THE BEHAVIOR OF ETHERNET-LIKE

COMPUTER COMMUNICATIONS NETWORKS

Guy T. Almes and Edward D. Lazowska
Department of Computer Science, FR-35
University of Washington
Seattle, WA 98195

ABSTRACT

Considering the widespread influence of Ethernet, a surprising amount of confusion exists concerning various important aspects of its design. Our objective in writing this paper is to spare future designers of local area networks the searching and speculation in which we were forced to engage.

We begin by describing the policies common to Ethernet-like systems and by using an analytic model to study their behavior. We then precisely describe the mechanisms used in Ethernet itself, exploring its detailed behavior by means of a simulation model. Results from the two models, and particularly from their comparison, provide insight into the nature of low-level protocols in local area broadcast networks.

Key Words and Phrases: broadcast communication, local area computer networks, packet switching, distributed computing systems, supervisory system design and evaluation.

CR Categories: 3.81, 4.3, 8.1.

1. Introduction

The last five years has seen rapidly growing interest in a class of computer systems known as local area broadcast networks. The seminal work on such systems, performed under the auspices of the Aloha [1] and DCS [6] projects, was followed by the development of Ethernet at the Xerox Palo Alto Research Center [11]. Ethernet, because of its simplicity, robustness, and low cost, has spawned a significant number of similar computer communications networks [2, 3, 5, 7, 12], to which we refer collectively as the Ethernet-like networks.

Considering the widespread influence of Ethernet, a surprising amount of confusion exists concerning various important aspects of its design:

- Precisely what is the low-level protocol of Ethernet?
- How does the performance of this mechanism compare to that of the easily-analyzed policy that is the basis for the Ethernet-like networks?
- How does heavy loading affect Ethernet's throughput?
- Can Ethernet satisfy the requirements of real-time applications such as voice transmission?
- How does Ethernet's performance with a variety of packet sizes compare to its performance with a fixed packet size, if the average packet size is the same in each case?
- How would anticipated improvements in technology, e.g., the use of fiber optics, affect Ethernet's performance?

In considering the design of yet another Ethernet-like system, we had to go to considerable lengths to obtain the answers to questions such as these. Our objective in writing this paper is that future designers will be better informed than we.

This work was supported in part by the National Science Foundation under Grant No. MCS78-04347. Computing equipment was provided in part by the National Science Foundation under Grant No. MCS78-09463.

In Section 2, we use various analytic techniques to study an idealized network that may be viewed as the conceptual model from which each of the Ethernet-like systems (including Ethernet itself) is derived. In Section 3, we describe in detail certain of the mechanisms used in Ethernet itself, and investigate the performance of Ethernet using a simulation model. Results from the two models, and particularly from their comparison, answer the questions listed above and provide new insight into the nature of low-level protocols in local area broadcast networks.

2. Analysis of Ethernet-Like Networks

In this section, we consider an idealized network that provides some number of stations with time-division multiple access to a broadcast channel (the ether), governed by an adaptive distributed control policy. The basic unit of information transfer is the variable-length packet.

Network time can be divided into three categories: idle intervals, during which no stations desire to use the ether, transmission intervals, during which a single station is transmitting a packet, and contention intervals, the remainder of the time, during which several stations are trying to acquire the ether in order to transmit a packet.

The key to the behavior of the network is its adaptive distributed control policy: the means by which the ether is acquired and contention is resolved. If a station hears no traffic on the network, that station can begin transmission of a packet. Because of propagation delays, several stations may, in fact, begin transmissions simultaneously, resulting in a collision. A collision is detectable within the round-trip propagation delay of the network.

Stations involved in a collision must attempt to retransmit their packets at a later time. A principal objective of the control policy is stability: throughput must be a non-decreasing function of offered load. The original Aloha Network did not provide stability; in Ethernet-like systems, it is achieved by the following policy, which we suggest provides a useful and precise test for membership in the Ethernet-like class:

Although Ethernet is asynchronous, for analytic convenience we make the simplifying assumption that time is divided into slots of length equal to the round-trip propagation delay. Consider a slot, during which some number Q (> 0) of stations desire to transmit a packet; we refer to Q as the instantaneous load on the ether. If no station transmits during that slot, the slot is wasted. If exactly one station transmits, that station acquires the ether and continues transmitting until it has finished sending its packet. If more than one station transmits, a collision occurs and the slot is wasted. The Ethernet control policy attempts to maximize the probability that exactly one station transmits during a slot by independently allowing each station to transmit with probability 1/Q when Q stations desire to use the ether. The result is a binomial distribution of transmitting stations, with mean equal to 1.

