delay does not differ significantly from that of the ideal system which corresponds to the $a' = 0$ case.

The walk time t' is determined by several factors. The first factor is the propagation time required for a signal to propagate from one station to another. This time is clearly a function of distance. Another factor is the time required for a station to begin transmitting after it has been polled. This time is an implementation issue. A third factor is the time required to transmit the polling message. These three factors combine to determine the total walk time of the system.

6.4.3 Token-Passing Rings

Polling can be implemented in a distributed fashion on networks with a ring topology. As shown in Figure 6.30, such ring networks consist of station interfaces that are connected by point-to-point digital transmission lines. Each interface acts like a repeater in a digital transmission line but has some additional functions. An interface in the listen mode reproduces each bit that is received at its input at its output after some constant delay, ideally in the order of one bit time. This delay allows the interface to monitor the passing bit stream for certain patterns. For example, the interface will be looking for the address of the attached station. When such an address is observed, the associated packet of information is copied bit by bit to the attached station. The interface also monitors the passing bit stream for the pattern corresponding to a "free token."

When a free token is received and the attached station has information to send, the interface changes the passing token to busy by changing a bit in the

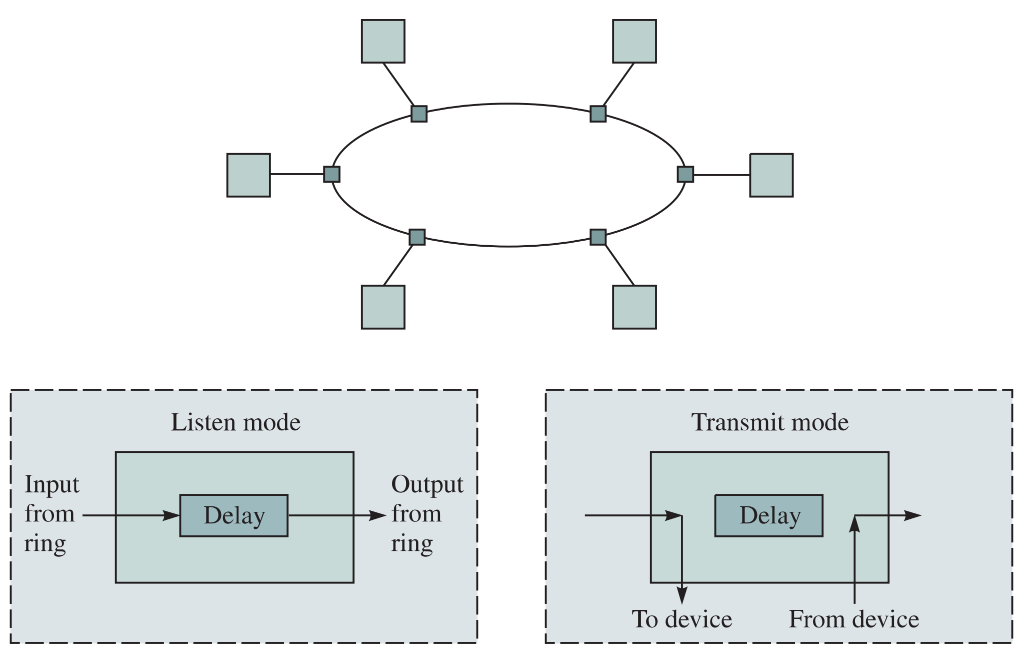

FIGURE 6.30 Token-passing rings

passing stream. In effect, receiving a free token corresponds to receiving a polling message. The station interface then changes to the transmit mode where it proceeds to transmit packets of information from the attached station. These packets circulate around the ring and are copied at the destination station interfaces.

While the station is transmitting its information, it is also receiving information at the input of the interface. If the time to circulate around the ring is less than the time to transmit a packet, then this arriving information corresponds to bits of the same packet that the station is transmitting. When the ring circulation time is greater than a packet transmission time, more than one packet may be present in the ring at any given time. In such cases the arriving information could correspond to bits of a packet from a different station, so the station must buffer these bits for later transmission.

