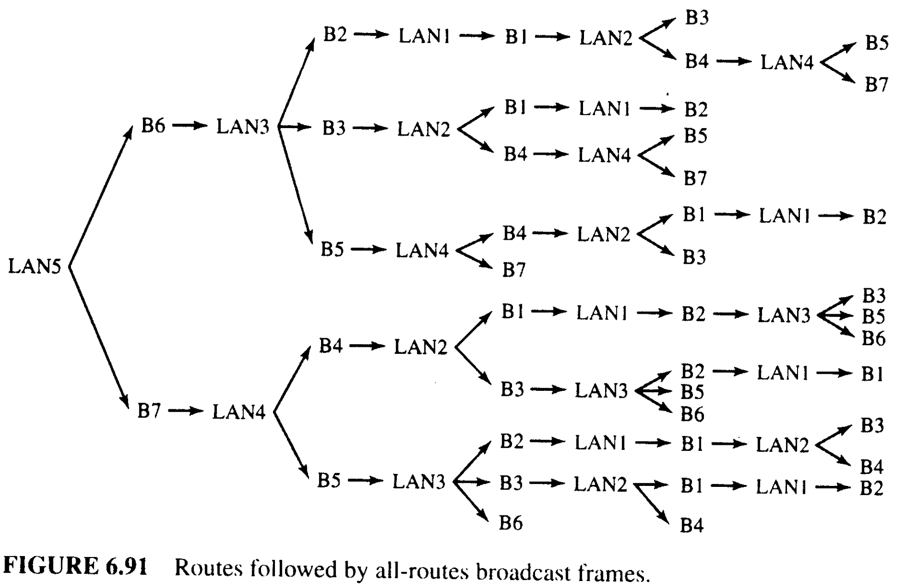


FIGURE 6.91 Routes followed by all-routes broadcast frames.

6.11.3 Mixed-Media Bridges

Bridges that interconnect LANs of different type are referred to as **mixed-media bridges**. This type of interconnection is not simple. We discuss mixed-media bridges in terms of the interconnection of Ethernet and token-ring LANs. These two LANs differ in their frame structure, their operation, and their speeds, and the bridge needs to take these differences into account.

Both Ethernet and token-ring LANs use six-byte MAC addresses but they differ in the hardware representation of these addresses. Token rings consider the first bit in a stream to be the high-order bit in a byte. Ethernet considers such a bit to be the low-order bit. A bridge between these two must convert between the two representations.

Ethernet and token-ring LANs differ in terms of the maximum size frames that are allowed. Ethernet has an upper limit of approximately 1500 bytes, whereas token ring has no explicit limit. Bridges do not typically include the ability to do frame fragmentation and reassembly, so frames that exceed the maximum limit are just dropped.

Token ring has the three status bits, A and C in the frame status field and E in the end delimiter field. Recall from Figure 6.61 that the A and C bits indicate whether the destination address is recognized and whether the frame is copied. The E bit indicates errors. Ethernet frames do not have corresponding fields to carry such bits. There is no clear solution on how to handle these bits in either direction, from Ethernet to token ring and from token ring to Ethernet. Similar questions arise regarding what to do with the monitor bit, the reservation, and the priority bits that are present in the token-ring header.

Another problem in interconnecting LANs is that they use different transmission rates. The bridge that is interposed between two LANs must have sufficient buffering to be able to absorb frames that may accumulate when a burst of frames arriving from a fast LAN is destined to a slow LAN.

Two approaches to bridging between transparent (Ethernet) bridging domains and source routing (token ring) domains have been proposed. *Translational bridging* carries out the appropriate reordering of address bits between Ethernet and token-ring formats. It also provides approaches for dealing with differences in maximum transfer unit size and for handling status bits. *Source-route transparent bridging* combines source route and transparent operations in a bridge that can forward frames produced from transparent and source route nodes.

6.11.4 Virtual LANs

An enterprise or a building with many stations typically contains separate LANs that are connected through routers. The stations are traditionally partitioned into separate LANs based on physical constraints such as location of a station and connectivity to a bridge or hub. A natural partition would be to create a separate LAN on each floor in a building. Unfortunately, such a physical association between a station and a LAN is not flexible. For example, rewiring is necessary when a station is physically moved to another floor but needs to maintain the association with the previous LAN.

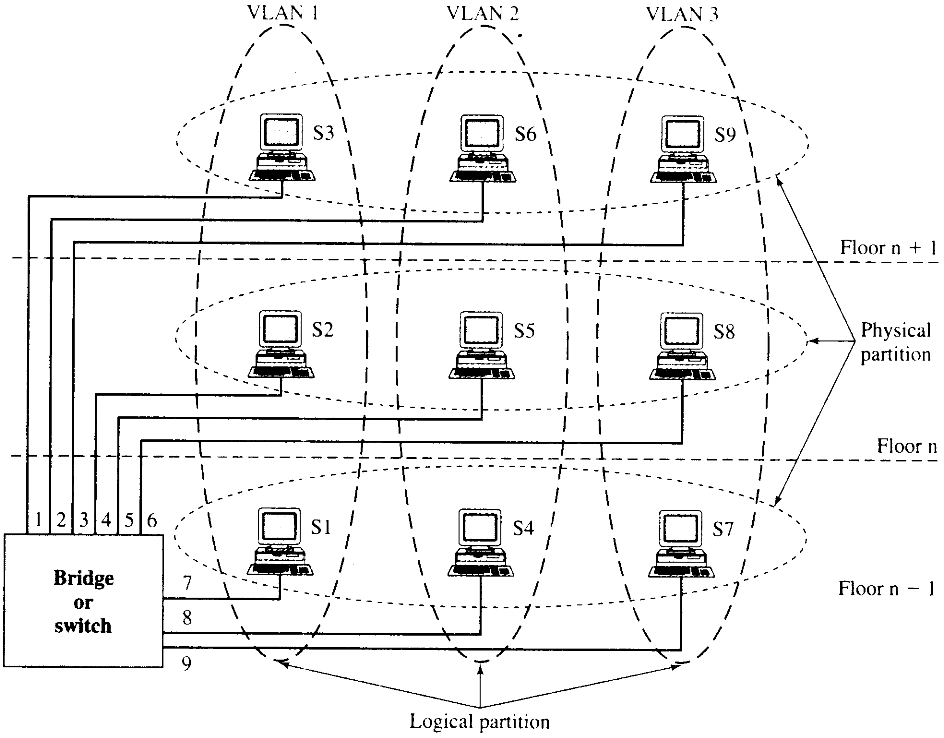


FIGURE 6.92 Physical and logical partitions.

The notion of a **virtual LAN (VLAN)** has been developed to address the preceding problem by allowing logical partitioning of stations into communities of interest (called *VLAN groups* or simply VLANs) that are not constrained by the physical location or connectivity. Figure 6.92 shows an example where three VLANs are created and stations at different physical locations can be associated to the same VLAN. When the association of a station needs to be changed to a different VLAN, the reassociation can be accomplished by simply configuring bridges or switches rather than rewiring the station's physical connectivity.

Recall that with an ordinary (i.e., *VLAN-unaware*) bridge, when station 1 sends a broadcast frame, every station in the broadcast domain (all stations in the figure) will receive the frame. With *VLAN-aware bridges*, the broadcast frame from station 1 will be forwarded only to stations that are associated with VLAN 1, which are stations 2 and 3. How does a VLAN-aware bridge provide the appropriate isolation among VLANs?

A simple and common approach is to implement **port-based VLANs** where each bridge port is individually associated to a particular VLAN. For example, in Figure 6.92, the network administrator would first have to configure the bridge so that ports 1, 4, and 7 are associated to VLAN 1, ports 2, 5, and 8 are associated to VLAN 2, and ports 3, 6, and 9 are associated to VLAN 3. After configuration, the bridge only forwards frames to the outgoing ports that are associated to the same VLAN as the incoming port the frames arrive on. If a station would like to send a frame to different VLAN, that station should first direct the frame to a router, which will perform forwarding at the network layer. One limitation of port-based VLAN is that all frames arriving on the same port must share the same VLAN.

A more flexible approach is to implement **tagged VLANs** where each frame is explicitly tagged to indicate which VLAN the frame belongs to. Explicit tagging is implemented by inserting a *VLAN tag* right after the source MAC address field in a frame. The VLAN tag consists of a VLAN protocol ID and a tag control information that contains the VLAN ID. A VLAN-aware bridge forwards tagged frames to the outgoing ports that are associated to the same VLAN as the VLAN ID found in the tagged frames. A VLAN ID can be associated to a port statically through configuration or dynamically through bridge learning.

SUMMARY

In the first part of this chapter we considered protocols for broadcast networks. These networks are characterized by the sharing of a transmission medium by many users or stations. They typically arise in situations where a low-cost communications system is required to interconnect a relatively small number of users. They also occur in the access portion of backbone networks.

We introduced static and dynamic approaches to channel sharing. The static approaches lead to channelization techniques, and the dynamic approaches lead to medium access control. We introduced the class of random access MAC protocols and explained how ALOHA and slotted ALOHA can provide low-delay frame transfer but with a limited maximum throughput. We also discussed carrier sensing and collision detection as techniques for improving delay throughput performance. We then considered reservation systems that can provide significant improvements in performance. We also introduced polling approaches to medium access control, and we investigated how latency affects system performance. Finally, we discussed ring networks and compared the performance of different approaches to handling token reinsertion.

We introduced the FDMA, TDMA, and CDMA approaches to creating channels in shared medium networks. We explained why channelization approaches are suitable for situations in which user traffic is constant, but not appropriate when it is bursty. We also explained how CDMA differs from conventional channelization approaches. We also introduced the AMPS, IS-54/136, GSM, and IS-95 cellular telephone standards as examples of how the channelization techniques are applied in practice.

We explored the frame transfer delay performance of medium access control and channelization approaches and provided quantitative results on the effect of delay-bandwidth product on the system performance.

In the second part of this chapter we introduced the IEEE 802 layered architecture that places LANs in the data link layer and defines a common set of services that the MAC sublayer can provide to the logical link layer, which in turn supports the network layer.

We discussed the IEEE 802.3 and the Ethernet LAN standards and their variations, noting in particular the trend toward switched implementations. The IEEE 802.5

token-ring and the FDDI ring standards were introduced, and the various possible ways of handling the token were discussed. We also explained the IEEE 802.11 wireless LAN standard. We introduced system design constraints that are particular to wireless networks. We saw how the requirement that movable and mobile users be accommodated within an extended LAN and the requirements to provide time-bounded and asynchronous services led to a standard that is broader than previous wired LAN standards.

LANs are limited in the number of stations that they can handle, so bridges are required to build extended LANs. We introduced the transparent bridge and source routing bridge approaches to building these extended LANs. Finally, we showed how VLANs could be used to partition LANs arbitrarily in a logical manner independent of the physical locations of stations.

