5.3 Token Ring Networks

Token ring networks are another widely used family of LANs. The token ring networks were developed by IBM in the early 1980s. We will first explain the operations of some token ring MAC protocols, and we will calculate their efficiency. We will then discuss the IEEE 802.5 standard, which specifies the physical layer and the MAC protocol of a token ring network.

Figure 5.12 is a schematic diagram of a token ring network. The transmission medium is typically a twisted pair or a coaxial cable, although some versions use optical fibers. The MAC protocol of the token ring is as follows: A specific bit pattern, called the *token*, circulates on the ring. When a node wants to transmit, it waits until the token comes by. It then replaces the token with another pattern (SFD) which indicates the *start of frame*, and it appends its packet. The node converts the token into an SFD by monitoring the signal it receives from the ring and by modifying the token while it is stored in the interface buffer (see Figure 5.12) before retransmitting it. For instance, if the token and the SFD are bit patterns that differ by only the value of the last bit, it is enough for the interface buffer to delay the ring signal by one bit. Once the packet has been transmitted, the node transmits the token, which then becomes available to another node. We will see two versions of this protocol: *release after transmission* (RAT) and *release after reception* (RAR). These versions differ by when the node releases the token.

A node that is not transmitting repeats the packets that it receives. To do this, the node recovers the bits from the signal it receives on the input cable

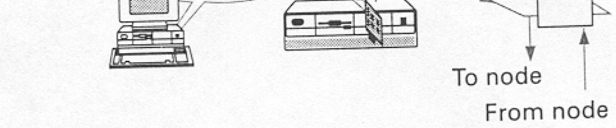


Figure 5.12 Token ring.

The nodes are attached to a unidirectional ring by a network interface. The network interface delays the signal on the ring by a few bits, and it sends the signal to the node. The node either retransmits the delayed signal unmodified on the ring or it modifies it.

and it retransmits these bits on the output cable. The rate of the bit transmission is controlled by a quartz oscillator in the transmitter. This transmission rate differs from the bit reception rate, which is the transmission rate of the upstream node, because no two quartz oscillators have exactly the same rate. This difference in rates means that each node must have a buffer. This node, called the *elasticity buffer*, stores the bits that accumulate when the input rate is higher than the transmission rate. A transmission rate higher than the input rate is handled as follows: the node starts transmitting when the buffer contains a specified number B of received bits; it stops when the complete incoming packet is repeated. The network designer selects a large enough number B for the buffer to prevent the buffer's becoming empty during the packet retransmission.

The steps of the *release after transmission* (RAT) *protocol* are shown in Figure 5.13. With this protocol, the node releases the token as soon as it has completely transmitted the packet.

We will calculate the *efficiency* η_{RAT} of the RAT protocol, defined as the fraction of time that the nodes transmit packets when they use that protocol and when all the nodes have packets to transmit. From the analysis, you will see that

$$\eta_{RAT} \approx \frac{1}{1+\frac{a}{N}} \text{ with } a = \frac{PROP}{TRANSP + TRANST}, \tag{5.11}$$

where N is the number of nodes, $TRANSP$ is the time to transmit a packet, and $PROP$ is the propagation time of a signal around the ring. Consider the timing diagram in Figure 5.14. Node 1 starts transmitting a packet and the transmission lasts $TRANSP$. The transmission of a token lasts $TRANST$. After $TRANSP + TRANST$, the last bit of the token has just been transmitted by node 1. That bit arrives at node 2 at time $TRANSP +, TRANST + PROP(1 \rightarrow 2)$, where $PROP(1 \rightarrow 2)$ designates the propagation time of a signal from node 1 to node 2. The first bit of the packet sent by node 2 leaves that node at time $TRANSP + TRANST + PROP(1 \rightarrow 2) + 1$ where the last 1 refers to the one-bit delay in node 2. That is, we assume that the nodes delay the signal by one bit to be able to replace the token with an SFD. The result of the analysis would not change if the node delay were a small number of bits different from one. Such a small delay is negligible compared with the packet transmission time. Each station takes an equivalent amount of time to transmit a packet and the token. Figure 5.14 shows that the N nodes transmit one packet each in a time

$$\begin{aligned} TOTAL &= N \times (TRANSP + TRANST) + PROP(1 \rightarrow 2) + \ldots + PROP(N \rightarrow 1) + N \\ &\approx N \times (TRANSP + TRANST) + PROP\ . \end{aligned}$$