This policy is optimal among all symmetric control policies, i.e. those in which all stations behave the same. Kleinrock and Yemini [10] note that assymmetric policies can, in fact, do better. The property that the probability of transmission decreases with increasing instantaneous load is one significant departure from the oft-analyzed Aloha Network control policy. Kleinrock and Lam [8] have shown that control policies without this property inevitably suffer from some combination of delay (which increases as the transmission probability decreases) and instability (which increases as the transmission probability increases).

We begin by summarizing the analysis of Metcalfe and Boggs [11], to which the reader may refer for details. We consider the probability that the ether is acquired by some station during a slot of contention. If Q stations desire to use the ether, then this probability equals:

$$A = (1 - \frac{1}{Q})^{Q-1} \qquad (1)$$

As the number of contending stations increases, this acquisition probability asymptotically approaches 1/e.

Next, we consider the number of slots devoted to contention prior to the acquisition of the ether by some station, under a specific instantaneous load. The probability that the ether is acquired on exactly the i-th slot is equal to $A(1-A)^i$. The mean number of slots devoted to contention is thus:

$$Z = \sum_{i=0}^{\infty} iA(1-A)^i = \frac{1-A}{A} \qquad (2)$$

Since the acquisition probability is asymptotically equal to 1/e, the mean number of slots devoted to contention is bounded by e-1.

We define the <u>instantaneous throughput efficiency</u> of the network to be the ratio of the proportion of time the network is successfully carrying packets (transmission intervals) to the proportion of time the network is busy (transmission intervals plus contention intervals), when the instantaneous load, Q, is artificially held constant. By definition there will be no idle intervals (although there will be portions of contention intervals during which no data is being transmitted), so instantaneous throughput efficiency is expressed by:

$$E = \frac{\frac{P}{C}}{\frac{P}{C} + SZ} \qquad (3)$$

where P is the average packet size in bits, C is the network carrying capacity in bits per second (bps) (thus P/C is the packet transmission time in seconds), S is the slot time in seconds, and Z is the mean number of slots devoted to contention.

Since the mean number of slots devoted to contention increases with instantaneous load, the instantaneous throughput efficiency of the network decreases with increasing instantaneous load. Since the mean number of slots devoted to contention is independent of the average packet size, the instantaneous throughput efficiency of the network increases with increasing average packet size.

The inherent stability of the adaptive distributed control policy can be demonstrated by calculating the asymptotic instantaneous throughput efficiency of the network as instantaneous load increases and average packet size decreases. As Q increases, A approaches 1/e. Let the packet size be the minimum feasible: a packet whose transmission time equals the slot time. Then:

$$E = \frac{1}{1 + \frac{1-A}{A}} = \frac{1}{1 + (e-1)} = \frac{1}{e} \qquad (4)$$

In other words, under heavy loads the throughput of the network will be at least 1/e times the network carrying capacity. For large average packet sizes, however, the <u>asymptotic throughput efficiency</u> may be considerably greater than 1/e.

Table 2-1 displays instantaneous throughput efficiency for various instantaneous loads and average packet sizes. We assume network characteristics typical of Ethernet-like networks: C = 3 Mbps, and S = 10 usec.[1]

	Packet Size, bits		
Q	256	512	2048
1	1.000	1.000	1.000
2	0.895	0.945	0.986
3	0.872	0.932	0.982
4	0.861	0.926	0.980
5	0.855	0.922	0.979
10	0.844	0.915	0.977
20	0.838	0.912	0.976
50	0.835	0.910	0.976
100	0.833	0.909	0.976

Table 2-1: Instantaneous Throughput Efficiency

[1] S = 1000 m. (network length) x 2 (round trip delay) / 200 m. per usec. (propagation rate)

At this point, we depart from the analysis of Metcalfe and Boggs. We note that instantaneous throughput efficiency is not an especially meaningful performance measure because of the artificial imposition of constant Q. Suppose instead that the network is subjected to an average load, ρ, measured as a proportion of the network's carrying capacity (in contrast to the instantaneous load, Q, which denotes the number of stations desiring the ether at a particular instant).2 Then the network will spend some proportion of time at each of a number of instantaneous loads, with corresponding instantaneous throughput efficiencies. Suppose further that this load comprises packets of average length P, i.e. stations are submitting new packets at an average rate of $\rho C/P$ per second. Using these values as input to a Markov model (see Appendix I) allows us to answer questions of the following sort:

- What proportion of network carrying capacity is devoted to contention resolution?
- What is the throughput efficiency of the network: the ratio of the proportion of time the network is successfully carrying packets to the proportion of time the network is busy?
- What is the average response time of the network: the average length of the interval between a station's desire to use the ether and the successful transmission of that station's packet, for packets of length P?
- What is the perceived efficiency of the network: the ratio of the theoretical transmission time for a packet of length P to the average response time seen by a station transmitting packets of that size?

