In More Depth: Ethernet

An Ethernet is essentially a standard bus with multiple masters (each computer can be a master) and a distributed arbitration scheme using collision detection. Most Ethernets are implemented using coaxial cable as the medium. When a particular node wants to use the bus, it first checks to see whether some other node is using the bus; if not, it places a carrier signal on the bus, and then proceeds to transmit. A complication can arise because the control is distributed and the devices may be physically far apart. As a result, two nodes can both check the Ethernet, find it idle, and begin transmitting at once; this is called a *collision*. A node detects collisions by listening to the network when transmitting to see whether a collision has occurred. A collision is detected when the node finds that what it hears on the Ethernet differs from what it transmitted. When collisions occur, both nodes stop transmitting and delay a random time interval before trying to resume using the network—just as two polite people do when they both start talking at the same time. Consequently, the number of nodes on the network is limited—if too many collisions occur, the performance will be poor. In addition, constraints imposed by the requirement that all nodes detect collisions limit the length of the Ethernet and the number of connections to the network. Although this idea sounds like it might not work, it actually works amazingly well and has been central to the enormous growth in the use of local area networks.

8.26 [3 days–1 week] <§§8.3–8.5> Write a program that simulates an Ethernet. Assume the following network system characteristics:

- A transmission bandwidth of 150 Mbits/sec.
- A latency for a signal to travel the entire length of the network and return to its origin of 12 µs. This is also the time required to detect a collision.

Make the following assumptions about the 100 hosts on the network:

- The packet size is 1000 bytes. Each host tries to send a packet after T seconds of computation, where T is exponentially distributed with mean M. Note that the host begins its T seconds of computation only after successfully transmitting a packet.
- If a collision is detected, the host waits a random amount of time chosen from an exponential distribution with a mean of 45 µs.

Simulate and plot the sustained bandwidth of the network compared to the mean time between transmission attempts (M). Also, plot the average wait time between trying to initiate a transmission and succeeding in initiating it (compared to M).

Ethernets actually use an exponential back-off algorithm that increases the mean of the back-off time after successive collisions. Assume that the mean of the distribution from which the host chooses how much to delay is doubled on successive collisions. How well does this work? Is the bandwidth higher than when a single distribution is used? Can the initial mean be lower?