EECS-122 Lecture 16, September 29, 1999

Token-Ring LANs

Stavros Tripakis

Motivation. The capacity of a network is the total available bit-rate, for instance, 10Mbps in 10BaseT shared Ethernet, 100Mbps in FDDI. The throughput of a network is the amount of data that can actually be transmitted per unit of time, that is, the bit-rate that can actually be achieved. Of course, throughput as defined, depends on conditions like which hosts are transmitting, at what times, to which destinations, and so on. Therefore, throughput is usually implicitly taken to be defined on heavy-load situations, that is, situations where every host has something to send. Then, the efficiency of a network is defined as $\frac{\text{throughput}}{\text{capacity}}$, or more simply as

$$\frac{\text{amount of time in which some data is sent}}{\text{total amount of time needed to send this data}}.$$

Intuitively, the efficiency shows how good the utilization of the network is: what percentage of the network resources is really used to transmit data, and what percentage is overhead.

High efficiency is a desirable feature of networks. Still, it is not the only thing one asks from a network. In particular, real-time applications (e.g., voice, video) need not only a (more-or-less constant) bit-rate (e.g., 64 kbps for voice) but also delay guarantees (e.g., “one packet every 50 ms”). 10 Mbps shared Ethernet does not offer those guarantees, and this is what motivated the development of token-ring networks. Notice that Ethernets are by far more widespread than token-ring LANs. The reasons for this are mainly economic, since Ethernets are typically cheaper than token-rings. Other reasons include the fact that Ethernets are mostly used in environments where real-time requirements are not an important issue: web traffic seems to currently prevail in the Internet, along with other delay-insensitive applications like e-mail, ftp and so on. Finally, switched Ethernets and new versions of Ethernet (100Mbps, Gigabps) are much more efficient provide or will soon provide much higher bit-rates than token-rings.

Basic idea of token rings. Nodes are ordered in a *logical ring*. For instance, if we have N nodes, they might be ordered $1, 2, \ldots, N$, or $1, N, 2, N-1, \ldots$ and so on. This order defines the order in which nodes can transmit data. The logical ring is implemented by circulating a special packet (called the *token*) around the ring. Each node passes the token to its *logical successor*, i.e., the node that follows in the ring order. Assuming the logical ring to be $1, 2, \ldots, N$, node 2 is the successor of node 1 (1 is the *predecessor* of 2), and 1 passes the token to 2, 2 to 3, and so on, until node N passes the token back to node 1. A node can transmit data only if it has the token. Moreover, the node cannot hold the token for arbitrarily long time: this prevents a node from using all the capacity in an unfair way, and allows to prove a bounded access delay until a node has the chance to transmit (i.e., until a node receives the token).

Different token-ring protocols. It is important to note that the logical ring can actually be implemented on top of different *physical* network architectures. There are in fact a number of token-ring protocols, some of which have become standards. Three well-known protocols based on the logical token-ring idea are the following:

- The IBM token-ring protocol which has given rise to the IEEE 802.5 token-ring standard.
- The IEEE 802.4 token-bus standard (defined around 1990).
- The ANSI fiber distributed data interface (FDDI) standard.

Figure 1 illustrates these three architectures. Notice that in the IEEE 802.4 token-bus standard the physical architecture is that of a multiple-access broadcast medium (i.e., a twisted-pair cable) but the medium-access control protocol is still based on a logical token ring.

We now discuss in more detail IEEE 802.5 and FDDI.

IEEE 802.5 Token-ring. The general architecture is illustrated in figure 1. Instead of the nodes (circles) being connected directly one to another, the links are concentrated in a central "box". This allows for a node that has failed, say A, to be isolated from the ring: this is done by by-passing A by connecting the output link of the predecessor of A to the input link of the successor of A (this feature is not shown in the figure).

There is a constant D (parameter of the protocol) which gives the maximum amount of time for which a node is allowed to transmit data when it has the token.

The medium access protocol implemented at each node X is as follows.

- When X receives the token: X sets a timer to D. This timer decreases as time passes.
- While X has the token: X transmits data to its output link and removes at the same time its own data from the input link. When the timer of X becomes zero, X passes the token to its successor node.
- While X awaits the token: X retransmits the data it receives from its input link to its output link, unless if it is data transmitted by X, in which case X removes it from the ring (i.e., does not retransmit it to its output link). If the data received from the input link is destined for X, X copies it to a buffer and passes it to the higher layer (it still retransmits it to the output link).

We now analyze the efficiency of the 802.5 token-ring protocol and determine a bound on the medium-access time for each node. Look at figure 2. We assume that there are n nodes in the ring. T denotes the time taken to transmit the token (the token is a special packet with a fixed size). P denotes the propagation delay from one node to its successor. The time for which a node is allowed to transmit data is at most D, as said above. Therefore, from the time a node (say 1) receives the token until the first next time the same node receives the token, the amount of time that has passed is at most $n \cdot (D + T + P)$ (when all nodes transmit for their maximum allowed time, D). This is an upper bound on the medium access time for a node.

The "useful" part in this time is the time for which the nodes transmit data, that is, $n \cdot D$. The rest, $n \cdot (T + P)$ can be seen as overhead. Therefore, the efficiency of the protocol is $\frac{n \cdot D}{n \cdot (D+T+P)} = \frac{D}{D+T+P}$. Notice that we are computing the efficiency under a heavy-load situation, where every node transmits for the maximum allowed time D.