Figure 2-1 illustrates the proportion of time devoted to transmission, contention and idle intervals for various average loads and for average packet sizes of 256 and 2048 bits. As average load increases, the proportion of time devoted to transmission keeps pace until it reaches the asymptotic throughput efficiency for the appropriate average packet size. For average loads greater than this value, the remainder of network capacity is devoted to contention. For average loads less than this value, the proportion of time devoted to contention rapidly decreases to a negligible value. The observed behavior of an existing Ethernet under an artificial load is consistent with this analysis in two important respects: the value of asymptotic throughput efficiency, and the average load beyond which a

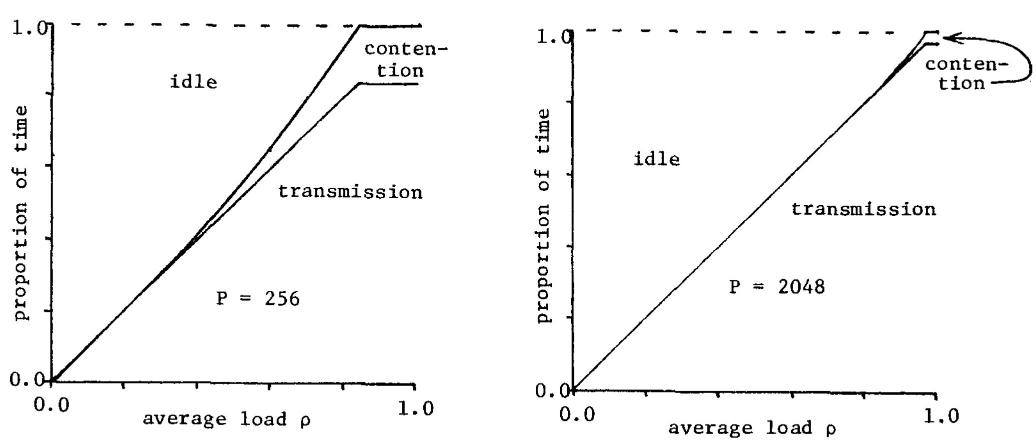

Figure 2-1: Time Transmitting, Contending, Idle

2Measurements of an existing Ethernet under normal operation [13] indicate that it is reasonable to speak in terms of "average load": a relatively low coefficient of variation of 1.4 in the time between packets was observed, as well as a surprisingly small difference in the maximum average utilization in any one hour, one minute, and one second.

noticeable proportion of time is devoted to contention, each as a function of average packet size [13].

Average response times for average packet sizes of 256 and 2048 bits are shown in Figure 2-2. As average load increases, an interval of negligible response time degradation is followed by a knee beyond which response times increase sharply. The average response time is asymptotically infinite for average loads equal to the asymptotic throughput efficiency. We note that average

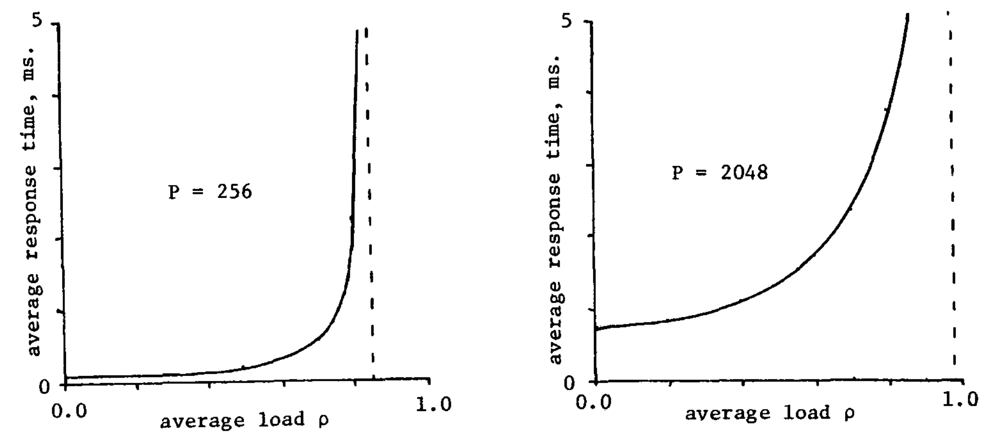

Figure 2-2: Average Response Times

response times of less than 1 msec. are achieved for average loads of up to 0.75 when the average packet size is 256 bits; the higher response times for a 2048-bit average packet size are due to the longer packet transmission time.

