3.3: Token Ring

A significant part of the previous chapter was devoted to classic Ethernet’s collision mechanism for supporting shared media access. After that, it may come as a surprise that there is a simple multiple-access mechanism that is not only collision-free, but which supports fairness in the sense that if N stations wish to send then each will receive 1/N of the opportunities.

That method is **Token Ring**. Actual implementations come in several forms, from Fiber-Distributed Data Interface (FDDI) to so-called “IBM Token Ring”. The central idea is that stations are connected in a ring:

![Token Ring Diagram](https://eng.libretexts.org/Bookshelves/Computer_Science/Book%3A_An_Introduction_to_Computer_Networks_(Dordal)/03%3F...)

Packets will be transmitted in one direction (clockwise in the ring above). Stations in effect forward most packets around the ring, although they can also remove a packet. (It is perhaps more accurate to think of the forwarding as representing the default cable connectivity; non-forwarding represents the station’s momentarily breaking that connectivity.)

When the network is idle, all stations agree to forward a special, small packet known as a **token**. When a station, say A, wishes to transmit, it must first wait for the token to arrive at A. Instead of forwarding the token, A then transmits its own packet; this travels around the network and is then removed by A. At that point (or in some cases at the point when A finishes transmitting its data packet) A then forwards the token.
In a small ring network, the ring circumference may be a small fraction of one packet. Ring networks become “large” at the point when some packets may be entirely in transit on the ring. Slightly different solutions apply in each case. (It is also possible that the physical ring exists only within the token-ring switch, and that stations are connected to that switch using the usual point-to-point wiring.)

If all stations have packets to send, then we will have something like the following:

- A waits for the token
- A sends a packet
- A sends the token to B
- B sends a packet
- B sends the token to C
- C sends a packet
- C sends the token to D
- ...

All stations get an equal number of chances to transmit, and no bandwidth is wasted on collisions. (A station constantly sending smaller packets will send the same number of packets as a station constantly sending larger packets, but the bandwidth will be smaller in proportion to the smaller packet size.)

One problem with token ring is that when stations are powered off it is essential that the packets continue forwarding; this is usually addressed by having the default circuit configuration be to keep the loop closed. Another issue is that some station has to watch out in case the token disappears, or in case a duplicate token appears.

Because of fairness and the lack of collisions, IBM Token Ring was once considered to be the premium LAN mechanism. As such, Token Ring hardware commanded a substantial price premium. But due to Ethernet’s combination of lower hardware costs and higher bitrates (even taking collisions into account), the latter eventually won out.

There was also a much earlier collision-free hybrid of 10 Mbps Ethernet and Token Ring known as Token Bus: an Ethernet physical network (often linear) was used with a token-ring-like protocol layer above that. Stations were physically connected to the (linear) Ethernet but were assigned identifiers that logically arranged them in a (virtual) ring. Each station had to wait for the token and only then could transmit a packet; after that it would send the token on to the next station in the virtual ring. As with “real” Token Ring, some mechanisms need to be in place to monitor for token loss.

Token Bus Ethernet never caught on. The additional software complexity was no doubt part of the problem, but perhaps the real issue was that it was not necessary.