CHECKLIST OF IMPORTANT TERMS

algorithm
1-Persistent CSMA
access point (AP)
ad hoc network
◆ Advanced Mobile Phone System (AMPS)
backoff
basic service set (BSS)
bridge
bridged LAN
broadcast address
broadcast network
carrier sensing multiple access (CSMA)
carrier sensing multiple access with collision avoidance (CSMA-CA)
carrier sensing multiple access with collision detection (CSMA-CD)
channelization scheme
clear-to-send frame (CTS)
◆ code division multiple access (CDMA)
collision domain
contention period (CP)
contention-free period (CFP)
cycle time
DCF interframe space (DIFS)
distributed coordination function (DCF)
distribution system
efficiency
Ethernet
Ethernet switch
extended LAN
extended service set (ESS)
Fiber Distributed Data Interface (FDDI)
frame transfer delay
frequency-division duplex (FDD)
◆ frequency-division multiple access (FDMA)
full duplex
gateway
Gigabit Ethernet
◆ Global System for Mobile Communications (GSM)
half duplex
hub
interframe space
◆ Interim Standard 54/136 (IS-54, IS-136)
◆ Interim Standard 95 (IS-95)
infrastructure network
LAN adapter card
load
logical link control (LLC)
medium access control (MAC)
minislot time
mixed-media bridge

multicast address
multiple access network
network allocation vector (NAV)
network interface card (NIC)
Non-Persistent CSMA
normalized delay-bandwidth product
◆ orthogonal sequences
PCF interframe space (PIFS)
◆ physical layer convergence procedure (PCLP)
◆ physical medium dependent (PMD)
point coordination function (PCF)
polling systems
p-Persistent CSMA
portal
port-based VLAN
repeater
request-to-send (RTS)
ring latency
router
short IFS (SIFS)
◆ soft handoff
source routing bridge
spectrum efficiency
◆ spreading factor G
Subnetwork Access Protocol (SNAP)
switched network
tagged VLAN
target token rotation time (TTRT)
time-division duplex (TDD)
◆ time-division multiple access (TDMA)
throughput
token
token bit
token holding time (THT)
token rotation timer (TRT)
total walk time
transparent bridge
unicast address
virtual LAN (VLAN)
vulnerable period
walk time
Walsh-Hadamard matrix

FURTHER READING

Backes, F., "Transparent Bridges for Interconnection of IEEE 802 LANs," *IEEE Network*, Vol. 2, No. 1, January 1988, pp. 5–9.

Bertsekas, D. and R. Gallager, *Data Networks*, Prentice-Hall, Englewood Cliffs, New Jersey, 1992.

Crow, B. P., I. Widjaja, J. G. Kim, and P. T. Sakai, "IEEE 802.11 Wireless Local Area Networks," *IEEE Communications Magazine*, September 1997, pp. 116–126.

Dixon, R. C. and Pitt, D. A., "Addressing, Routing, and Source Routing," *IEEE Network*, Vol. 2, No. 1, January 1988, pp. 25–32.

Garg, V. K. and J. E. Wilkes, *Wireless and Personal Communications Systems*, Prentice-Hall PTR, Upper Saddle River, New Jersey, 1996.

Gibson, J. D., ed., *The Mobile Communications Handbook*, CRC Press, Boca Raton, Florida, 1999.

Goodman, D. J., *Wireless Personal Communications Systems*, Addison-Wesley, Reading, Massachusetts, 1997.

IEEE, *IEEE 802.3 Standard—Local and Metropolitan Networks*, 2002.

IEEE, *IEEE 802.11 Standard—Wireless LAN*, 1999.

Perlman, R., *Interconnections: Bridges, Routers, Switches, and Internet Protocols*, Addison-Wesley, Reading, Massachusetts, 2000 (the most comprehensive book on bridges and routers).

Rappaport, T. S., *Wireless Communications: Principles and Practice*, Prentice-Hall PTR, Upper Saddle River, New Jersey, 1996.

Schwartz, M., *Telecommunication Networks: Protocols, Modeling, and Analysis*, Addison-Wesley, Reading, Massachusetts, 1987.

Stüber, G. L., *Principles of Mobile Communication*, Kluwer Academic Publishers, Boston, 1996.

Viterbi, A. J., *CDMA: Principles of Spread Spectrum Communication*, Addison-Wesley, Reading, Massachusetts, 1995.

See our website for additional references available through the Internet.

PROBLEMS

6.1. Why do LANs tend to use broadcast networks? Why not use networks consisting of multiplexers and switches?

6.2. Explain the typical characteristics of a LAN in terms of network type, bit rate, geographic extent, delay-bandwidth product, addressing, and cost. For each characteristic, can you find a LAN that deviates from the typical? Which of the above characteristics is most basic to a LAN?

6.3. Compare the two-channel approach (Figure 6.4) with the single-channel approach (Figure 6.5) in terms of the types of MAC protocols they can support.

6.4. Suppose that the ALOHA protocol is used to share a 56 kbps satellite channel. Suppose that frames are 1000 bits long. Find the maximum throughput of the system in frames/second.

6.5. Let G be the total rate at which frames are transmitted in a slotted ALOHA system. What proportion of slots go empty in this system? What proportion of slots go empty when the system is operating at its maximum throughput? Can observations about channel activity be used to determine when stations should transmit?

6.6. Modify the state transition diagram of Stop-and-Wait ARQ to handle the behavior of a station that implements the ALOHA protocol.

6.7. Suppose that each station in an ALOHA system transmits its frames using spread spectrum transmission. Assume that the spreading sequences for the different stations have been selected so that they have low cross-correlations. What happens when transmissions occur at the same time? What limits the capacity of this system?

6.8. Consider four stations that are attached to two different bus cables. The stations exchange fixed-size frames of length 1 sec. Time is divided into slots of 1 sec. When a station has a frame to transmit, the station chooses either bus with equal probability and transmits at the beginning of the next slot with probability p. Find the value of p that maximizes the rate at which frames are successfully transmitted.

6.9. In a LAN, which MAC protocol has a higher efficiency: ALOHA or CSMA-CD? What about in a WAN? Explain.

6.10. A channel using random access protocols has three stations on a bus with end-to-end propagation delay τ. Station A is located at one end of the bus, and stations B and C are together located at the other end of the bus. Frames arrive at the three stations and are ready to be transmitted at stations A, B, and C at the respective times $t_A = 0$, $t_B = \tau/2$, and $t_C = 3\tau/2$. Frames require transmission times of 4τ. In appropriate figures, with time as the horizontal axis, show the transmission activity of each of the three stations for.

(a) ALOHA

(b) Non-Persistent CSMA

(c) Non-Persistent CSMA-CD

6.11. Estimate the maximum throughput of the CDPD system assuming a packet length of 1096 bytes. Hint: What is a for this system?

6.12. Can the Digital Sense Multiple Access protocol, which is used by CDPD, also be used on the digitial carrier of GSM? If yes, explain how.

6.13. M terminals are attached by a dedicated pair of lines to a hub in a star topology. The distance from each terminal to the hub is d meters, the speed of the transmission lines is R bits/second, all frames are of length 12,500 bytes, and the signal propagates on the line at a speed of 2.5 (10^8) meters/second. For the four combinations of the following parameters ($d = 25$ meters or $d = 2500$ meters; $R = 10$ Mbps or $R = 10$ Gbps), compare the maximum network throughput achievable when the hub is implementing slotted ALOHA and CSMA-CD.

6.14. Consider the star-topology network Problem 6.13 when the token-ring protocol is used for medium access control. Assume single-frame operation, eight-bit latency at each station, $M = 125$ stations. Assume a free token is three bytes long.

(a) Find the effective frame transmission time for the four combinations of d and R.

(b) Assume that each station can transmit up to a maximum of k frames/token. Find the maximum network throughput for the four cases of d and R.

6.15. A wireless LAN uses polling to provide communications between M workstations and a central base station. The system uses a channel operating at 25 Mbps. Assume that all stations are 100 meters from the base station and that polling messages are 64 bytes long. Assume that frames are of constant length of 1250 bytes. Assume that stations indicate that they have no frames to transmit with a 64-byte message.

(a) What is the maximum possible arrival rate that can be supported if stations are allowed to transmit an unlimited number of frames/poll?

(b) What is the maximum possible arrival rate that can be supported if stations are allowed to transmit N frames/poll?

(c) Repeat parts (a) and (b) if the transmission speed is 2.5 Gbps.

6.16. A token-ring LAN interconnects M stations using a star topology in the following way. All the input and output lines of the token-ring station interfaces are connected to a cabinet where the actual ring is placed. Suppose that the distance from each station to the cabinet is 100 meters and that the ring latency per station is eight bits. Assume that frames are 1250 bytes and that the ring speed is 25 Mbps.

(a) What is the maximum possible arrival rate that can be supported if stations are allowed to transmit an unlimited number of frames/token?

(b) What is the maximum possible arrival rate that can be supported if stations are allowed to transmit 1 frame/token using single-frame operation? using multitoken operation?

(c) Repeat parts (a) and (b) if the transmission speed is 2.5 Gbps.

6.17. Suppose that a LAN is to carry vcice and packet data traffic. Discuss what provisions if any are required to handle the voice traffic in the reservation, polling, token ring, ALOHA, and CSMA-CD environments. What changes if any are required for the packet data traffic?

6.18. A wireless LAN has mobile stations communicating with a base station. Suppose that the channel available has W Hz of bandwidth and suppose that the inbound traffic from the mobile stations to the base is K times smaller than the outbound traffic from the base to the workstations. Two methods are considered for dealing with the inbound/outbound communications. In frequency-division duplexing the channel is divided into two frequency bands, one for inbound and one for outbound communications. In time-division duplexing all transmissions use the full channel but the transmissions are time-division multiplexed for inbound and outbound traffic.

(a) Compare the advantages and disadvantages of the two methods in terms of flexibility, efficiency, complexity, and performance.

(b) How is the ratio K taken into account in the two methods?

6.19. Consider the following variation of FDMA. Each station is allotted two frequency bands: a band on which to transmit and a band in which to receive reservations from other stations directing it to listen to a transmission from a certain station (frequency band) at a certain time. To receive a frame, a station tunes in the appropriate channel at the appropriate time. To make a reservation, a station transmits at the receiving station's reservation channel. Explain how transmitting and receiving stations can use the reservation channels to schedule frame transmissions.

6.20. Compare FDMA, TDMA, and CDMA in terms of their ability to handle groups of stations that produce information flows that are produced at constant but different bit rates.

6.21. Calculate the autocorrelation function of the pseudorandom sequence in Figure 6.30 as follows. Replace each 0 by -1 and each 1 by $+1$. Take the output sequence of the generator and shift it with respect to itself; take the product of seven (one period) symbol pairs and add. Repeat this calculation for shift values of $0, 1, \ldots, 7$. In what sense does the result approximate the autocorrelation of a random sequence?

6.22. Construct the Walsh orthogonal spreading sequences of length 16.

6.23. Decode the sum signal in Figure 6.33 using the Walsh sequence for channel 4. What do you get? Explain why.

6.24. Compare IS-54 and GSM in terms of their handling of speech and the effect on spectrum efficiency.

6.25. Suppose that the A provider in the 800 MHz cellular band uses GSM and the B provider uses IS-95. Explain how a call from a mobile user in system B to a user in system A is accomplished.

6.26. Suppose that a 1 MHz channel can support a 1 Mbps transmission rate. The channel is to be shared by 10 stations. Each station receives frames with exponential interarrivals and rate $\lambda = 50$ frames/second, and frames are constant length $L = 1000$ bits. Compare the total frame delay of a system that uses FDMA to a system that uses TDMA.

6.27. Discuss how the delay and throughput performance of GPRS vary with the allocation in the number of access request channels, access grant channels, and data channels.

6.28. Consider an "open concept" office where 64 carrels are organized in an 8×8 square array of 3 m × 3 m space per carrel with a 2 m alley between office rows. Suppose that a conduit runs in the floor below each alley and provides the wiring for a LAN to each carrel.
(a) Estimate the distance from each carrel to a wiring closet situated at the side of the square office so that distances are minimized.
(b) Does it matter whether the LAN is token ring or Ethernet? Explain.
(c) Discuss the merits of using a wireless LAN in this setting.

6.29. Suppose that a LAN is to provide each worker in Problem 6.28 with the following capabilities: digital telephone service; H.261 video conferencing; 250 ms retrieval time for a 1 Mbyte file from servers in the wiring closet; 10 e-mails/hour sent and received by each worker (90 percent of e-mails are short, and 10 percent contain a 100-kilobyte attachment).
(a) Estimate the bit rate requirements of the LAN.
(b) Is it worthwhile to assign the users to several LANs and to interconnect these LANs with a bridge?