Average response time is a useful performance measure, but it is deficient in at least two respects. First, it is generally recognized that quantiles of response times are significantly more meaningful. In order to investigate the suitability of the network for a particular real-time application, for example, it might be necessary to know the average load beneath which more than 95% of all 512-bit packets experience a response time less than 25 msec. We defer consideration of the distribution of response times to the next section. Second, the shapes of the response time curves in Figure 2-2 cannot be used directly to compare the behavior of network policies for various average packet sizes, since the theoretical transmission time for a packet varies with packet size. The transmission time for a 256-bit packet is 85 usec. in a network with a carrying capacity of 3 Mbps; for a 2048-bit packet it is 683 usec. One approach to comparing network behavior with different average packet sizes is to consider <u>normalized</u> <u>response</u> <u>time</u>: mean response time scaled by packet length. In Figure 2-3, we display normalized response times for average packet sizes of 256, 512 and 2048 bits. We normalize to the scale of 512-bit average packet size: response times for the 256-bit average packet size are multiplied by 2; response times for the 2048-bit average packet size are divided by 4.

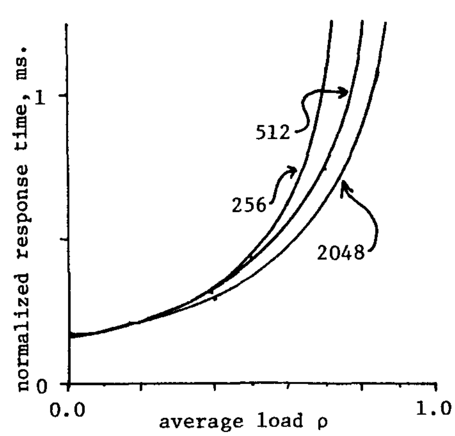

Figure 2-3: Normalized Average Response Times

Figure 2-3 makes it clear that network performance is in fact improved for larger average packet sizes. (Of course, this statement assumes that packets are fully utilized.)

Perceived efficiency is perhaps an even more informative measure. Figure 2-4 illustrates perceived efficiency as a function of average load for average packet sizes of 256 and 2048 bits. As an example, a perceived efficiency of 0.75 would result if the average response time for packets of 256 bits were 113 usec. Somewhat surprisingly, we note that perceived efficiency decreases linearly with increasing average load, reaching zero for an average load equal to the asymptotic throughput efficiency for the appropriate average packet size. At average loads greater than this value, the network is saturated. In other words, stations are submitting packets at a rate greater than the network's ability to carry them, given the proportion of network capacity that will be devoted to contention at that average packet size. This linear behavior means, for example, that an average response time equal to twice the theoretical packet transmission time can be achieved at an average load equal to half the asymptotic throughput efficiency for the appropriate average packet size.

For a specific average packet size, it is useful to define <u>relative load</u> to be the ratio of the average load, ρ , to the asymptotic throughput efficiency for that packet size. Relative load normalizes with respect to average packet size by factoring out the proportion of network carrying capacity that will be devoted to contention at saturation. For a specific relative load, perceived efficiency is independent of average packet size; in fact, perceived efficiency is equal to 1 minus the relative load. This observation provides an extremely succinct characterization of many of the important properties of Ethernet-like systems.

