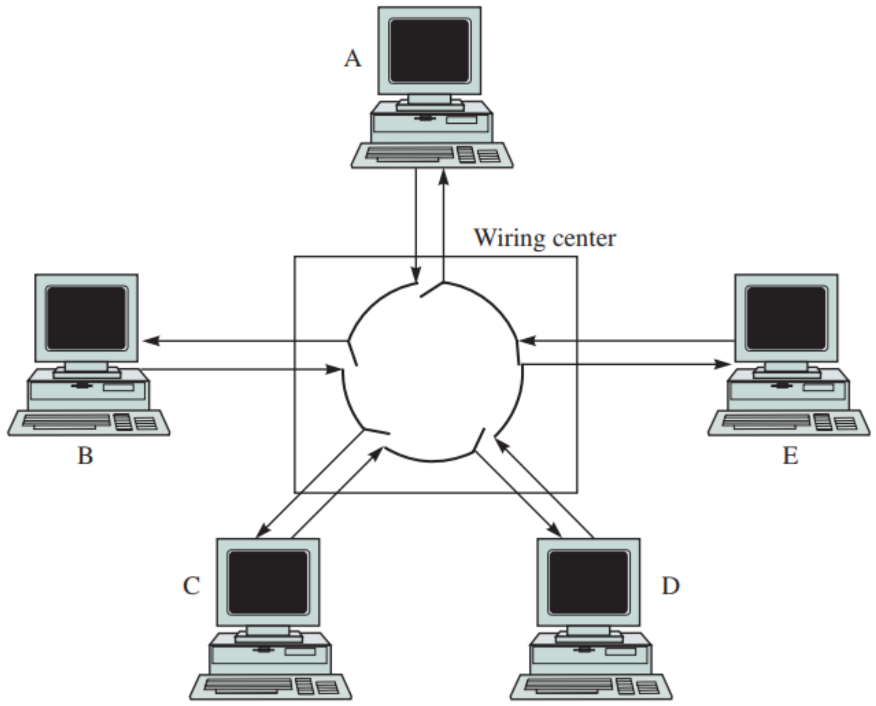
Token Ring Protocol

- 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.*
- 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.



Token Ring Protocol

- **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.

Token Ring Protocol

- **Approach 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**.
- Suppose that the ring has $\boldsymbol{M}$ stations.
- Each station interface introduces $\boldsymbol{b}$ 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 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** is defined as the time that it takes for a bit to travel around the ring and is given by

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

Ring Latency and Token Reinsertion

- Relation between ring latency and the token reinsertion method.

- First suppose that we have a ring that operates at a speed of R (data rate) = 4 Mbps with M (no. of stations) = 20 stations separated by 100 meters and b (latency at each station) = 2.5 bits.

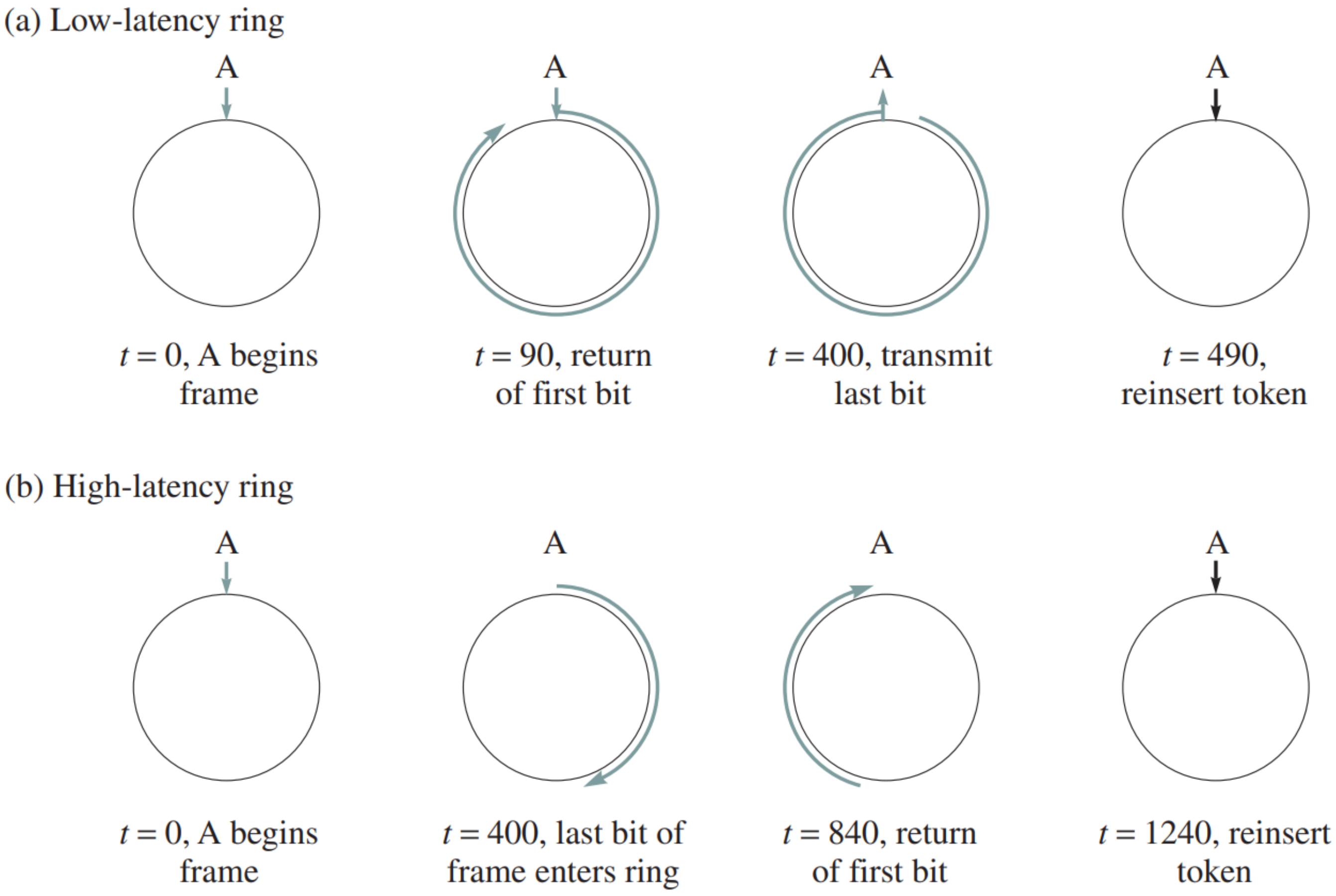
- The ring latency (in bits) is then
  - $20 \times 100 \times 4 \times 10^6/(2 \times 10^8) + 20(2.5) = 90$ bits.
  - *Thus the first bit in a frame returns to the sending station 90 bit times after being inserted.*

- 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 a** 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 b**.
  - 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



- Thus it cab be seen 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.

- 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 a**.
- The sending station can therefore reinsert the token immediately after transmitting bit 400 of the frame.

- **Figure b** 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.