6.30. Consider a LAN that connects 64 homes arranged in rows of 8 homes on 20 m × 30 m lots on either side of a 10-meter-wide street. Suppose that an underground conduit on either side of the street connects the homes to a pedestal at the side of this rectangular array.
(a) Estimate the distance from each house to the pedestal.
(b) Estimate the bit rate requirements of the LAN. Assume two telephone lines, three MPEG2 televisions, and intense peak-hour Web browsing, say two Web page retrievals/minute at an average of 20 kilobytes/page.
(c) Can a single LAN meet the service requirements of the 64 homes? Explain.

6.31. Use HDLC and Ethernet to identify three similarities and three differences between medium access control and data link control protocols. Is HDLC operating as a LAN when it is used in normal response mode and multipoint configuration?

6.32. An application requires the transfer of network layer packets between clients and servers in the same LAN. Explain how reliable connection-oriented service can be provided over an Ethernet LAN. Sketch a diagram that shows the relationship between the PDUs at the various layers that are involved in the transfer.

6.33. Calculate the difference in header overhead between a DIX Ethernet frame and an IEEE 802.3 frame with SNAP encapsulation.

6.34. Suppose that a group of 10 stations is serviced by an Ethernet LAN. How much bandwidth is available to each station if (a) the 10 stations are connected to a 10 Mbps Ethernet hub; (b) the 10 stations are connected to a 100 Mbps Ethernet hub; (c) the 10 stations are connected to a 10 Mbps Ethernet switch.

6.35. Suppose that an Ethernet LAN is used to meet the requirements of the office in Problem 6.28.
(a) Can the requirements of one row of carrels be met by a 10 Mbps Ethernet hub? by a 10 Mbps Ethernet switch?

(b) Can the requirements of the office be met by a hierarchical arrangement of Ethernet switches as shown in Figure 6.57?

6.36. Suppose that 80 percent of the traffic generated in a LAN is for stations in the LAN and 20 percent is for stations outside the LAN. Is an Ethernet hub preferable to an Ethernet switch? Does the answer change if the percentages are reversed?

6.37. Calculate the parameter *a* and the maximum throughput for a Gigabit Ethernet switch with stations at a 100-meter distance and average frame size of 512 bytes; 1500 bytes; and 64,000 bytes.

6.38. Provide a brief answer and explanation for each of the following questions:
(a) Under a light load, which LAN has a smaller delay: Ethernet or token ring?
(b) Under a high load, which LAN has a smaller delay: Ethernet or token ring?

6.39. Suppose that a token-ring LAN is used to meet the requirements of the office in Problem 6.28.
(a) Calculate the ring latency if all carrels are to be connected in a single ring as shown in Figure 6.58. Repeat for a ring for a single row of carrels.
(b) Can the requirements of one row of carrels be met by a 16 Mbps token ring?
(c) Can the requirements of the office be met by a FDDI ring?

6.40. Suppose that a group of 32 stations is serviced by a token-ring LAN. For the following cases calculate the time it takes to transfer a frame using the three token reinsertion strategies: after completion of transmission, after return of token and after return of frame.
(a) 1000-bit frame; 10 Mbps speed; 2.5-bit latency/adapter; 50 meters between stations.
(b) Same as (a) except 100 Mbps speed and 8-bit latency/adapter.
(c) Same as (a) except 1 km distance between stations.

6.41. Suppose that an FDDI LAN is used to meet the packet voice requirements of a set of users. Assume voice information uses 64 kbps coding and that each voice packet contains 20 ms worth of speech.
(a) Assume that each station handles a single voice call and that stations are 100 meters apart. Suppose that the FDDI ring is required to transfer each voice packet within 10 ms. How many stations can the FDDI accommodate while meeting the transfer requirement?
(b) How many simultaneous calls can be handled if each station is allowed to handle up to 8 calls?

6.42. Use IEEE 802.3 and IEEE 802.11 to discuss three differences between wired and wireless LANs.

6.43. For data packet radio networks, discuss the advantages and disadvantages of providing reliability by (a) implementing error correction at the physical layer, (b) implementing error control as part of the MAC layer, and (c) implementing error control at the LLC layer.

6.44. Consider the distributed coordination function in IEEE 802.11. Suppose that all frame transmissions are preceded by an RTS-CTS handshake. Find the capacity of this protocol following the analysis used for CSMA-CD.

6.45. Suppose one station sends a frame to another station in an IEEE 802.11 ad hoc network. Sketch the data frame and the return ACK frame that are exchanged, showing the contents in the relevant fields in the headers.

6.46. Suppose one station sends a frame to another station in a different BSS in an IEEE 802.11 infrastructure network. Sketch the various data frames and ACK frames that are exchanged, showing the contents in the relevant fields in the headers.

6.47. Why is error control (ARQ and retransmission) included in the MAC layer in IEEE 802.11 and not in IEEE 802.3?

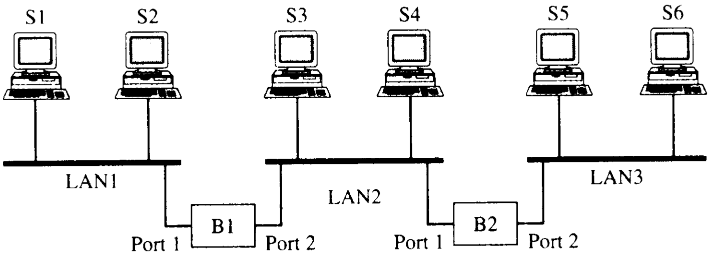
6.48. Consider the exchange of CSMA-CA frames shown in Figure 6.70. Assume the IEEE 802.11 LAN operates at 2 Mbps using a frequency-hopping physical layer. Sketch a time diagram showing the frames transmitted including the final ACK frame. Show the appropriate interframe spacings and NAV values. Use Table 6.6 to obtain the appropriate time parameters. Assume that the data frame is 2000 bytes long.

6.49. Suppose that four stations in an IEEE 802.11 infrastructure network are in a polling list. The stations transmit 20 ms voice frames produced by 64 kbps speech encoders. Suppose that the contention-free period is set to 20 ms. Sketch a point-coordination frame transfer with the appropriate values for interframe spacings, NAV, and data and ACK frames.

6.50. Can a LAN bridge be used to provide the distribution service in an IEEE 802.11 extended service set? If so, explain how the service is provided and give an example of how the frames are transferred between BSSs.

6.51. Can a router be used to provide the distribution service in an IEEE 802.11 extended service set? If so, explain how addressing is handled and give an example of how the frames are transferred between BSSs.

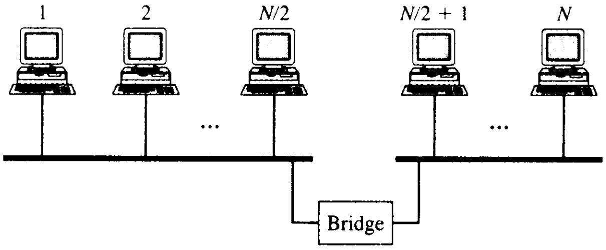
6.52. Six stations (S1-S6) are connected to an extended LAN through transparent bridges (B1 and B2), as shown in the figure. Initially, the forwarding tables are empty. Suppose the following stations transmit frames: S2 transmits to S1, S5 transmits to S4, S3 transmits to S5, S1 transmits to S2, and S6 transmits to S5. Fill in the forwarding tables with appropriate entries after the frames have been completely transmitted.



Station	Port

Station	Port

6.53. Suppose N stations are connected to an extended Ethernet LAN, as shown in the figure, operating at the rate of 10 Mbps. Assume that the efficiency of each Ethernet is 80 percent. Also assume that each station transmits frames at the average rate of R bps, and each frame is equally likely to be destined to any station (including itself). What is the maximum number of stations N that can be supported if R is equal to 100 Kbps? If the bridge is replaced with a repeater, what is the maximum number of stations that can be supported? (Assume that the efficiency of the entire Ethernet is still 80 percent.)



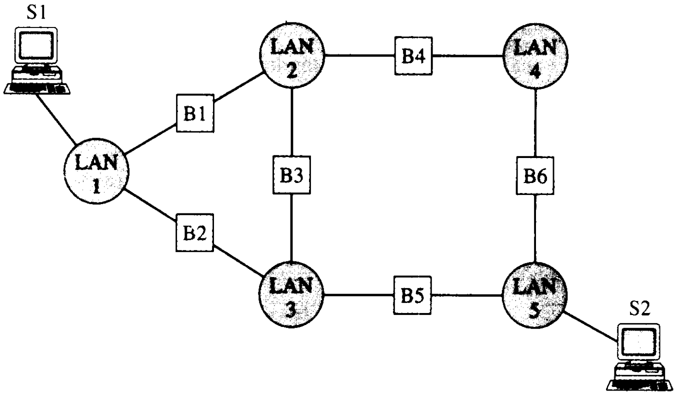
6.54. Five LANs, as shown in the figure, are interconnected by using source routing bridges. Assume that bridges 3 and 4 are not part of the initial spanning tree.

(a) Show the paths of the single route broadcast frames when S1 wants to learn the route to S2.

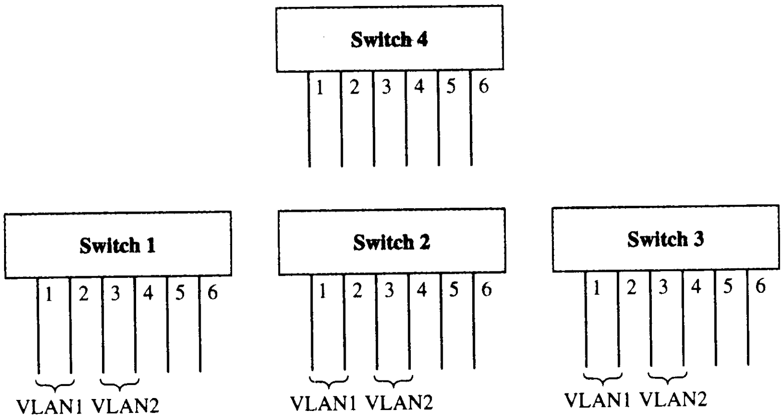
(b) Show the paths of all routes broadcast frames returned by S2.

(c) List all possible routes from S1 to S2 from part (b).

(d) How many LAN frames are required to learn the possible routes?



6.55. Ports 1 through 4 in switches 1, 2, and 3, as shown in the figure, provide port-based VLAN connectivity to stations attached to these ports. A network administrator needs to interconnect the other ports and configure the switches so that stations of the same VLAN connected to different switches can communicate. Use the association notation: switch a, VLAN n, port x, port y, ..., port z, to indicate that ports x, y, ..., z are associated to VLAN n in switch a, and the connectivity notation: (switch a, port x) to (switch b, port y) to indicate that port x in switch a is connected to port y in switch b. Show the required association and connectivity for all the switches in the figure.



CHAPTER 7

Packet-Switching Networks

Traditional telephone networks operate on the basis of circuit switching. A call setup process reserves resources (time slots) along a path so that the stream of voice samples can be transmitted with very low delay across the network. The resources allocated to a call cannot be used by other users for the duration of the call. This approach is inefficient when the amount of information transferred is small or if information is produced intermittently in bursts, as is the case in many computer applications. In this chapter we examine **packet-switching networks**, which transfer blocks of information called **packets**. With appropriate mechanisms, packet-switching networks can be designed to support computer applications and real-time applications such as telephony.