A packet that is inserted into the ring must be removed. One approach to packet removal is to have the destination station remove the packet from the ring. Another approach is to allow the packet to travel back to the transmitting station. This approach is usually preferred because the transmitting station interface can then forward the arriving packet to its attached station, thus providing a form of acknowledgment.

Token rings can also differ according to the method used to reinsert the token after transmission has been completed. There are three approaches to token reinsertion, as shown in Figure 6.31. The main differences between the methods arise when the ring latency is larger than the packet length. The **ring latency** is defined as the number of bits that can be simultaneously in transit around the ring. In the *multitoken operation*, the free token is transmitted immediately after the last bit of the data packet. This approach minimizes the time required to pass a free token to the next station. It also allows several packets to be in transit in different parts of the ring.

The second approach, the *single-token operation*, involves inserting the token after the last bit of the busy token is received back. If the packet is longer than the ring latency, then the free token will be inserted immediately after the last bit of the packet is transmitted, so the operation is equivalent to multitoken

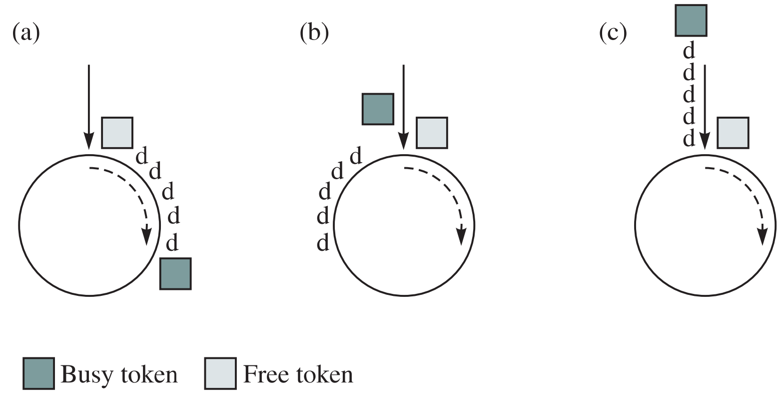

FIGURE 6.31 Approaches to token reinsertion: (a) multitoken, (b) single token, and (c) single packet

operation. However, if the ring latency is greater than the packet length, then a gap will occur between the time of the last bit transmission and the reinsertion of the free token as shown in Figure 6.31. The recovery from errors in the token is simplified by allowing only one token to be present in the ring at any given time.

In the third approach, a *single-packet operation*, the free token is inserted after the transmitting station has received the last bit of its packet. This approach allows the transmitting station to check the return packet for errors before relinquishing control of the token. Note that this approach corresponds to multi-token operation if the packet length is augmented by the ring latency.

The token-ring operation usually also specifies a limit on the time that a station can transmit. One approach is to allow a station to transmit an unlimited number of packets each time a token is received. This approach minimizes the delay experienced by packets but allows the time that can elapse between consecutive arrivals of a free token to a station to be unbounded. For this reason, a limit is usually placed either on the number of packets that can be transmitted each time a token is received or on the total time that a station may transmit information into the ring. These limits have the effect of placing a bound on the time that elapses between consecutive arrivals of a free token at a given station.

The introduction of limits on the number of packets that can be transmitted per token affects the maximum achievable throughput. Suppose that a maximum of one packet can be transmitted per token. Let τ' be the ring latency (in seconds) and a' be the ring latency normalized to the packet transmission time. We then have

$$\tau' = \tau + \frac{Mb}{R} \qquad a' = \frac{\tau'}{E[X]}$$

where τ is the total propagation delay around the ring, b is the number of bit delays in an interface, Mb is the total delay introduced by the M station interfaces, and R is the speed of the transmission lines. The maximum throughput occurs when all stations transmit a packet. If the system uses multitoken operation, the total time taken to transmit the packets from the M stations is $ME[X] + \tau'$. Because $ME[X]$ of this time is spent transmitting information, the maximum throughput is then

$$\rho_{\max} = \frac{ME[X]}{ME[X] + \tau'} = \frac{1}{1 + \tau'/ME[X]} = \frac{1}{1 + a'/M} \text{ for multitoken.}$$