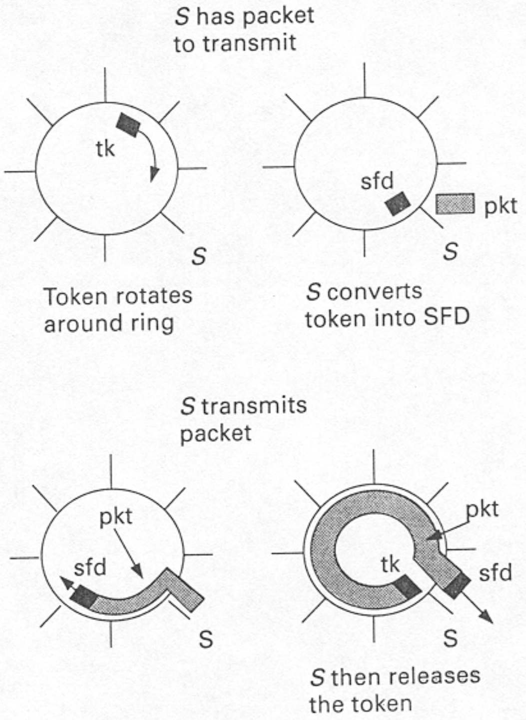


Figure 5.13 Release after transmission (RAT) protocol.

When using the RAT protocol, a node that has a packet to transmit waits for the token. It then converts the token into a start frame delimiter (SFD) and appends its packet. The node transmits a token (releases the token) as soon as it has finished transmitting its packet.

Of this *TOTAL* time, only $N \times TRANSP$ time units are occupied by the actual transmission of packets. Thus, the fraction of time η_{RAT} when the nodes transmit packets is equal to

$$\eta_{RAT} = \frac{N \times TRANSP}{TOTAL} \approx \frac{1}{1 + \frac{a}{N}},$$

as we stated in (5.11). To derive the approximation, we assume that TRANST $\approx$ TRANSP.

The efficiency η_{RAT} of this protocol can be further improved by enabling a node to hold the token and to transmit packets for up to some time, called the maximum *token holding time* (*THT*). By adapting our derivation to this variation, we find that the efficiency becomes $\eta_{RAT}(THT)$ given by

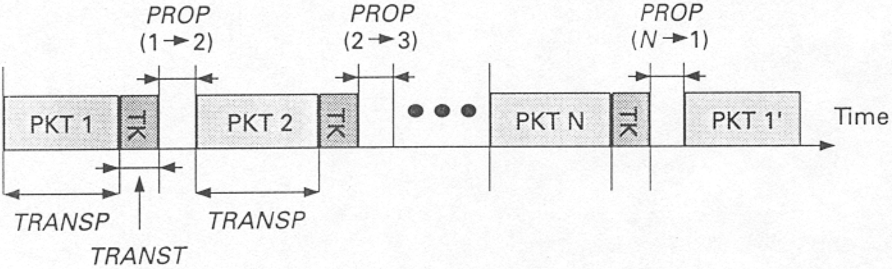


Figure 5.14 Timing diagram of release after transmission.

Time flows from left to right. Node 1 transmits packet 1 (PKT 1) and then the token (TK). The token propagates from node 1 to node 2 during the time epoch marked *PROP* (1 → 2) on the figure. The other nodes follow the same procedure. Eventually, node 1 gets the token again and transmits another packet designated PKT 1′.

$$\eta_{RAT}(THT) \approx \frac{1}{1 + \dfrac{PROP}{N \times THT}} . \qquad (5.12)$$