We can view packet networks from two perspectives. One perspective involves an *external view* of the network and is concerned with the services that the network provides to the transport layer that operates above it at the end systems. Here we are concerned with whether the network service requires the setting up of a connection and whether the transfer of user data is provided with quality-of-service guarantees. Ideally the definition of the network services is independent of the underlying network and transmission technologies. This approach allows the transport layer and the applications that operate above it to be designed so that they can function over any network that provides the given services.

A second perspective on packet networks is concerned with the *internal* operation of a network. Here we look at the physical topology of a network, the interconnection of links, switches, and routers. We are concerned with the approach that is used to direct information across the network: datagrams, or virtual circuits. We are also concerned with addressing and routing procedures, as well as with dealing with congestion inside the network. We must also manage traffic so that the network can deliver information with the quality of service it has committed to.

It is useful to compare these two perspectives in the case of broadcast networks and LANs from Chapter 6 and the packet-switched networks considered here. The first perspective, involving the services provided to the layer above, does not differ in a fundamental way between broadcast and switched packet networks. The second perspective, however, is substantially different. In the case of LANs, the network is small, addressing is simple, and the frame is usually transferred in one hop so no routing is required. In the case of packet-switching networks, addressing must accommodate extremely large-scale networks and must work in concert with appropriate routing algorithms. These two challenges, addressing and routing, are the essence of the network layer.

In this chapter we deal with the general issues regarding packet-switching networks. Later chapters deal with specific architectures, namely, Internet Protocol (IP) packet networks and asynchronous transfer mode (ATM) packet networks. The chapter is organized as follows:

1. *Network services and internal network operation.* We elaborate on the two perspectives on networks, and we discuss the functions of the network layer, including internetworking.
2. *Physical view of networks.* We examine typical configurations of packet-switching networks. This section defines the role of multiplexers, LANs, switches, and routers in network and internetwork operation.
3. *Datagrams and virtual circuits.* We introduce the two basic approaches to operating a packet-switching network, and we use IP and ATM as examples of these approaches.
4. *Routing.* We introduce the basic approaches for selecting routes across the network, and examine how routing tables in a network steer packets from the source to the destination.
5. *Shortest path algorithms.* We continue our discussion of routing, focusing on two shortest-path routing algorithms: the Bellman-Ford algorithm and Dijskstra's algorithm.
6. *ATM networks.* We introduce ATM networks as an example of an advanced virtual-circuit packet-switching network that can support multiple types of services.
7. *Traffic management at the packet level.* We discuss packet-level traffic management operating in a short time scale. This type of traffic management is mainly concerned with packet queueing and packet scheduling to provide differentiated treatment for packets belonging to different QoS classes.
8. *Traffic management at the flow level.* We continue with traffic management at the flow level operating in a medium time scale. Common approaches include admission control and congestion control.
9. *Traffic management at the flow-aggregate level.* We conclude with an introduction to traffic management at the flow-aggregate level operating in a long time scale. The main objective is to map aggregated flows of traffic onto the network so that resources are efficiently utilized.

The material on ATM and traffic management is relatively advanced. The corresponding sections (7.6 to 7.9) can be skipped and the reader may proceed to Chapter 8, depending on their background or interest.

7.1 NETWORK SERVICES AND INTERNAL NETWORK OPERATION

The essential function of a network is to transfer information among the users that are attached to the network or internetwork. In Figure 7.1 we show that this transfer may involve a single block of information or a sequence of blocks that are temporally related. In the case of a single block of information, we are interested in having the block delivered correctly to the destination, and we may also be interested in the delay experienced in traversing the network. In the case of a sequence of blocks, we may be interested not only in receiving the blocks correctly and in the right sequence but possibly also in delivering a relatively unimpaired temporal relation.

Figure 7.2 shows a transport protocol that operates end to end across a network. The transport layer peer processes at the end systems accept messages from their higher layer and transfer these messages by exchanging segments end to end across the network. The figure shows the interface at which the network service is visible to the transport layer. The network service is all that matters to the transport layer, and the manner in which the network operates to provide the service is irrelevant.

The network service can be **connection-oriented** or **connectionless**. A *connectionless service* is very simple, with only two basic interactions between the transport layer (user of the service) and the network layer (provider of the service): a request to the network layer that it send a packet and an indication from the network layer that a packet has arrived. The user can request transmission of a packet at any time, and *does not need to inform the network layer* that the user intends to transmit information ahead of time. A connectionless service puts total responsibility for error control, sequencing, and flow control on the end-system transport layer.

The network service can be *connection-oriented.* In this case the transport layer cannot request transmission of information until a connection between the end systems has been set up. The essential points here are that *the network layer must be informed* about the new flow that is about to be sent to the network and that the network layer maintains state information about the flows it is handling. During connection setup, parameters related to usage and quality of service may be negotiated and network resources may be allocated to ensure that the user flow can be handled as required. A connection-release procedure may also be required to terminate the connection. It is clear that providing connection-oriented service entails greater complexity than connectionless service in the network layer.

It is also possible for a network layer to provide a choice of services to the user of the network. For example, the network layer could offer: (1) best-effort connectionless

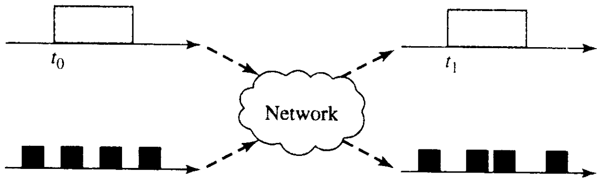


FIGURE 7.1 A network transfers information among users.

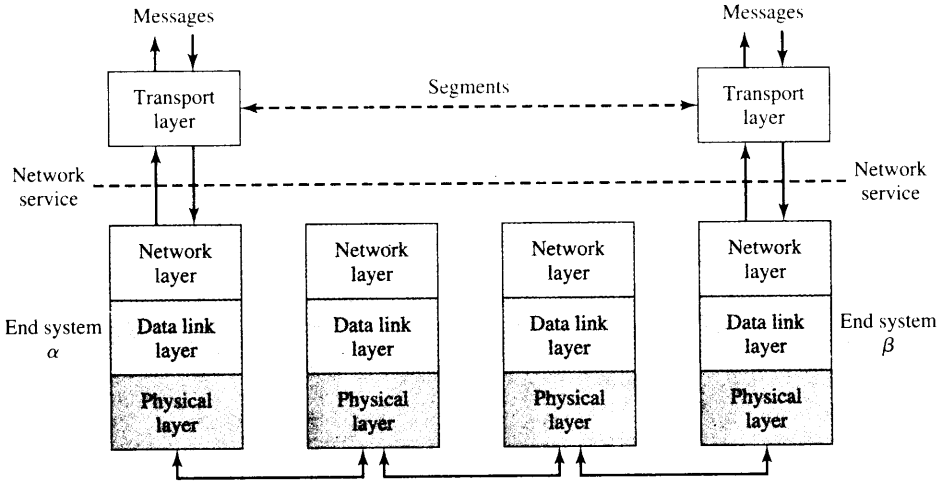


FIGURE 7.2 Peer-to-peer protocols operating end to end across a network—protocol stack view.

service; (2) low-delay connectionless service; (3) connection-oriented reliable stream service; and (4) connection-oriented transfer of packets with delay and bandwidth guarantees. It is easy to come up with examples of applications that can make use of each of these services. However, it may not follow that all the services should be offered by the network layer. Two interrelated reasons can be given for keeping the set of network services to a minimum: the end-to-end argument and the need for network scalability.

When applied to the issue of choice of network services, the end-to-end argument suggests that functions should be placed as close to the application as possible, since it is the application that is in the best position to determine whether a function is being carried out completely and correctly. This argument suggests that as much functionality as possible should be located in the transport layer or higher and that the network services should provide the minimum functionality required to meet application performance.

Up to this point we have considered only the services offered by the network layer. Let us now consider the internal operation of the network. Figure 7.3 shows the relation between the service offered by the network and the internal operation. We say that the internal operation of a network is *connectionless* if packets are transferred within the network as datagrams. Thus in the figure each packet is routed independently. Consequently packets may follow different paths from α to β and so may arrive out of order. We say that the internal operation of a network is *connection-oriented* if packets follow a virtual circuit along a forward path that has been established from a source to a destination. Thus to provide communications between α and β, routing to set up a virtual circuit is done once, and thereafter packets are simply forwarded along the established virtual circuit. If resources are reserved during connection setup, then bandwidth, delay, and loss guarantees can be provided.

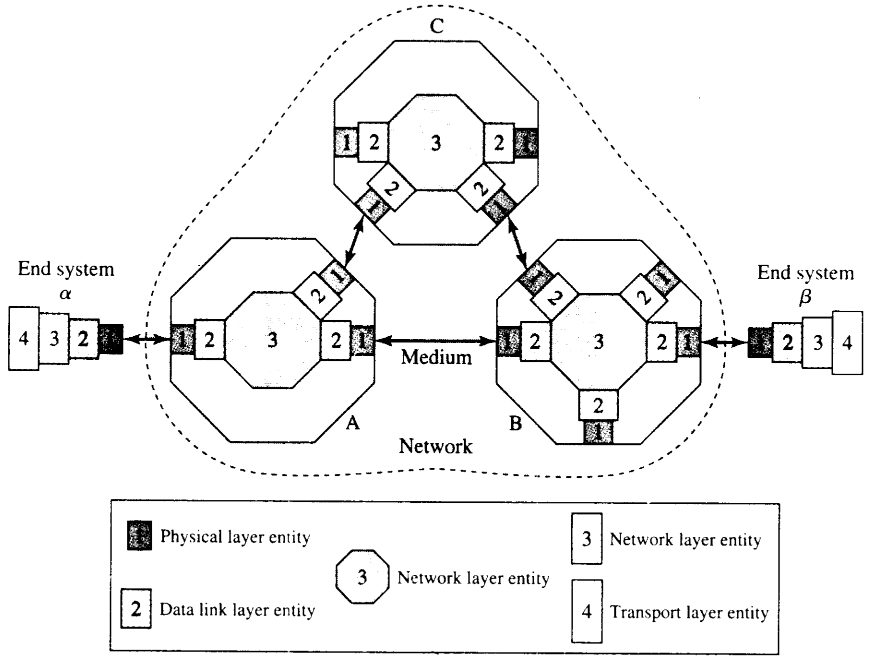


FIGURE 7.3 Layer 3 entities work together to provide network service to layer 4 entities.

THE END-TO-END ARGUMENT FOR SYSTEM DESIGN

The *end-to-end argument* in system design articulated in [Saltzer 1984] states that an end-to-end function is best implemented at a higher level than at a lower level. The reason is that the correct end-to-end implementation requires *all* intermediate low-level components to operate correctly. This feature is difficult and sometimes impossible to ensure and is frequently too costly. The higher-level components at the ends are in a better position to determine that a function has been carried out correctly and in better position to take corrective action if they have not. In certain cases, low-level actions to support the end-to-end function may be justified only as performance enhancements.

We already encountered the end-to-end argument in the comparison of end-to-end error control and hop-by-hop error control in Chapter 5. The argument here is that the end system will have to implement error control on an end-to-end basis regardless of lower-level error-control mechanisms that may be in place because the individual low-level mechanisms cannot cover all sources of errors, for example, errors introduced within a node. Consequently, lower-level mechanisms are not essential and should be introduced only to enhance performance. For example, the transmission of a long file over a sequence of nearly error-free links does not require per link error control. On the other hand, the transmission of such a file over a sequence of error-prone links does argue for per link error control.

The fact that a network offers connection-oriented service, connectionless service, or both does not dictate how the network must operate internally. In discussing TCP and IP, we have already seen that a connectionless packet network (e.g., IP) can support connectionless service (UDP) as well as connection-oriented service (TCP). We will also see that a connection-oriented network (e.g., ATM) can provide connectionless service as well as connection-oriented service. We discuss virtual-circuit and datagram network operation in more detail in a later section. However, it is worthwhile to compare the two at this point at a high level.