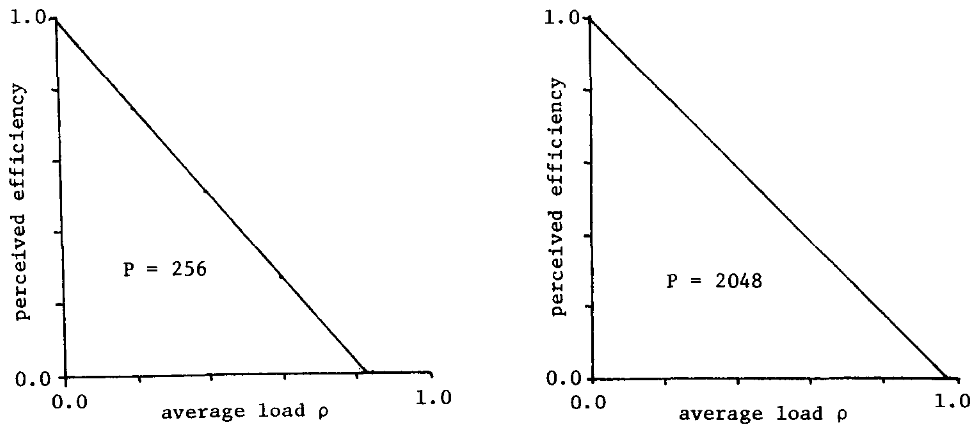

Figure 2-4: Perceived Efficiency vs. Average Load

To summarize, consider a simple example. Suppose that the network is operating with an average packet size of 512 bits, and that the average load is 0.4. We calculate that the asymptotic throughput efficiency for this average packet size is 0.91, so the relative load is 0.4/0.91 = 0.44 and the perceived efficiency will be 1-0.44 = 0.56. Since perceived efficiency is the ratio of theoretical packet transmission time to average response time for packets of the average size, average response time for 512-bit packets will be 171/0.56 = 305 usec.

Finally, we investigate the sensitivity of network performance to two design parameters: the slot time and the carrying capacity.

The slot time is equal to twice the network length divided by the propagation rate. If the network length were multiplied by some factor, the slot time would be multiplied by that same factor. The impact of increasing the slot time is to increase the length of contention intervals, degrading instantaneous throughput efficiencies and thus all performance measures. The extent of this degradation can be determined from data already presented. From Equation 3 we note that for constant network carrying capacity C, multiplying the slot time S by some factor has the same effect on instantaneous throughput efficiency as does dividing the average packet size P by the same factor. In other words, for an 8 km. network with a 2048-bit average packet size, the proportion of time spent in transmission, contention and idle intervals (as a function of average load) and the perceived efficiency (as a function of either average load or relative load) will be identical to the corresponding measures for a 1 km. network with a 256-bit average packet size. Only mean response times will differ; they will be greater by a factor of 8 for the 8 km. network, in which the average packet size is 8 times as large. In summary, the policies described in this section are sensitive to the slot time, and are applicable only over a restricted range of network lengths. In a specific implementation a basic time unit somewhat greater than the slot time may be selected for reasons of convenience, e.g., the granularity of an existing clock. Our analysis suggests that the choice of basic time unit may have a significant effect on performance.

To put the roles of slot time and network carrying capacity in perspective, consider the effect of introducing fiber optic technology to replace the present coaxial cable technology. For a given network length, slot times can improve only slightly, since the propagation rate is limited by the speed of light, roughly half again as fast as that achieved by coax. Carrying capacity, on the other hand, can be expected to grow to at least 100 Mbps. Since instantaneous throughput efficiencies decrease with decreasing packet transmission times, large average packet sizes will be necessary if the full benefits of this increased capacity are to be realized.

The impact of an increase in carrying capacity can also be assessed from data already presented. From Equation 3 we note that for constant slot time S, multiplying the carrying capacity C by some factor has the same effect on instantaneous throughput efficiency as does dividing the average packet size P by the same factor. In other words, for a 24 Mbps network with a 2048-bit average packet size, the proportion of time spent in transmission, contention and idle intervals (as a function of average load), the average response time (as a function of average load) and the perceived efficiency (as a function of either average load or relative load) will be identical to the corresponding measures for a 3 Mbps network with a 256-bit average packet size. Of course, for a given average load the former network will be carrying 8 times the number of bits per second as the latter.

The objective of this section has been to describe the policies common to Ethernet-like computer communications networks, and to understand certain aspects of the behavior of this class of networks. The analysis applies to any network whose control policy closely approximates the optimal 1/Q policy, regardless of implementation details. Although the analysis has achieved its objectives, a number of issues remain to be investigated:

- It may be that the mechanisms employed in the various Ethernet-like networks are sufficiently far removed from the policies modelled here that the analysis is misleading in certain respects. (The close correspondence to Shoch and Hupp's preliminary measurements [13] would suggest otherwise.) Even should the analysis prove to be valid, it is unable to distinguish performance variations among the Ethernet-like networks. Presumably their subtle differences in design influence performance to some extent.

- Although mean response times and perceived efficiencies are meaningful performance measures, knowledge of the distribution of response times is necessary in order to assess the suitability of the network for real-time applications.

- Presumably the performance of the network is sensitive to the distribution of packet sizes. The analysis presented here draws no distinction between an interval during which each packet is exactly 512 bits and an interval during which 6/7 of the packets are exactly 256 bits and 1/7 are exactly 2048 bits.