The timing diagram of Figure 5.14 can be used to calculate the *maximum media access time* (MMAT) for the nodes using the RAT protocol. The MMAT is defined as the maximum time that a node has to wait before it can transmit. The value of MMAT may be an important design element for networks that are used in control equipment. The value of MMAT for nodes on an Ethernet network is infinite since a station can be unlucky and keep on colliding and randomly generating backoff times that are longer than those of the other stations. However, the MMAT for nodes using the RAT protocol is finite. This maximum access time occurs when all the other nodes have a packet to transmit. That situation is shown in Figure 5.15. The figure shows the sequence of transmissions. Assume that an urgent packet arrives at node 1 and must be transmitted as soon as possible. In the worst case how long does the packet have to wait? The worst case is shown in the figure. It occurs for a packet that arrives at node 1 just after that node has started to transmit another packet and when node 1 cannot transmit another packet without exceeding the admissible token holding time, THT. The packet must then wait for the N nodes to transmit and for the token to travel around the ring. If the $N-1$ nodes other than node 1 transmit for the maximum acceptable duration (THT), then the figure shows that the packet must wait for a time $MMAT_{RAT}$ given by

$$MMAT_{RAT} = PROP + N \times TRANST + TRANSP + (N-1) \times THT . \qquad (5.13)$$

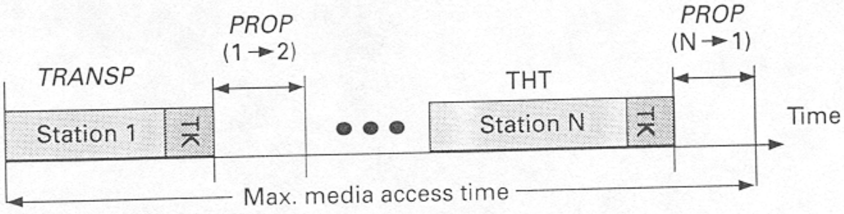


Figure 5.15 Maximum media access time for RAT protocol.

This figure shows the timing diagram of transmissions on a token ring using the RAT protocol. Node 1 must wait until the token comes back to it before it can transmit another packet. In the worst case, node 1 must wait for all the stations to transmit and for the token to travel around the ring before it can transmit another packet. This worst case waiting time is the maximum media access time.

The IEEE 802.5 standard specifies the *release after reception* (RAR) MAC protocol. RAR differs from RAT in that the transmitting node releases the token only after the complete frame has come back. As you will see from our analysis, RAT is significantly more efficient than RAR when the propagation time around the ring is not much smaller than the packet transmission time. However, RAR has the advantage of simplifying acknowledgements: a destination node can signal the correct reception of a packet by appending an acknowledgement at the end of the frame. That acknowledgement is then received by the transmitting node. The steps in the RAR protocol are shown in Figure 5.16.

Let us calculate the efficiency of the RAR protocol. The timing diagram of transmissions on a token ring when the nodes use the RAR protocol is given in Figure 5.17. As in Figure 5.14, this diagram assumes that all the nodes always have packets to transmit. The figure also assumes that the delays introduced by the nodes are negligible. The figure shows that the time between successive transmissions by node 1 is now equal to $TOTAL = N \times (TRANSP + TRANST + PROP) + PROP$. This identity implies that the efficiency is given by

$$\eta_{RAR} = N \times \frac{TRANSP}{TOTAL} \approx \frac{1}{1+a} \text{ with } a := \frac{PROP}{TRANSP}. \tag{5.14}$$

The efficiency of the RAR protocol increases when the nodes can hold the token for up to a specified maximum token holding time THT. The efficiency $\eta_{RAR}(THT)$ of the protocol with this modification is given by (5.14), where $TRANSP$ is replaced by THT, i.e., by