The approach suggested by the end-to-end argument keeps the network service (and the network layer that provides the service) as simple as possible while adding complexity at the edge only as required. This strategy fits very well with the need to grow networks to very large scale. We have seen that the value of a network grows with the community of users that can be reached and with the range of applications that can be supported. Keeping the core of the network simple and adding the necessary complexity at the edge enhances the scalability of the network to larger size and scope.

This reasoning suggests a preference for a connectionless packet network, which has lower complexity than a connection-oriented packet network. The reasoning does allow the possibility for some degree of "connection orientation" as a means to ensure that applications can receive the proper level of performance. Indeed current research and standardization efforts (discussed in Chapter 10) can be viewed as an attempt in this direction to determine an appropriate set of network services and an appropriate mode of internal network operation.

We have concentrated on high-level arguments up to this point. What do these arguments imply about the functions that should be in the network layer? Clearly, functions that need to be carried out at every node in the network must be in the network layer. Thus functions that route and forward packets need to be done in the network layer. Priority and scheduling functions that direct how packets are treated in a node to ensure a certain quality of service also need to be in the network layer. Functions that belong in the edge should, if possible, be implemented in the transport layer or higher. A third category of functions can be implemented either at the edge or inside the network. For example, while congestion takes place inside the network, the remedy may involve reducing input flows at the edge of the network. Indeed, congestion control has been implemented in the transport layer and in lower layers.

Another set of functions is concerned with making the network service independent of the underlying transmission systems. For example, different transmission systems (e.g., optical versus wireless) may have different limits on the frame size they can handle. The network layer may therefore be called upon to carry out segmentation inside the network and reassembly at the edge. Alternatively, the network could send error messages to the sending edge, requesting that the packet size be reduced. A more challenging set of functions arises when the "network" itself may actually be an internetwork. In this case the network layer must also be concerned not only about differences in the size of the units that the component networks can transfer but also about differences in addressing and in the services that the component networks provide.

In the remainder of the chapter we deal with the general aspects of internal network operation. In Chapters 8 and 9 we discuss the specific details of IP and ATM networks.

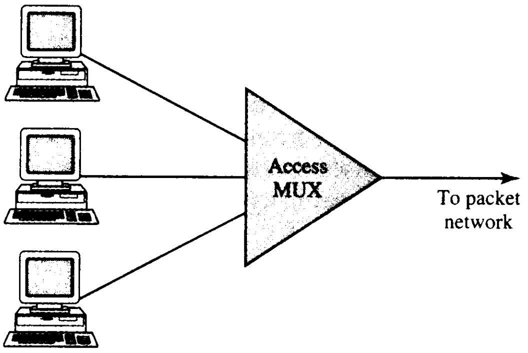


FIGURE 7.4 Access network.

7.2 PACKET NETWORK TOPOLOGY

This section considers existing packet-switching networks. We present an end-to-end view of existing networks from a personal computer, workstation, or server through LANs and the Internet and back.

First let us consider the way in which users may access packet networks such as the Internet. Figure 7.4 shows an *access network* with a point-to-point topology where computers located in subscriber homes are connected to an access multiplexer located in the service provider network. An example of an access multiplexer includes a Digital Subscriber Loop Access Multiplexer (DSLAM) located in a telephone central office. The computer in the subscriber home is connected to the DSLAM in the central office via an ADSL modem described in Chapter 3. In another scenario, multiple users may share the same transmission line to the access multiplexer, resulting in a point-to-multipoint topology. Such a system arises in a cable TV access network where the access multiplexer is the cable modem termination system described in Chapter 3. In either case, the main purpose of the access multiplexer is to combine the typically bursty traffic flows from the individual computers into aggregated flows so that the transmission line to the packet network (typically the Internet) is used more efficiently. Besides providing connectivity services to the packet network, the service provider can also offer other services such as e-mail and databases.

EXAMPLE Oversubscription

Suppose N subscribers are connected to the access multiplexer and the transmission line between the subscriber and the multiplexer has a capacity c bits/second (bps). The multiplexer in turn is connected to the packet network with a transmission line of capacity $C = nc$ (some integer multiple of c). The service provider is said to be oversubscribing its bandwidth resource to the packet network if the capacity of the transmission line C is less than Nc (the maximum total transmission rate from all N subscribers). Oversubscription takes advantage of the bursty nature of the subscriber's traffic due to the fact that it is unlikely that all subscribers are transmitting at maximum rates simultaneously, and the term *oversubscription ratio* is defined by N/n. Suppose that each subscriber transmits data intermittently at the rate of c bps when there is data to send and zero otherwise, and that the average transmission rate is r bps ($r < c$). Table 7.1 shows various levels of oversubscription for different values of r/c, assuming that the overflow probability (defined in Problem 7.11) is less than 1 percent. The table also shows the advantage of oversubscription as the number of subscribers increases: the degree of oversubscription that can be accommodated at a given level of quality increases with larger population. Oversubscription is widely used in the access portion of packet-switching networks to optimize the use of bandwidth resources.

TABLE 7.1 Oversubscription levels for different values of r/c.

N	r/c	N/n
10	0.01	10
10	0.05	3.3
10	0.1	2.5
20	0.1	3.3
40	0.1	4.4
100	0.1	5.5

Often the subscriber may have multiple computers connected to the same access multiplexer. In this case, another level of multiplexing occurs through a device in the subscriber *home network*. Besides aggregating traffic from multiple computers, this device (known as an application-level gateway) often performs a *network address translation (NAT)* function. The need for network address translation arises when the service provider assigns a single global network address to the subscriber in order to conserve address space. To accommodate multiple computers in the home network, the subscriber assigns a private network address that is only defined within the subscriber home network to each computer. The purpose of the gateway is to translate the private network address of each packet to the global network address when a packet leaves the home network and vice versa when a packet arrives at the home network. The address translation usually uses higher-layer information such as service access point (SAP) identifiers to ensure that the mapping is one-to-one. Note that NAT is an example that violates the end-to-end argument and is considered as a temporary solution by many.

Local area networks (LANs) also provide the access to packet-switching networks in many environments. Figure 7.5 shows a structure of a *campus network* that interconnects multiple LANs in an organization. LANs for a large group of users such as a department are interconnected in an extended LAN through the use of LAN switches, identified by lowercase s in the figure. Resources such as servers and databases that are primarily of use to this department are kept within the subnetwork. This approach reduces delays in accessing the resources and contains the level of traffic that leaves the subnetwork. Each subnetwork has access to the rest of the organization through a router R that accesses the

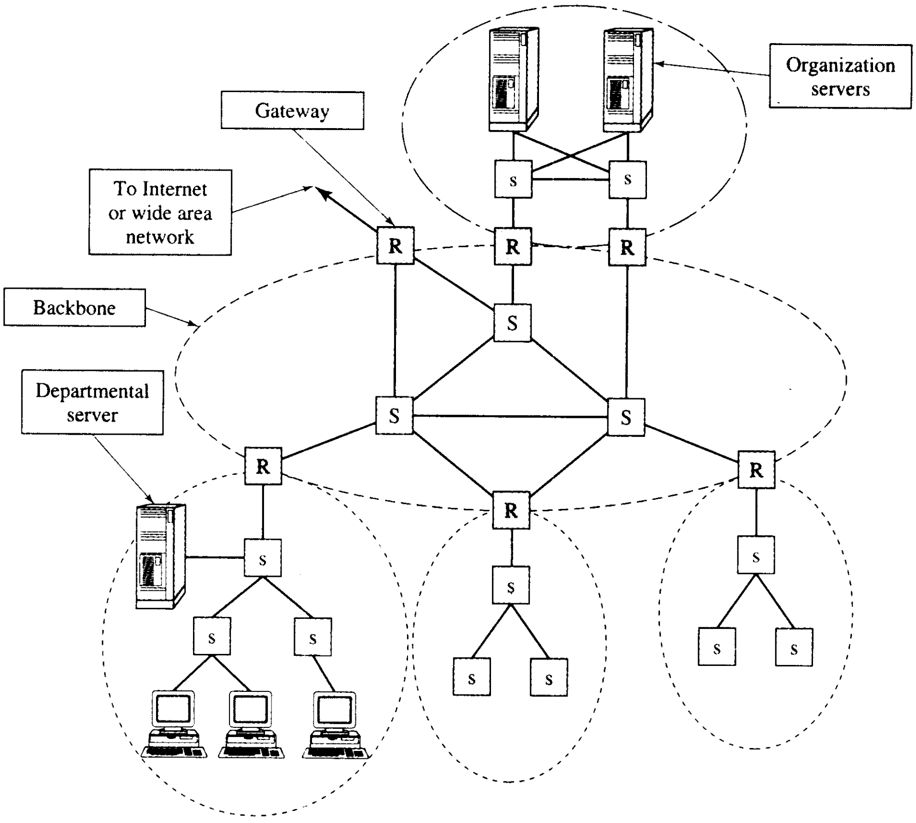


FIGURE 7.5 Campus network.

campus backbone network. A subnetwork also uses the campus backbone to reach the "outside world" such as the Internet or other sites belonging to the organization through a border router. Depending on the type of organization, the border router may implement firewall functions to control the traffic that is allowed into and out of the campus network.

Servers containing critical resources that are required by the entire organization are usually located in a data center where they can be easily maintained and where security can be enforced. As shown in Figure 7.5, the critical servers may be provided with redundant paths to the campus backbone network. These servers are usually placed near the backbone network to minimize the number of hops required to access them from the rest of the organization.

The traffic within an extended LAN is delivered based on the *physical* LAN addresses. However, applications in host computers may operate on the basis of *logical* IP addresses. Therefore, the physical address corresponding to an IP address needs to be determined every time an IP packet is to be transmitted over a LAN. This *address resolution* problem can be solved by using IP address to physical address translation tables. In Chapter 8 we discuss the *Address Resolution Protocol* that IP uses to solve this problem.

The routers in the campus network are interconnected to form the campus backbone network, depicted by the mesh of switches, designated *S*, in Figure 7.5. Typically, for large organizations such as universities these routers are interconnected by very high speed LANs, for example, Gigabit Ethernet or an ATM network. The routers use the Internet Protocol (IP), which enables them to operate over various data link and network technologies. The routers exchange information about the state of their links to dynamically calculate routing tables that direct packets across the campus network. This approach allows the network to adapt to changes in topology due to faults in transmission links or equipment.

The routers in the campus network may forn a *domain* or *autonomous system*. The term *domain* indicates that the routers run the same routing protocol. The term *autonomous* system is used for one or more domains under a single administration. All routing and policy decisions inside the autonomous system are independent of any other network.

Organizations with multiple sites may have their various campus networks interconnected through routers interconnected by leased digital transmission lines or frame relay connections. In this case access to the *wide area network* may use an access multiplexer such as the one shown in Figure 7.4. In addition the campus network may be connected to an *Internet service provider (ISP)* through one or more border routers as shown in Figure 7.6. To communicate with other networks, the autonomous system must provide information about its network routes in the border routers. The border router communicates on an interdomain level, whereas other routers in a campus network operate at the intradomain level.

