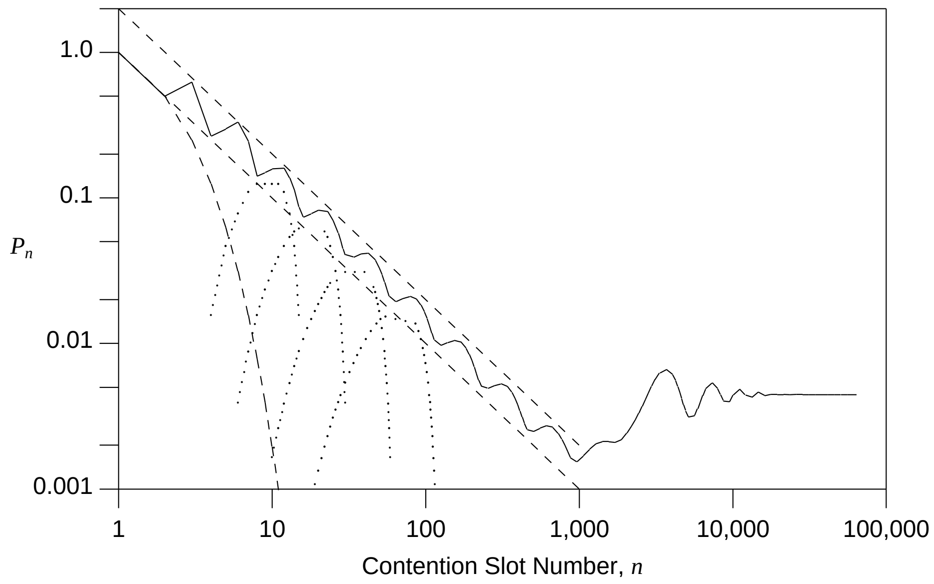

Figure 6: Network load drops off slowly after the ''Big Bang''. Solid line indicates the probability, P_n, that a given host will attempt a retransmission in the nth backoff slot, assuming none of its earlier attempts were successful. Dashed lines bounding P_n are the harmonic functions $1/n$ and $2/n$, respectively. Dashed line at lower left is the geometric function $1/2^n$. ''Arch'' shaped dotted lines show, from left to right, the component functions $P_n(3), \ldots, P_n(6)$.

Given this information about the function P_n, we can calculate the average number of contention slots from the ''Big Bang'' until the first successful packet transmission. From this data it will be clear that reaction time for BEB to respond to a transient overload condition is a *linear* function of the number of active hosts, m. We proceed as follows.

First, we recognize that, if the number of active hosts is large, then we can treat the actions of each of them independently to arrive at the average number of transmission attempts in each backoff slot. This is precisely the type of situation that can be represented by the *binomial distribution*, which in the general case may be expressed as:

$$\beta(M, N, P) \equiv \mathrm{P}[M \text{ out of } N \text{ independent trials, each with bias } P\text{, comes out positive}]$$
$$= \binom{N}{M} P^M (1-P)^{N-M}.$$

In this application, we see that $S_{n,m} \approx \beta(1,m,P_n)$ is the probability that one of the m active hosts acquires the channel for a successful transmission in the nth backoff slot, given that none of them has been successful in an earlier attempt. (The result is only an approximation because we treat the actions of each host in this slot independently, i.e., we assume that each of them may decide to make another attempt in slot n with probability P_n.) Given this function $S_{n,m}$, we can easily calculate L_m, the average number of backoff slots until the first successful transmission occurs, starting from an m-host ''Big Bang'', using the following formula:

$$L_m = \sum_{n=1}^{\infty} n \left[S_{n,m} \prod_{j=1}^{n-1} (1 - S_{j,m}) \right]$$
$$= \sum_{n=1}^{\infty} \left[\sum_{k=1}^{n} S_{n,m} \prod_{j=1}^{n-1} (1 - S_{j,m}) \right] = \sum_{k=1}^{\infty} \left[\sum_{n=k}^{\infty} S_{n,m} \prod_{j=1}^{n-1} (1 - S_{j,m}) \right]$$