Now suppose that the ring uses single-token operation. Assume that packets are of constant length L and that their transmission time is $X = L/R$. From Figure 6.31 we can see that the effective packet duration is the maximum of X and τ'. Therefore, the maximum throughput is then

$$\rho_{\max} = \frac{MX}{M\max\{X, \tau'\} + \tau'} = \frac{1}{\max\{1, a'\} + \tau'/MX}$$
$$= \frac{1}{\max\{1, a'\} + a'/M} \text{ for single token.}$$

When the packet transmission time is greater than the ring latency, we see that the single-token operation has the same maximum throughput as multitoken operation. However, when the ring latency is larger than the packet transmission time, that is, $a' > 1$, then the maximum throughput is less than that of multitoken operation.

Finally, in the case of single-packet operation the effective packet transmission time is always $E[X] + \tau'$. Therefore, the maximum throughput is given by

$$\rho_{\max} = \frac{ME[X]}{M(E[X] + \tau') + \tau'} = \frac{1}{1 + a'\left(1 + \frac{1}{M}\right)} \text{ for single-packet.}$$

We see that the maximum throughput for single-packet operation is the lowest of the three approaches. Note that when the ring latency is much bigger than the packet transmission time, the maximum throughput of both the single-token and single-packet approaches is approximately $1/a'$. Recall from Figure 6.31 that this situation occurs when the distance of the ring becomes large or the transmission speed becomes very high. Figure 6.32 shows the maximum throughput for the three approaches for different values of a'. It is clear that single-packet operation has the lowest maximum throughput for all values of a'. Multitoken operation, on the other hand, has the highest maximum throughput for all values of a'. In fact, multitoken operation is sensitive to the per hop latency a'/M, not the overall ring latency a'. The figure also shows how single-token operation approaches single-packet operation as a' becomes large.

We conclude this section with a comparison of the average waiting time incurred by packets in several variations of token ring. These waiting time results are taken from [Bertsekas 1992]. The following expressions assume that the packet transmission time X includes the total time required to transmit the token. The normalized mean waiting time for a token ring in which there is no limit on the number of packet transmissions/token is given by

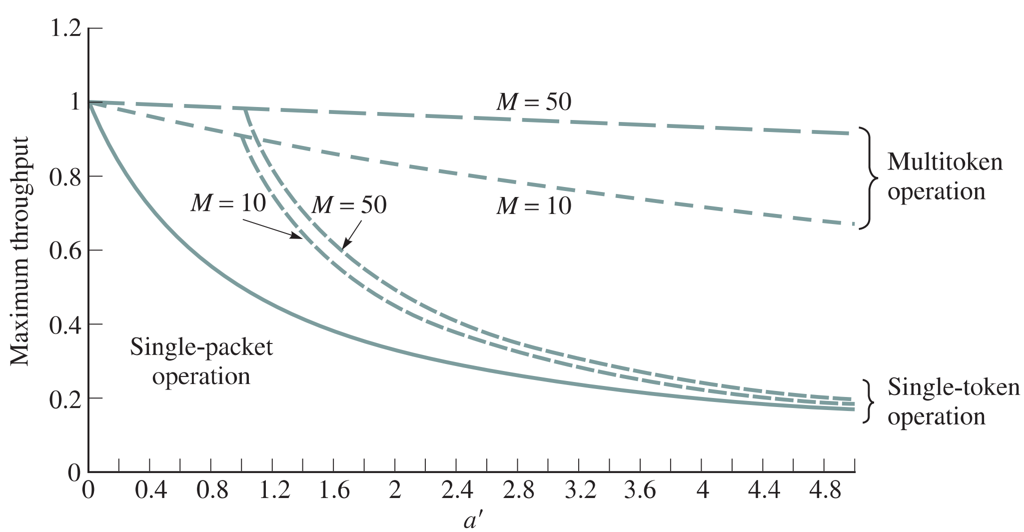

FIGURE 6.32 Throughput comparisons for single packet per token schemes

	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

TABLE 6.4 IEEE 802.3z Gigabit Ethernet medium alternatives

6.6.2 Token-Ring and IEEE 802.5 LAN Standard

Several versions of token-ring networks were developed in the 1970s and 1980s; in token rings a number of stations are connected by point-to-point transmission links in a ring topology. 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, station E in Figure 6.58 has been bypassed by the relay circuit in the wiring center. The star topology also has 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.