A national ISP provides points of presence (POPs) in various cities where customers can connect to their network. The ISP has its own national backbone network for interconnecting its POPs. This backbone network could be based on ATM, or it

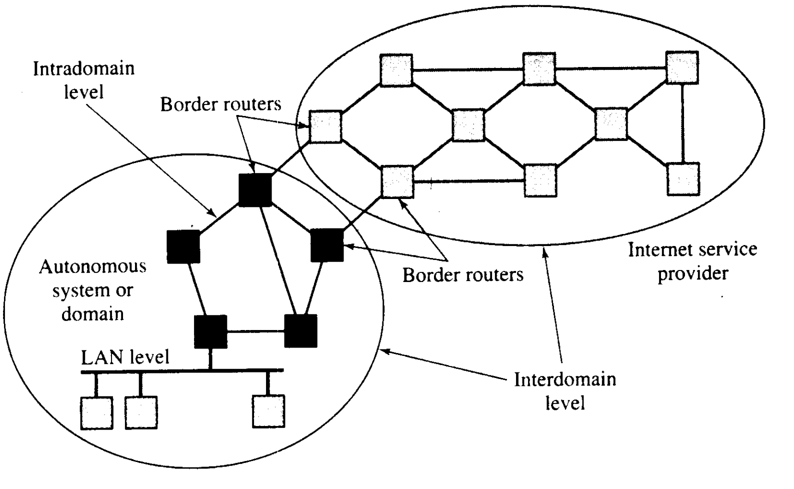


FIGURE 7.6 Intradomain and interdomain levels.

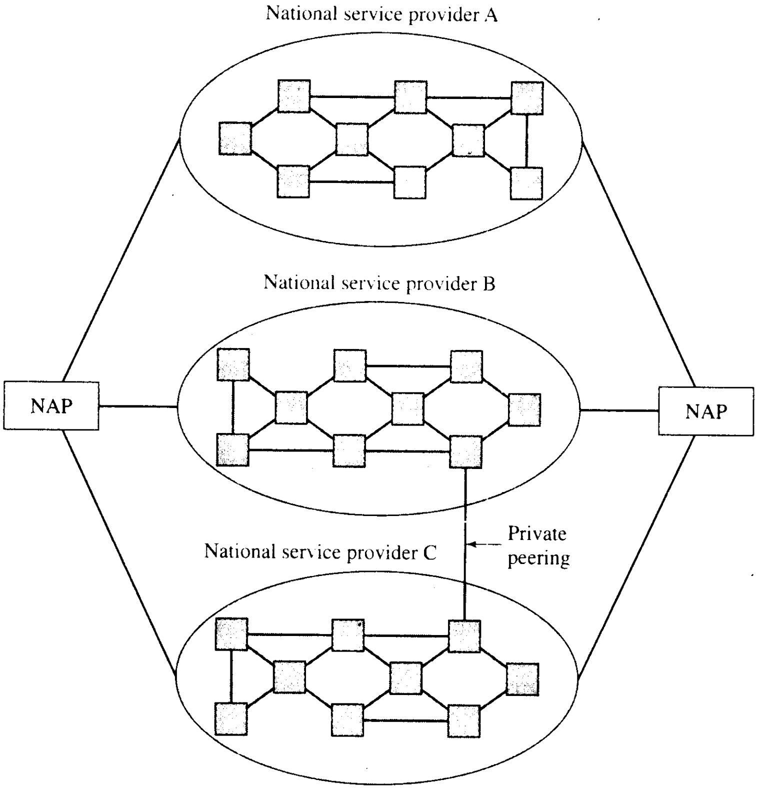

National service provider A
National service provider B
National service provider C
NAP
NAP
Private peering

FIGURE 7.7 National ISPs exchange traffic at NAPs.

might use some newer technology such as MPLS (discussed in Chapter 10). The ISPs in turn exchange traffic at public *peering points* called *network access points (NAPs)*, as shown in Figure 7.7. A NAP is a colocated set of high-speed routers through which the routers from different ISPs can exchange traffic, and as such NAPs are crucial to the interconnectivity provided by the Internet. (Four NAPs were originally set up by the National Science Foundation). The ISPs interconnected to a NAP need to exchange routing information. If there are n such ISPs, then $n(n-1)/2$ pairwise route exchanges are required. This peering relationship poses a scalability problem as the number of ISPs becomes very large. A route server is introduced to solve the scalability problem. Each ISP sends routing information to the route server, which knows the policies of every ISP. The route server in turn delivers the processed routing information to the ISPs. Public peering points were historically plagued with congestion problems. Nowadays, most major national ISPs increasingly use private peering points connecting two ISPs directly to exchange traffic. A key issue at these peering points is the enforcement of routing policies that dictate what traffic is exchanged. For example, ISP A and ISP B may have a peering agreement whereby traffic originating at A and destined to B can be exchanged, but traffic from A to other ISPs may not.

Note that a national service provider also has the capability of interconnecting a customer's various sites by using its own IP network, in which case the customer's sites appear as a single private network. This configuration is an example of a virtual private network (VPN).

Small office and home office (SOHO) users obtain packet access through ISPs. The access is typically through modem dial-up, but it could be through ADSL, ISDN, or cable modem. When a customer connects to an ISP, the customer is assigned an IP address for the duration of the connection.[1] Addresses are shared in this way because the ISP has only a limited number of addresses. If the ISP is only a local provider, then it must connect to a regional or national provider and eventually to a NAP.

Thus we see that a multilevel hierarchical network topology arises for the Internet, which is much more decentralized than traditional telephone networks. This topology comprises multiple domains consisting of routers interconnected by point-to-point data links, LANs, and wide area networks.

The principal task of a packet-switching network is to provide connectivity among users. The preceding description of the existing packet-switching network infrastructure reveals the magnitude of this task. Routers exchange information among themselves and use routing protocols to build a consistent set of routing tables that can be used in the routers to direct the traffic flows in these networks. The routing protocols must adapt to changes in network topology due to the introduction of new nodes and links or to failures in equipment. Different routing algorithms are used within a domain and between domains. A key concern here is that the routing tables result in stable traffic flows that make efficient use of network resources. Another concern is to keep the size of routing tables manageable even as the size of the network continues to grow at a rapid pace. In this chapter we show how hierarchical addressing structures can help address this problem. A third concern is to deal with congestion that inevitably occurs in the network. It makes no sense to accept packets into the network when they are likely to be discarded. Thus when congestion occurs inside the network, that is, buffers begin filling up as a result of a surge in traffic or a fault in equipment, the network should react by applying congestion control to limit access to the network only to traffic that is likely to be delivered. A final concern involves providing the capability to offer Quality-of-Service guarantees to some packet flows. We deal with these topics also in the remainder of the chapter.

7.3 DATAGRAMS AND VIRTUAL CIRCUITS

A packet-switching network is usually represented as a cloud with multiple input sources and output destinations as shown in Figure 7.8. The network can be viewed as a generalization of a physical cable in the sense of providing connectivity among multiple users. Unlike a cable, a packet-switching network is geographically distributed and consists of a graph of transmission lines (links) interconnected by packet switches

[1] The Dynamic Host Configuration Protocol (DHCP) provides users with temporary IP addresses and is discussed in Chapter 8.

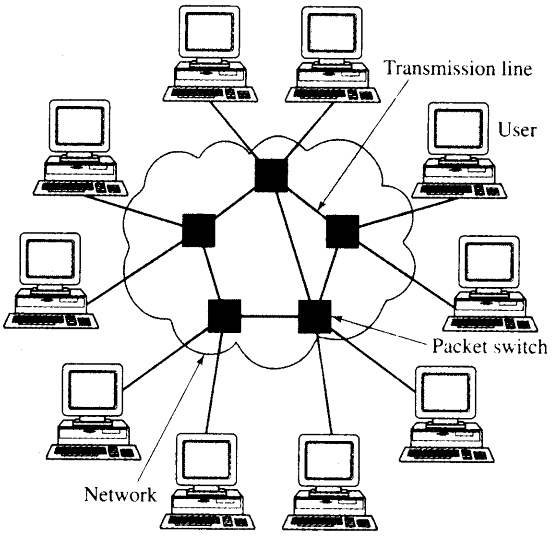


FIGURE 7.8 Switched network.

(nodes). These transmission and switching resources are configured to enable the flow of information among users.

Packet-switching networks provide for the interconnection of sources to destinations on a dynamic basis. Resources are typically allocated to an information flow only when needed. In this manner the resources are shared among the community of users resulting in efficiency and lower costs. In this section we discuss the two fundamental approaches to transferring information over a packet-switching network. A connection-oriented network involves setting up a connection across the network before information can be transferred. The setup procedure typically involves the exchange of signaling messages and the allocation of resources along the path from the source to the destination for the duration of the connection. A connectionless network does not involve setting up connections. Instead a packet of information is routed independently from node to node until the packet arrives at its destination. Both approaches involve the use of packet switches to direct packets across the network.

7.3.1 Connectionless Packet Switching

Packet switching has its origin in **message switching**, where a message is relayed from one switch to another until the message arrives at its destination, as shown in Figure 7.9. A message switch typically operates in the **store-and-forward** fashion whereby a message has to be completely received (and thus stored) by the switch before it can be forwarded to the next switch. At the source each message has a header attached to it to provide source and destination addresses. CRC checkbits are attached to detect errors. The message is transmitted in its entirety from one switch to the next switch. Each switch performs an error check, and if no errors are detected, the switch examines the header to determine the next hop in the path to the destination. If errors are detected, a retransmission may be requested. After the next hop is determined, the message waits

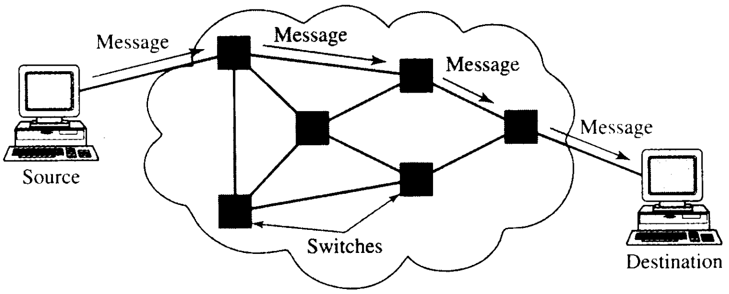


FIGURE 7.9 Message switching.

for transmission over the corresponding transmission line. Because the transmission lines are shared, the message may have to wait until previously queued messages are transmitted. Message switching does not involve a call setup. Message switching can achieve a high utilization of the transmission line. This increased utilization is achieved at the expense of queueing delays. In addition, loss of messages may occur when a switch has insufficient buffering to store the arriving message.[2] End-to-end mechanisms are required to recover from these losses.

Figure 7.10 shows the minimum delay that is incurred when a message is transmitted over a path that involves two intermediate switches. The message must first traverse the link that connects the source to the first switch. We assume that this link has a propagation delay of τ seconds.[3] We also assume that the message has a transmission time of T seconds. The message must next traverse the link connecting the two switches, and then it must traverse the link connecting the second switch and the destination. For simplicity we assume that the propagation delay and the bit rate of the transmission lines are the same. It then follows that the minimum end-to-end message delay is $3\tau + 3T$. Note that this delay does not take into account any queueing delays that

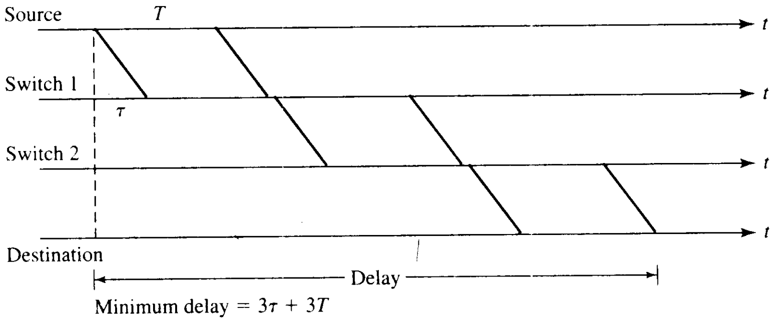


FIGURE 7.10 Delays in message switching.

[2] The trade-offs between delay and loss are explored in Chapter 5, Section 5.7.1.