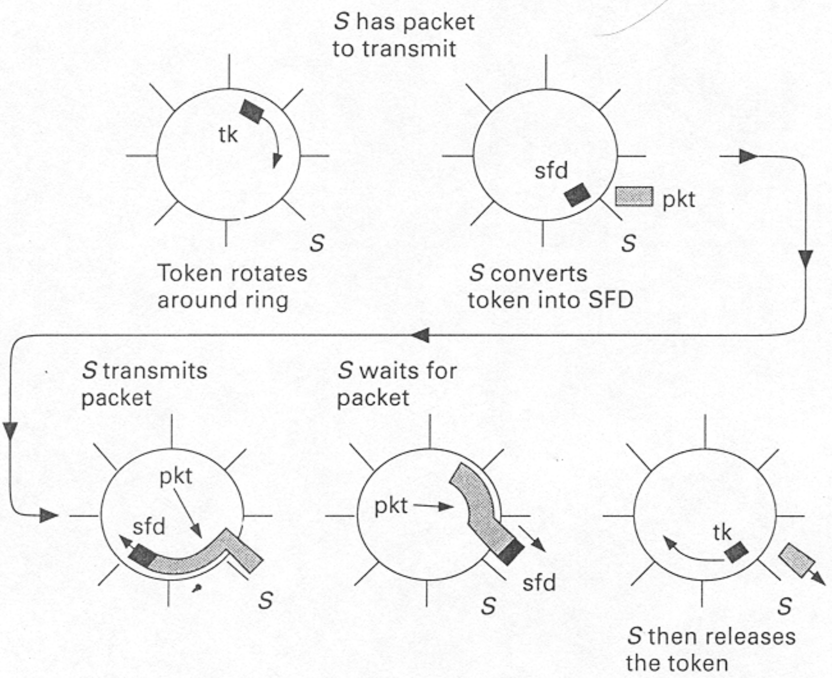


Figure 5.16 Release after reception (RAR) protocol.

A node that has a packet to transmit waits for the token (TK), converts it into a start of frame delimiter (SFD), and appends its packet. The node waits for the complete packet to come back completely before it releases the token.

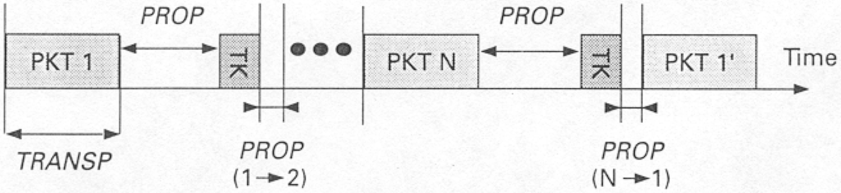


Figure 5.17 Timing diagram for release after reception.

Node 1 transmits a packet PKT 1, then waits for *PROP* for the packet to come back. Node 1 then transmits a token TK which takes *PROP* (1 → 2) to propagate to node 2. These steps are then repeated by the other nodes.

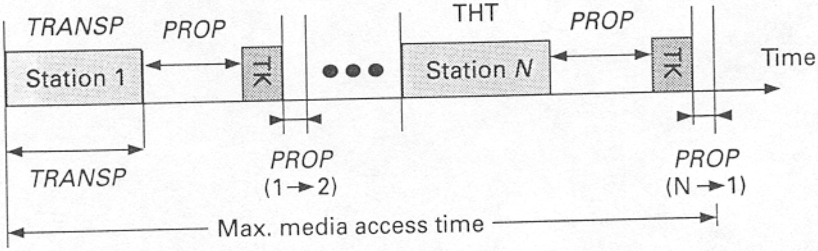


Figure 5.18 Maximum media access time for the RAR protocol.

The timing diagram shows the worst case waiting time for node 1 until it can transmit another packet.

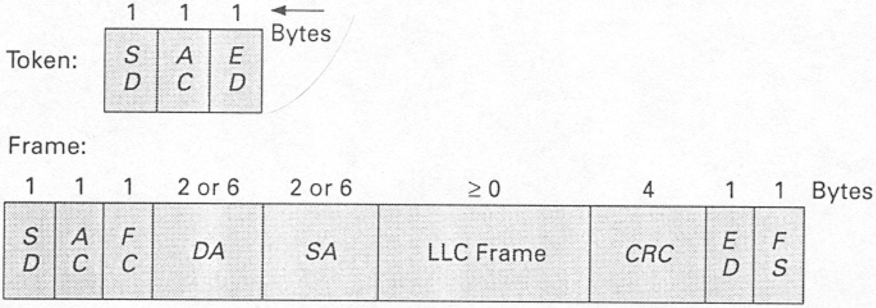
$$\eta_{RAR}(THT) \approx \frac{1}{1 + \dfrac{PROP}{THT}}. \tag{5.15}$$

The maximum media access time $MMAT_{RAR}$ of the RAR protocol can be derived from the timing diagram shown in Figure 5.18, as we did for the RAT protocol. One finds

$$MMAT_{RAR} = (N + 1)\times PROP + N\times TRANST + TRANSP + (N - 1)\times THT. \tag{5.16}$$

Note that the protocols RAT and RAR are substantially more efficient than the CSMA-CD protocol. In some versions of the token ring protocol, the token is released by the receiving node. That is, once the node has received the packet, it transmits a token. This protocol is more efficient than RAR, as you can verify by drawing the timing diagram of the transmissions when nodes use that protocol.