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.[10] The station then proceeds to transmit its frame into its

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

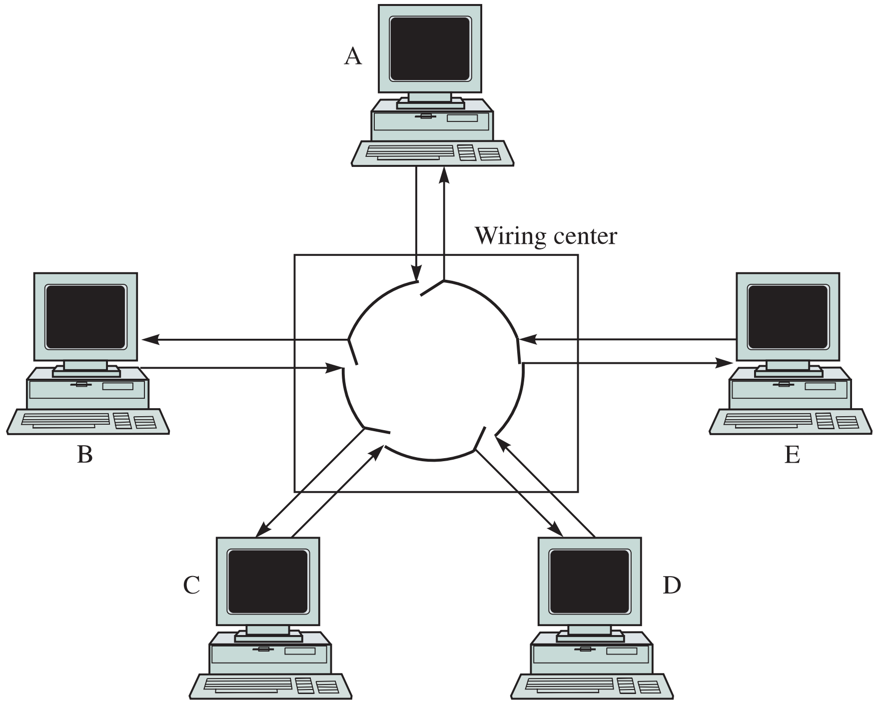

FIGURE 6.58 Token-ring network implemented using a star topology

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.[11] 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.

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

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** is defined as the time that it takes for a bit to travel around the ring and is given by

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

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 $20 \times 100 \times 4 \times 10^6/(2 \times 10^8) + 20(2.5) = 90$ 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. 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