[3] The propagation delay is the time that elapses from when a bit enters a transmission line to when it exits the line at the other end.

may be incurred in the various links waiting for prior messages to be transmitted. It also does not take into account the times required to perform the error checks or any associated retransmissions.

EXAMPLE Long Messages versus Packets

Suppose that we wish to transmit a large message ($L = 10^6$ bits) over two hops. Suppose that the transmission line in each hop has an error rate of $p = 10^{-6}$ and that each hop does error checking and retransmission. How many bits need to be transmitted using message switching?

If we transmit the message in its entirety, the probability that the message arrives correctly after the first hop, assuming independent bit errors, is

$$P_c = (1-p)^L = (1-10^{-6})^{1000000} \approx e^{-Lp} = e^{-1} \approx 1/3 \qquad (7.1)$$

Therefore, on the average it will take three tries to get the message over the first hop. Similarly, the second hop will require another three full message transmissions on the average. Thus a total of 6 Mbits will need to be transmitted to get the 1 Mbit message across the two hops.

Now suppose that the message is broken up into ten 10^5-bit packets. The probability that a packet arrives correctly after the first hop is

$$P_c' = (1-10^{-6})^{100000} \approx e^{-1/10} \approx 0.90 \qquad (7.2)$$

Thus each packet needs to be transmitted $1/0.90 = 1.1$ times on the average. The message gets transmitted over each hop by using an average of 1.1 Mbits of transmission resource. The total number of bits transmitted over the two hops is then 2.2 Mbits.

The preceding example reiterates our observation on ARQ protocols that the probability of error in a transmitted block increases with the length of the block. Thus very long messages are not desirable if the transmission lines are noisy because they lead to a larger rate of message retransmissions. This situation is one reason that it is desirable to place a limit on the maximum size of the blocks that can be transmitted by the network. Thus long messages should be broken into smaller blocks of information, or *packets*.

Message switching is also not suitable for interactive applications because it allows the transmission of very long messages that can impose very long waiting delays on other messages. By placing a maximum length on the size of the blocks that are transmitted, packet switching limits the maximum delay that can be imposed by a single packet on other packets. Thus packet switching is more suitable than message switching for interactive applications.

In the **datagram**, or **connectionless packet-switching** approach, each packet is routed independently through the network. Each packet has an attached header that provides all of the information required to route the packet to its destination. When a packet arrives at a packet switch, the destination address (and possibly other fields) in the header are examined to determine the next hop in the path to the destination.

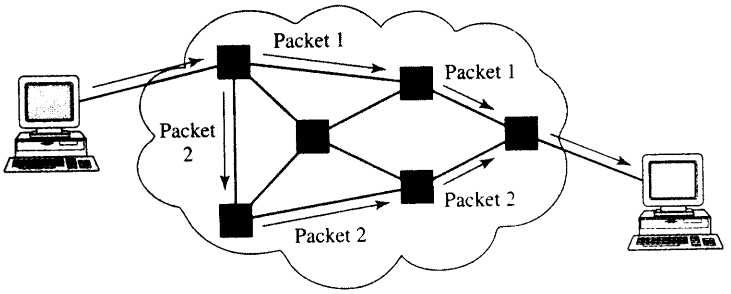


FIGURE 7.11 Datagram packet switching.

The packet is then placed in a queue to wait until the given transmission line becomes available. By sharing the transmission line among multiple packets, packet switching can achieve high utilization at the expense of packet queueing delays. We note that routers in the Internet are packet switches that operate in datagram mode.

Because each packet is routed independently, packets from the same source to the same destination may traverse different paths through the network as shown in Figure 7.11. For example, the routes may change in response to a network fault. Thus packets may arrive out of order, and resequencing may be required at the destination.

Figure 7.12 shows the minimum delay that is incurred by transmitting a message that is broken into three separate packets. Here we assume that the three packets follow the same path and are transmitted in succession. We neglect the overhead due to headers and suppose that each packet requires $P = T/3$ seconds to transmit. The three packets are transmitted successively from the source to the first packet switch.

The first packet in Figure 7.12 arrives at the first switch after $\tau + P$ seconds. Assuming that the packet arrives correctly, it can begin transmission over the next hop after a brief processing time. The first packet is received at the second packet switch at time $2\tau + 2P$. Again we assume that the packet begins transmission over the final hop after a brief processing time. The first packet then arrives at the destination at time $3\tau + 3P$. As the first packet traverses the network, the subsequent packets follow

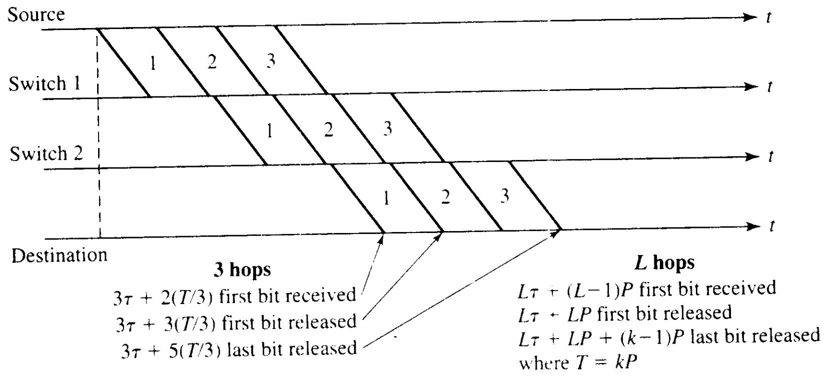

Source
Switch 1
Switch 2
Destination
t
t
t
t
1
2
3
1
2
3
1
2
3
3 hops
$3\tau + 2(T/3)$ first bit received
$3\tau + 3(T/3)$ first bit released
$3\tau + 5(T/3)$ last bit released
L hops
$L\tau + (L-1)P$ first bit received
$L\tau + LP$ first bit released
$L\tau + LP + (k-1)P$ last bit released
where $T = kP$

FIGURE 7.12 Delays in datagram packet switching.

immediately, as shown in the figure. In the absence of transmission errors, the final packet will arrive at the destination at time $3\tau + 3P + 2P = 3\tau + 5P = 3\tau + T + 2P$, which is less than the delay incurred in the message switching example in Figure 7.10. In general, if the path followed by a sequence of packets consists of L hops with identical propagation delays and transmission speeds, then the delay incurred by a message that consists of k packets is given by

$$L\tau + LP + (k - 1)P \tag{7.3}$$

In contrast, the delay incurred using message switching is

$$L\tau + LT = L\tau + L(kP) \tag{7.4}$$

Thus message switching involves an additional delay of $(L - 1)(k - 1)P$. We note that the above delays neglect the queueing and processing times at the various hops in the network.

Figure 7.13 shows a routing table for a datagram network. Each table contains an entry for each possible destination in the network. Each entry in a routing table specifies the next hop that is to be taken by packets with the associated destination. When a packet arrives, the destination address in the header is used to perform a table lookup. The result of the lookup determines the output port to which the packet must be forwarded. For the datagram network to operate correctly, the routing tables must implement a consistent set of routes so that each packet is correctly routed hop-by-hop across the network. When the number of destinations becomes very large, the size of the routing table may exceed the practical implementation limit. We will see how the concept of address aggregation can be used to reduce the size of the routing table.

Destination address	Output port
0785	7
1345	12
1566	6
2458	12

FIGURE 7.13 Routing table in connectionless packet switching.

EXAMPLE IP Internetworks

The Internet Protocol provides for the connectionless transfer of packets across an interconnected set of networks called the Internet. In general the component networks may use different protocols so the objective of IP is to provide communications across these dissimilar networks. Each device that is attached to the Internet has a two-part address: a network part and a host part. To transmit an IP packet, a device sends an IP packet encapsulated using its local network protocol to the nearest router. The routers are packet switches that act as gateways between the component networks. The router performs a route lookup algorithm on the network part of the destination address of the packet to determine whether the destination is in an immediately accessible network or, if not, to determine the next router in the path to the destination. The router then forwards the IP packet across the given network by encapsulating the IP packet using the format and protocol of the given network. In other words, IP treats the component networks as data link layers whose role is to transfer the packet to the next router or to the destination. IP packets are routed in connectionless fashion from router to router until the destination is reached.

7.3.2 Virtual-Circuit Packet Switching

Virtual-circuit packet switching involves the establishment of a fixed path, often called a **virtual circuit** or a **connection**, between a source and a destination prior to the transfer of packets, as shown in Figure 7.14. As in circuit switching, the virtual-circuit setup procedure usually takes place before any packets can flow through the network.[4] Unlike circuit switching where the circuits reside at the physical layer, virtual circuits reside at the network layer. Section 7.4.2 shows how a virtual circuit can be created by appropriately configuring the routing tables in switches along the path.

Figure 7.15 shows the delay that is incurred when a message broken into three packets is transmitted over a virtual circuit. Observe that the minimum delay in virtual-circuit packet switching is similar to that in datagram packet switching, except for an additional delay required to set up the virtual circuit.

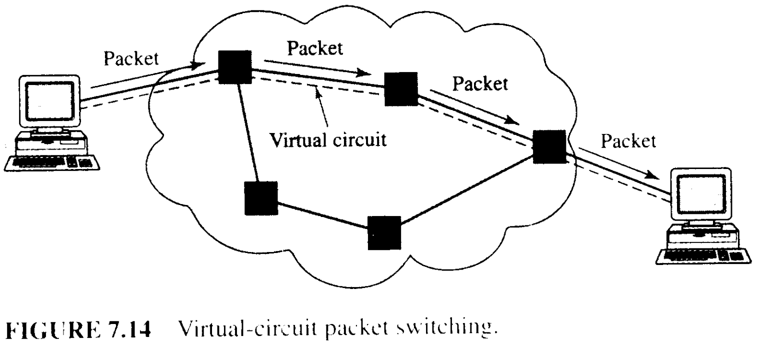


FIGURE 7.14 Virtual-circuit packet switching.

[4] In some cases *permanent* virtual circuits are established a priori.

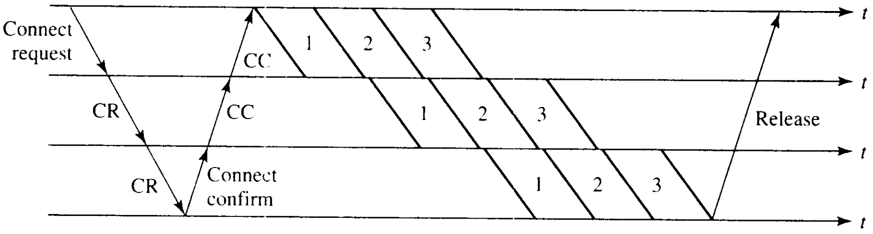


FIGURE 7.15 Delays in virtual-circuit packet switching.

The virtual-circuit setup procedure first determines a path through the network and then sets parameters in the switches by exchanging *connect-request* and *connect-confirm* messages, as shown in Figure 7.16. Every switch along the path is involved in the exchange of signaling messages to set up the virtual circuit. If a switch does not have enough resources to set up a virtual circuit, the switch alternately responds to a connect-request message with a connect-reject message and the setup procedure fails. In general, in virtual-circuit packet switching, buffer and transmission resources need not be dedicated explicitly for the use of the virtual circuit, but the number of flows admitted may be limited to control the load on certain links. As in the datagram approach, packets from many flows share the same transmission line. Unlike the datagram approach, virtual-circuit packet switching guarantees the order of the packets since packets for the same source-destination pair follow the same path.[5]

In datagram packet switching each packet must contain the full address of the source and destination. In large networks these addresses can require a large number of bits and result in significant packet overhead and hence wasted transmission bandwidth. One advantage of virtual-circuit packet switching is that *abbreviated headers* can be used. The call setup procedure establishes a number of entries in routing tables located in the various switches along the path. At the input to every switch, the virtual circuit is identified by a **virtual-circuit identifier (VCI)**. When a packet arrives at an input port, the VCI in the header is used to access the table, as shown in the example in Figure 7.17. The table lookup provides the output port to which the packet is to be forwarded and the VCI that is to be used at the input port of the next switch. Thus the call setup procedure sets up a chain of pointers across the network that direct the flow of packets in a connection. The table entry for a VCI can also specify the type of priority that is to be given to the packet by the scheduler that controls the transmissions in the next output port.