$$= \sum_{k=1}^{\infty} \left[\prod_{j=1}^{k-1} (1 - S_{j,m}) \right] \equiv \sum_{k=1}^{\infty} F_{k-1} \,, \qquad (5)$$

where $F_0 = 1$, and $F_k = (1 - S_{k,m}) \cdot F_{k-1}$ for all $k > 0$, represents the probability that all hosts have failed to acquire the network during the first k backoff slot times. Since the function F_k is decreasing at least geometrically fast as k increases, L_m is easy to evaluate despite the infinite summation. Figure 7 shows a plot of L_m as a function of m. Also shown is the sample mean obtained by Monte Carlo simulation, where we have repeated the m-host ‘‘Big Bang’’ experiment until the width of the 95% confidence interval is below 4% of the mean. Notice the excellent agreement between Eq. (5) and the simulation data, especially for $m \geq 4$, which shows that the independence assumption is not very important. Note also that L_m is growing *linearly* with the number of active of hosts, m, with $L_m \approx 2/9 \cdot m$ when $m \gg 1$.

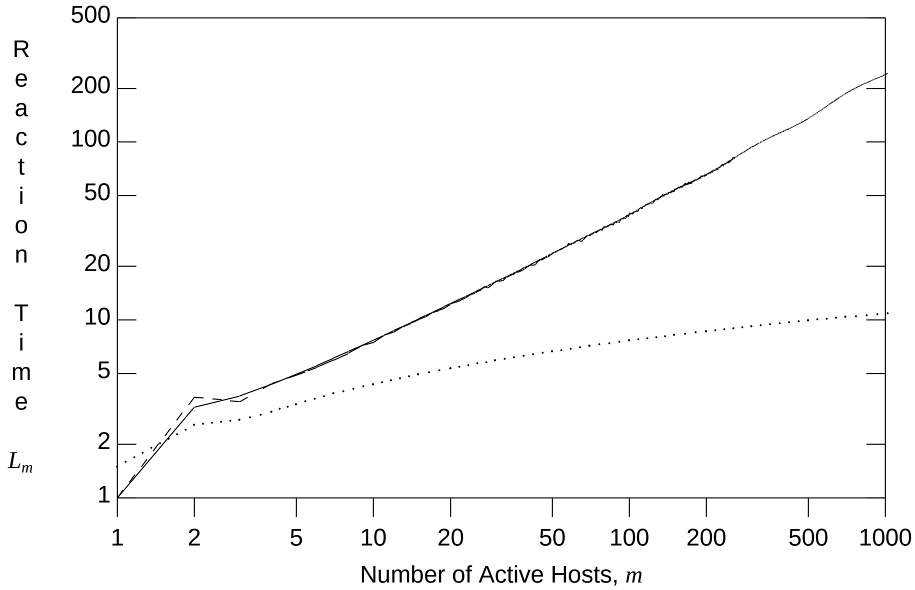

Figure 7: Average time to the first successful transmission after the ‘‘Big Bang’’, starting from m active hosts. Solid line is from Eq. (5), which is approximate because of an independence assumption. Dashed line is the sample mean obtained by Monte Carlo simulation of the exact system. Dotted line is from Eq. (6), which is the equivalent result for the Binary Logarithmic Arbitration Method that will be introduced in section III.

II.b. BEB is Supposed to be Stable for up to 1024 Hosts, but Isn't

Another aspect of Figure 6 worth noting is the convergence to steady-state after the 1000th slot time, as the possibility that the collision counter has ‘‘wrapped around’’ after 16 unsuccessful attempts becomes more significant. In particular, the pattern of damped oscillations has completely decayed to zero in less than 1 second, at which point the steady-state probability that the given host transmits in each backoff slot is approximately[10] 1/225. This result means that Ethernet using BEB will become *bistable* if

---

[10] We can calculate its exact value quite easily using the following argument. The first attempt for a new packet takes exactly 1 slot, the second attempt is equally likely to take 1 or 2 slots, and so on, so the average time until we hit an excessive collision error at the end of the 16th attempt is:

$$1 + \frac{1+2}{2} + \frac{1+2+3+4}{4} \cdots + 6 \times \frac{1+2+\cdots+1024}{1024} = 1 + \frac{3}{2} + \frac{5}{2} + \frac{9}{2} + \cdots + \frac{513}{2} + 6 \times \frac{1025}{2} = \frac{13 + 7 \times 1024}{2} \,.$$