We conclude the discussion of the token ring with some details specific to the IEEE 802.5 standard. These networks transmit at 1 or 4 Mbps over shielded twisted pairs and at 16 Mbps over shielded twisted pairs or optical fibers. Products are also available for transmitting over unshielded twisted pairs. The MAC protocol is release after reception (RAR). Every station can hold the token for up to 10 ms. In addition, the IEEE 802.5 standard provides for multiple packet *priorities*. The node with the packet that has the highest priority gets to transmit before the other nodes. This feature of the IEEE 802.5 is rarely implemented in actual networks. It can be used to speed up the delivery of urgent packets, such as control packets, by giving them priority over less pressing pack-



Abbreviations

SD = Starting delimiter; *AC* = Access control; *FC* = Frame control;
DA = Destination address; *SA* = Source address; *ED* = Ending delimiter;
CRC = Cyclic redundancy check bits; *FS* = Frame status

Figure 5.19 IEEE 802.5 frame structure.

The figure shows the structure of a token and of a regular frame. *SD* is the start frame delimiter, and *AC* is the access control field that indicates whether the bits are a token or a frame. In addition, the *AC* field specifies the priority and reservation levels. *FC* is the frame control field used for monitoring the operations of the ring. The destination address *DA* and source address *SA* are as in IEEE 802.3. *ED* is the end delimiter. *FS* is the frame status field. It indicates whether the destination is down or up and whether it accepted or rejected the frame.

ets, such as those carrying electronic mail messages. The frame structure is shown in Figure 5.19.

Tokens are identified by a special access control (*AC*) field that specifies the priority and reservation levels. To transmit a packet at some priority level, a node needs to wait for a token with a lower level and to place a reservation. If a reservation is already indicated on the token, a node can make another reservation at a higher level. After transmitting at a given level, the node lowers the token priority level to its previous value. The frame control (*FC*) field is used to monitor the ring. One of the nodes is the ring monitor. That node verifies that there is a token and that there is no cycling frame. The possible values of the field *FC* include *monitor_present*, which is issued periodically by the monitor, *reinitialize* to restart the ring, and *I_want_to_be_monitor* issued by a node when it realizes that the monitor is down. The frame status (*FS*) field is set by the destination to signal that it is up and whether it accepted the frame or not.

Token Rings

- Token rings are LANs with nodes that are connected in a ring topology and that use a token passing MAC protocol.
- In the RAT protocol, a node releases the token when it completes the transmission of a packet. The efficiency of this protocol is given by

$$\eta_{RAT}(THT) \approx \frac{1}{1 + \dfrac{PROP}{N \times THT}}. \tag{5.12}$$

- The maximum media access time of the RAT protocol is given by

$$MMAT_{RAT} = PROP + N \times TRANST + TRANSP + (N-1) \times THT. \tag{5.13}$$

- In the RAR protocol, a node waits until the frame it transmitted comes back before releasing the token. The efficiency is given by

$$\eta_{RAR}(THT) \approx \frac{1}{1 + \dfrac{PROP}{THT}}. \tag{5.15}$$

- The maximum media access time of the RAR protocol is

$$\begin{aligned} MMAT_{RAR} = {} & (N+1) \times PROP + N \times TRANST + TRANSP \\ & + (N-1) \times THT. \end{aligned} \tag{5.16}$$

- The IEEE 802.5 MAC protocol is the RAR protocol. The destination can indicate whether it accepted the frame in the *FS* field.

5.4 Token Bus Networks

As the name indicates, the token bus network is a bus network that uses a token-passing MAC protocol. IEEE 802.4 specifies a standard for the physical layer and the MAC sublayer of token bus networks. The MAP (manufacturing automation protocol) of General Motors uses the token bus network. MAP is used to interconnect sensors, tools, and processors with bounded communication delays (see Chapter 8).

A typical layout of a token bus network is shown in Figure 5.20. The figure shows four nodes attached to a common coaxial cable. The nodes use a token-passing mechanism to regulate the access to the cable and to avoid collisions. The figure illustrates one example of the sequence of events. Initially, node