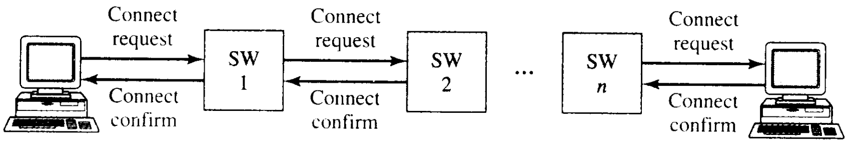


FIGURE 7.16 Signaling message exchanges in call setup.

[5] However, virtual-circuit packet switching may or may not provide reliable packet delivery service.

Input VCI	Output port	Output VCI
12	13	44
15	15	23
27	13	16
58	7	34

Entry for packets with VCI 15

FIGURE 7.17 Example of virtual-circuit routing table for an input port.

The number of bits required in the header in virtual-circuit switching is reduced to the number required to represent the maximum number of simultaneous virtual circuits over an input port. This number is much smaller than the number required for full destination network addresses. This factor is one of the advantages of virtual-circuit switching relative to datagram packet switching. In addition, the use of abbreviated headers and hardware-based table lookup allows fast processing and forwarding of packets. Virtual-circuit packet switching can do a table lookup through direct indexing; datagram packet switching traditionally was much slower because the more demanding lookup procedure required software processing of the header to determine the next hop in the route. This situation has somewhat changed with recent developments in fast lookup algorithms and hardware-based lookup engines.

Another advantage of virtual-circuit packet switching is that resources can be allocated during call setup. For example, a certain number of buffers may be reserved for a virtual circuit at every switch along the path, and a certain amount of bandwidth can be allocated at each link in the path. In addition, the call setup process ensures that a switch is able to handle the volume of traffic that is allowed over every transmission link. In particular, a switch may refuse a virtual circuit over a certain link when the delays or link utilization exceed certain thresholds.

However, virtual-circuit packet switching does have disadvantages relative to the datagram approach. The switches in the network need to maintain information about the flows that pass the switches. Thus, the amount of required "state" information grows very quickly with the number of flows. Another potential disadvantage is evident when failures occur. In the case of virtual-circuit packet switching, when a fault occurs in the network all affected connections must be set up again.

A modified form of virtual-circuit packet switching, called **cut-through packet switching**, can be adopted when retransmissions are not used in the underlying data link control. In this modified form, a packet is forwarded as soon as the header is received and the table lookup is carried out. As shown in Figure 7.18, the minimum delay in transmitting the message is then reduced to approximately the sum of the propagation delays in the various hops plus the one-message transmission time. This scenario assumes that all lines are available to transmit the packet immediately.

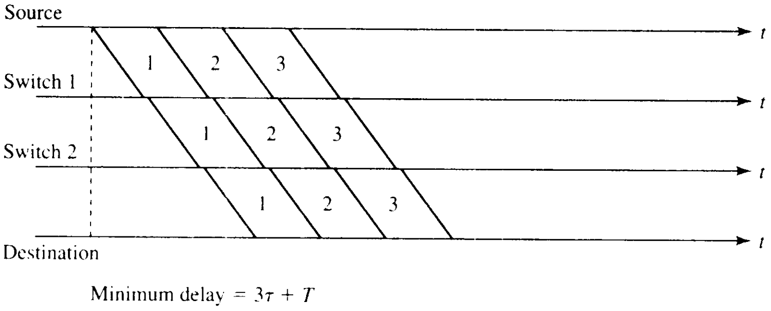


FIGURE 7.18 Cut-through packet switching.

Cut-through packet switching may be desirable for applications such as speech transmission, which has a delay requirement but can tolerate some errors. Cut-through packet switching is also appropriate when the transmission is virtually error free, as in the case of optical fiber transmission, so that hop-by-hop error checking is unnecessary.

EXAMPLE ATM Networks

ATM networks provide for the connection-oriented transfer of information across a network. ATM requires all user information to be converted into fixed-length packets called cells. A connection setup phase precedes the transfer of information. During this setup a negotiation takes place in which the user specifies the type of flow that is to be offered to the network, and the network commits to some quality of service that is to be provided to the flow. The connection setup involves setting up a path across the network and allocating appropriate resources along the path.

An ATM connection is defined in terms of a chain of local identifiers called VCIs, which identify the connection in each link along the path. Cells are forwarded by ATM switches that perform a table lookup on the VCI to determine the next output port and the VCI in the next link. ATM assumes low-error rate optical connections so error control is done only end to end. We discuss ATM in more detail in Section 7.6.

FLOWS, RESERVATIONS, AND SHORTCUTS

Here we note the emergence of packet-switching approaches that combine features of datagrams and virtual circuits. These hybrid approaches are intended for packet-switching networks that handle a mix of one-time packet transfers (for which datagram mode is appropriate) and sustained packet flows such as long file transfers, Web page downloads, or even steady flows as in audio or video streaming (for which virtual-circuit forwarding is appropriate). In essence these systems attempt to identify longer-term packet flows and to set up shortcuts by using forwarding tables so that packets in a flow are forwarded immediately without the need for route lookup processing. This approach reduces the delay experienced in the packet switch and is discussed further in Chapter 10. Resource reservation procedures for allocating resources to long-term flows have also been developed for datagram networks. We also discuss this in Chapter 10.

7.3.3 Structure of a Packet Switch

A packet switch performs two main functions: *routing* and *forwarding*. The routing function uses algorithms to find a path to each destination and store the result in a routing table. The forwarding function processes each incoming packet from an input port and forwards the packet to the appropriate output port based on the information stored in the routing table. In this section we discuss the basic structure of a packet switch and explain how a packet switch performs these two basic functions. We also explain how a packet switch implements the layer 1, 2, and 3 functions indicated in Figure 7.3.

Figure 7.19a shows a generic packet switch consisting of input ports, output ports, an interconnection fabric, and a switch controller. Input ports and output ports are normally paired. A line card often contains several input/output ports so that the capacity of the link connecting the line card to the interconnection fabric, which is typically of high speed, is fully utilized. The line card implements physical and data link layer functions, as well as certain network layer functions. Thus the line card is concerned with symbol timing, line coding, framing, physical layer addressing, and error checking. To

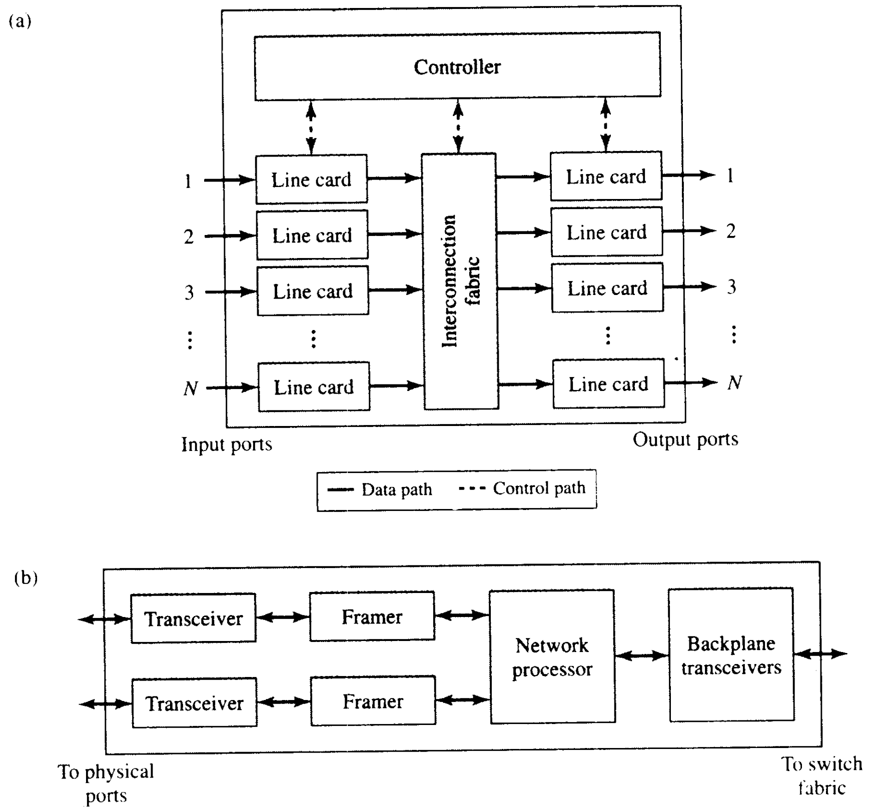


FIGURE 7.19 (a) Components of a generic packet switch and (b) organization of a line card.

handle a broadcast network, the line card may also supports a medium access control protocol typically implemented by a special-purpose chip set. In many cases, network-layer routing tables may also reside in the line card, and a special-purpose engine is needed to perform a fast table lookup to determine the output port and other relevant information. Finally, the line card also contains some buffers and the associated scheduling algorithms (scheduling is described in Section 7.7). The typical organization of a line card made up of various chip sets is shown in Figure 7.19b. Here a programmable network processor performs packet-related tasks such as table lookup and packet scheduling.

The controller in a packet switch contains a general-purpose processor to carry out a number of control and management functions depending on the type of packet switching. For example, the controller in a packet switch operating in a connectionless mode typically executes some routing protocols, while the controller in a packet switch operating in a connection-oriented mode may also be responsible for handling signaling messages. Acting as a central coordinator, the controller also communicates with each line card and the interconnection fabric so that various internal parameters can be configured and maintained.

The function of the interconnection fabric is to transfer packets between the line cards. Note that Figure 7.19a shows an "unfolded" version of the switch where the line cards appear twice, once with input ports and again with output ports. In the actual implementation the receive (ingress) and transmit (egress) functions take place in a single line card (the reader may collapse the ingress and egress line cards into one line card and replace the unidirectional links with bidirectional links). However, the function of various types of switch architectures is easier to visualize this way.

An examination of Figure 7.19 reveals that the interconnection fabric is likely to be the bottleneck if there are many high-speed line cards, since all traffic from the input line cards have to go through to the interconnection fabric. A bus-type interconnection structure whereby packets are transferred serially from input ports to output ports does not scale to large sizes since the speed of the bus has to be about N times faster than the port speed. On the other hand, a crossbar interconnection fabric can transfer packets in parallel between input ports and output ports. For packet switching, buffers need to be added to the crossbar to accommodate packet contention. The buffers can be located at the input ports or output ports, as shown in Figure 7.20.

A crossbar with output buffering needs to run N times faster than the port speed since up to N packets may simultaneously arrive at a particular output. If the output is idle, one packet is transmitted and the rest are temporarily buffered. Because only one packet is allowed to proceed to a particular output with the input-buffering case, the crossbar does not need a speedup. However, input buffering causes another problem. Consider a situation where there are two packets at input buffer *2*, as shown in Figure 7.20. The first packet would like to go to output *3* and the second packet to output *8*. Suppose that a packet from input buffer *1* would also like to go to output *3* at the same time. Suppose that the fabric arbiter decides to transmit the packet from input buffer *1*. Then the first packet from input buffer *2* needs to wait until output *3* has transferred the packet from input buffer *1*. Meanwhile, the second packet has to wait behind the first packet even though output *8* is idle. This situation results in performance degradation of the crossbar with input buffering, and the problem of