Thus, recognizing that a given host makes 16 transmission attempts over such a ‘‘cycle’’, we see that in steady state its contribution to the channel traffic is $(13 + 7 \times 1024)/32 = 1/224.4$ transmission attempts per slot time. It is also the number of hosts in a single collision domain is significantly larger than 225, and in particular that anything approaching 1024 hosts will be problematic. Thus, if a situation ever arose in such a system where most of the hosts were trying to transmit packets at the same time, there is a chance that the system could enter the ‘‘steady state’’ operating mode, where each host independently cycles around its backoff/retry loop, generating an average of 1 packet every 225 slot times. If the number of hosts is much larger than 225, then almost every attempt will end in another collision and the system will remain in this degraded operating mode for a very long time. Note that bistable behaviour does not necessarily mean that the system, when started from a ‘‘good’’ state (like all hosts quiet), will not work at all. It simply means that eventually it will move from its ‘‘good’’ operating mode to its ‘‘bad’’ one, and possibly back again. The time spent in the ‘‘good’’ operating mode depends on such factors as average load on the system and detailed topology and traffic information, which is beyond the scope of this paper. However, it is important to note that the operation of the system during the ‘‘bad’’ operating mode does *not* depend on detailed traffic patterns: as long as each host’s transmit queue remains non-empty, its future transmission times are completely determined by the MAC layer protocol. And in particular, since BEB is a discrete time algorithm, synchronizing the transmissions by different hosts to some integer multiple of a slot time beyond end-of-carrier, even topology is of minor importance in an overload situation.

Figure 8 illustrates the instability problem with large numbers of hosts. In this Figure, Armyros [5] has used simulation to reproduce the measurement experiments reported by Boggs, et al. [6] on an overloaded Ethernet, so that configurations with many hundreds of hosts could be investigated. What is most striking about this Figure is the dramatic qualitative change in the observed behaviour when we reach hundreds of hosts. (Please note the use of a logarithmic *x*-axis for most of the graphs.) In particular, most packets suffer an excessive collision error, and the average value of the collision counter for those packets which *are* successfully transmitted is approaching its maximum value. Note also how the utilization is falling as the number of active hosts increases, especially with short packets.

The reversal of the effects of varying the packet size between *Ethernet Throughput* and *Percentage of Dropped Frames* can be explained as follows. First, since even the occasional successful transmission of a large packet will contribute significantly to the utilization in bits/sec., utilization is an increasing function of packet size. However, since each of those successful packets occupies the channel for such a long time, many hosts (even those waiting for substantially different backoff timeouts to expire) can enter the deferring state during one of those transmissions. Because of the 1-persistent transmission rule, all of those hosts will collide with each other at the next end-of-carrier event. Thus, the collision rate is also an increasing function of packet size.

II.c. But Exponential Backoff Causes Huge Delays for Some Packets, even During Light Load

Even though BEB acts *globally* like a linear search for Q, one must not forget that *locally,* i.e., from the perspective of an individual host, the backoff delays after each collision are increasing exponentially. As a result, the access time distribution has a very long ‘‘tail’’, i.e., a small fraction of the packets experience delays that are extremely large compared to the mean. As we already discussed in section I.c. in the context of delay variability, the fact that a few packets suffer large delays is important because it interferes with interactive response times and confuses higher-layer protocol adaptive timeout/congestion control algorithms. In other words, if the time constants in higher level protocols are measured in seconds, then even if only one packet from among the thousands transmitted over a second is *not* delivered in a timely fashion, the user level grade of service may be significantly affected.

---

easy to see that the steady-state solution to Eq. (4) is $P_1(\infty) = P_2(\infty) = \cdots = P_{15}(\infty)$, and hence that we can deduce nothing about the current value of a host’s collision counter from the fact that it chose to transmit now.