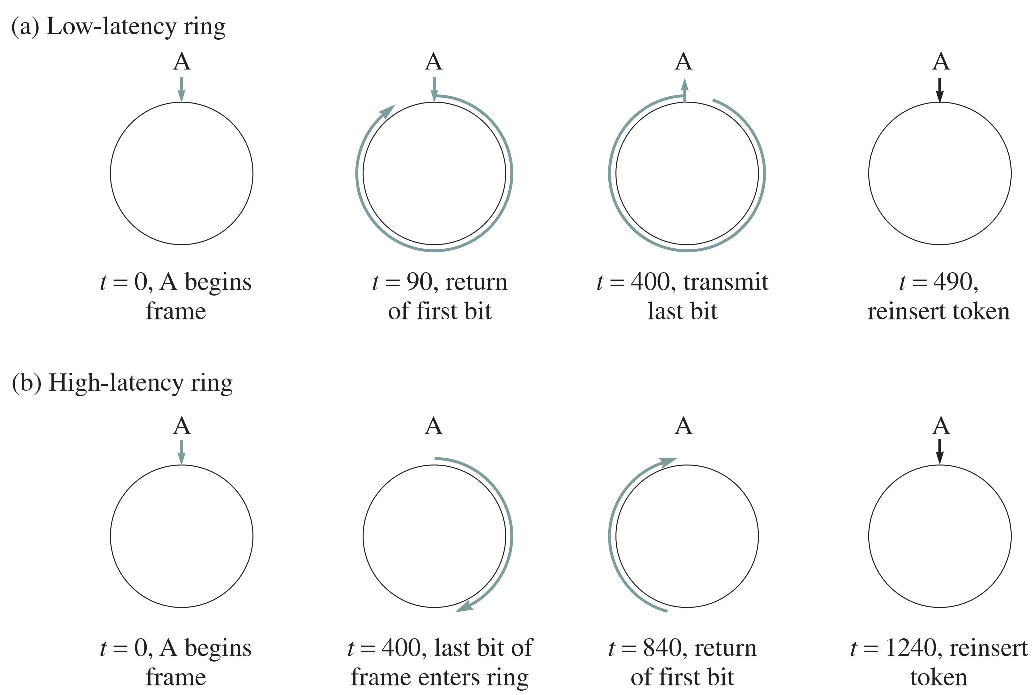

FIGURE 6.59 Ring latency and token reinsertion strategies

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. 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.

Finally suppose that the token reinsertion strategy had been to reinsert the token immediately after the frame transmission is completed. 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

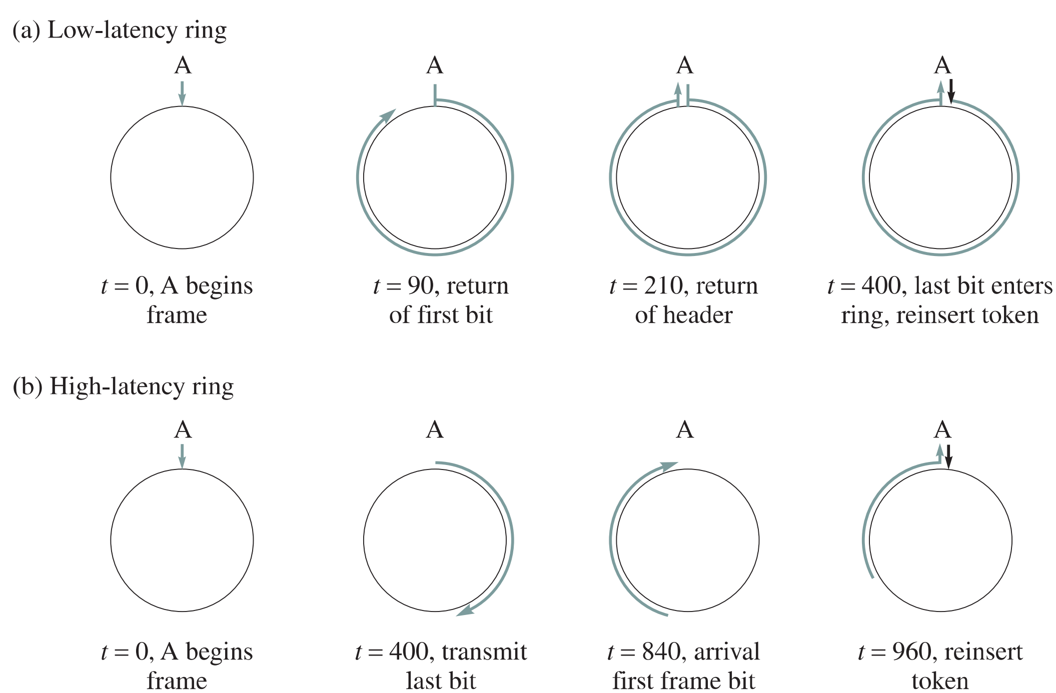

FIGURE 6.60 Reinsert token after header of frame returns

IBM token-ring LAN for 26 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 6.4.3.

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.

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 3.25) 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 interfaces can 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. The IEEE 802.5 standard specifies both 16-bit and 48-bit addressing, using the same format as the IEEE 802.3 Ethernet standard uses. The address fields are followed