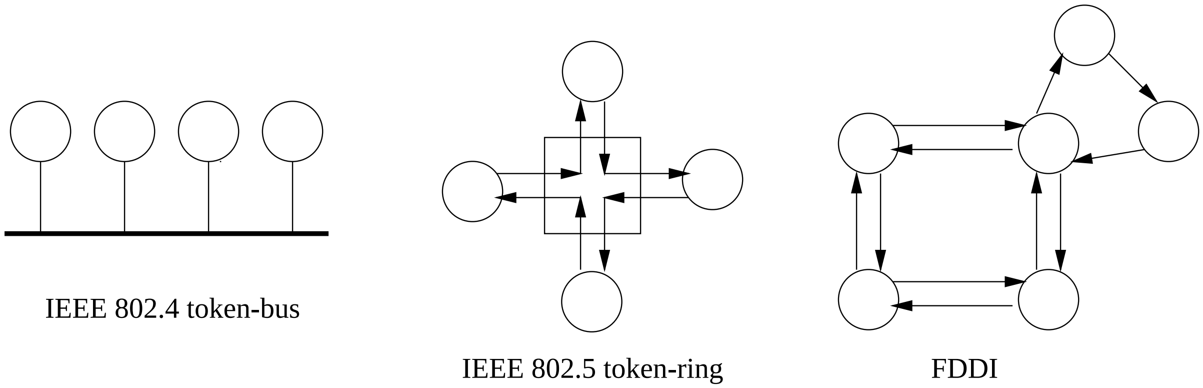

Figure 1: Schematic representation of IEEE 802.5, IEEE 802.4 and FDDI

Typically, $D >> T$, $D >> P$ and T is in the same order of magnitude as P. We could set $T \approx P$ and $D \approx 100 \cdot T$, which gives a rough estimate for the efficiency: $\frac{100 \cdot T}{100 \cdot T + T + T} = \frac{100}{102} \approx 100\%$.

FDDI. The general architecture is illustrated in figure 1.

Traffic in FDDI is classified as *synchronous* (i.e., real-time, such as voice or video) and *asynchronous* (e.g., e-mail). The protocol guarantees a bounded medium access time for synchronous traffic and best-effort delay for asynchronous traffic. Moreover, it uses an adaptive technique for adjusting the maximum amount of time a node can transmit data while the node has the token. This technique makes it possible for a node to use a large percentage of the capacity of the network if other nodes use only a small percentage.

More precisely, FDDI defines a parameter called the *target token rotation time* (TTRT). This can be seen as an "ideal" delay that a node should experience between two successive receptions of the token. The TTRT is actually *voted* among all nodes in the ring by a *bidding* process: initially, each node proposes a value for TTRT and the smaller value "wins" (i.e., becomes the TTRT of the ring).

A node i is allowed to transmit synchronous time for at most S_i time units, when it has the token. Moreover, node i has two timers, the *token-holding timer* THT_i (decreasing timer) and the *token-rotation timer* TRT_i (increasing timer). Node i behaves as follows:

- It receives the token.
- It sets $THT_i := TTRT - TRT_i$.
- It sets $TRT_i := 0$.
- It transmits synchronous traffic for S_i time units.
- If THT_i is still positive (i.e., if $TTRT - TRT_i$ was greater than S_i upon reception of the token) then node i transmits asynchronous traffic until THT_i reaches 0.
- It passes the token to its successor.

So, TRT_i counts the time elapsed since node i has last received the token. If TRT_i is larger than the "ideal" TRT (TTRT) then the protocol is "lagging" behind, so node i is not allowed to transmit asynchronous traffic. If TRT_i is smaller than TTRT (at

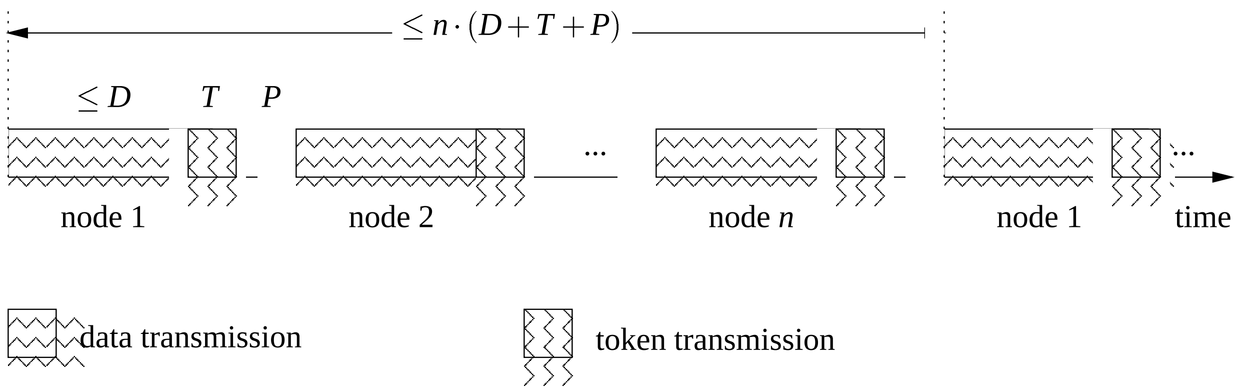

Figure 2: Analysis of IEEE 802.5 token-ring protocol.

least by S_i) then there is "room available" for asynchronous traffic.

Regarding efficiency, we can see that FDDI is very efficient: this is done similarly to the analysis of the IEEE 802.5 token-ring protocol. Regarding medium-access time, we can prove that each node will receive the token at most $2 \cdot TTRT$ time units since the last time it received the token (c.f. section 4.10.3 in Walrand's book).