1 Token Bus

In cases where an Ethernet is not appropriate—situations that require predictable performance—a Token Bus can be used. Physically, the network is organized as a Bus, but logically, it is a Ring.

Basically each station only transmits when it has a 'token'. It gets a token from the 'previous' station on the logical ring. The ring is formed via MAC addresses in sorted order; ie: lower addresses send to succeeding higher addresses.

Stations can use different priorities for different types of data.

There needs to be some protocol for ring management: we need to be able to add stations to the ring, remove stations from the ring (either voluntarily, or due to some error).

Also, when stations cannot sense any activity on the bus, and their timer has expired, they need to be able to regenerate a token. There is a scheme where they can send out a 'claim-token' frame (regenerate the token), and figure out the 'leader' who will regenerate the token.

The situation is similar for a newly setup token bus.

At the physical layer, the signal is encoded using Manchester encoding, which is then modulated using FSK—using 2 frequencies (3.75 MHz for Low, and 6.25 Mhz for High)—a 0 is modulated as High to Low frequency, and 1 is modulated as Low to High frequency.

2 Token Ring

The network is a physical ring. A simple 3 byte token circulates from station to station until it encounters a station with data to send. The station keeps the token, and sends a data frame. Each intermediate station examines the frame and destination address, and passes on the frame to the next subsequent node on the ring.

When a station encounters it's own address as the destination address, it flags the frame as 'received' (by changing a few bits), and passes it on. Eventually, when the frame gets to the sender, the sender looks at the changed bits and knows whether the frame was received or not (and whether to resend the frame again later).

The way the connection to the ring is organized is that if a station is off-line, then the data just goes past that station without interruption. The data only travels in one direction though, so if a ring breaks, data will have no way of getting around the ring.

Ring management involves figuring out when a frame has been damaged and asking for the source to retransmit the frame.

There may also be a situation of an orphaned frame—the computer sends out a frame, and then crashes. The frame will go around the ring indefinitely, without anyone removing it. The station holding the token may crash, and not pass on the token.

These problems are usually solved with a monitor station; a single station that monitors the ring (it ensures that no frame circles the ring twice, and can regenerate a token if a certain amount of time has passed without activity).

The ring usually passes on the token right after it is done sending the data. This ensures that the entire ring could be transmitting something at the same time, (ie: no station has to wait for it's frame to cycle through the whole ring).