In the next section, we use a detailed simulation model to investigate questions such as these.

3. Simulation of Ethernet

In the previous section we suggested that the essential properties of Ethernet-like systems were captured by the 1/Q control policy. We begin this section with a description of certain implementation details in the original Ethernet that illuminate its relationship to the 1/Q model.

Many aspects of Ethernet are well-known and adequately described by Metcalfe and Boggs (cf. [11], esp. Section 4). Other aspects, although perhaps not widely known, are not significant in understanding the behavior of the system. In this category, we include issues such as phase encoding and decoding, cyclic redundancy checking, and collision concensus reenforcement. However, we do want to probe more deeply the calculation of retransmission intervals. When a station initially desires to transmit a packet and finds the ether busy, it defers to the passing packet, and then immediately attempts to transmit its own packet. When a station experiences a collision, on the other hand, it first delays for some retransmission interval, then defers to any passing packet, and finally retries the transmission.[3] Retransmission intervals are drawn from a uniform distribution whose mean is set initially to some base value, doubled after each collision, and finally reset to the base value after a successful transmission. The actual method used to calculate these retransmission intervals is quite interesting, and may surprise the reader. The mean of the distribution is determined by a mask, initially zero. Whenever a collision occurs, this mask is shifted left one bit, and the low-order bit is set to one. The mask is then ANDed with the low-order eight bits of a clock within the station. The resulting value determines the number of 38.08 usec. clock ticks in the retransmission interval. If a shift of the 16-bit mask results in a carry out of the high-order bit, the transmission is aborted.

Several aspects of this implementation are worthy of note. Upon its first collision, a station waits for 0 or 1 38.08 usec. ticks; then for 0, 1, 2, or 3 ticks; then for 0, 1, ..., 7 ticks; etc. Each of these shifts corresponds to an upward revision of the station's estimate of Q. After its first collision, the station makes an implicit estimate of Q=2; the station then retransmits immediately with probability 1/2 and waits for 38.08 usec. with probability 1/2. One departure from the optimal 1/Q model is that 38.08 usec. is considerably larger than the 10 usec. slot time. A second is that Q is only estimated. The first collision tells us only that Q must be greater than 1; upon successive collisions, the mask is shifted, corresponding to a doubling of the estimate of Q. A third departure is the limit on backoff. Note that after eight collisions the estimate of Q stops growing; the maximum retransmission interval is about 10 msec. Eight more attempts are made, then the transmission is aborted. (The whole process may, of course, be repeated at the request of higher level software.)

[3] Thus, in the terminology of [9], the Ethernet control policy is <u>1-persistent</u>. One recent analysis of similar network control policies deals with the 0-persistent case [14].

Finally, we note the potential for multiple collisions immediately following the transmission of a large packet. A number of stations may have generated traffic during this period; still more stations may have come to the end of their retransmission intervals. All of these stations will transmit when the ether becomes free.

Ethernet's implementation of the 1/Q model can thus be seen to comprise two parts. The first is an estimation of the value of Q; the second is a backoff strategy that aims to have each station attempt retransmission at each slot time (actually 38.08 usec. here) with probability 1/Q.

We model the Ethernet implementation using a Simula 67 program which appears in Appendix II. Confidence intervals for our simulations were derived using the regenerative method [4], but are not reported here. Based upon these simulations we make several observations about Ethernet performance. The first concerns contention time for three fixed packet lengths: 256, 512, and 2048 bits. Consider the mean contention intervals shown in Figure 3-1.

In each case the contention intervals are very short at light loads; they grow dramatically at heavier loads, but are bounded as the load approaches the asymptotic throughput efficiency for the appropriate packet length. The maximum contention interval, about 40 usec. for 512-bit packets, corresponds to the (e-1)S of Section 2. The existence of this upper bound is the key to Ethernet's stability. As average load increases, idle intervals vanish and the ether alternates between intervals of contention and transmission. Since the mean contention interval has an upper bound, throughput has a lower bound. One deviation from our analysis of the 1/Q model is the dependence of this bound on the packet length. At light loads, long packet lengths have shorter contention intervals; at heavy loads, they have longer contention intervals. In no case, though, is stability threatened, and the ceiling of (e-1)*38.08 usec. (about 68 usec.) is never exceeded.

Normalized response times for packet lengths of 256, 512, and 2048 bits are shown in Figure 3-2. We observe that the curves are similar: they grow

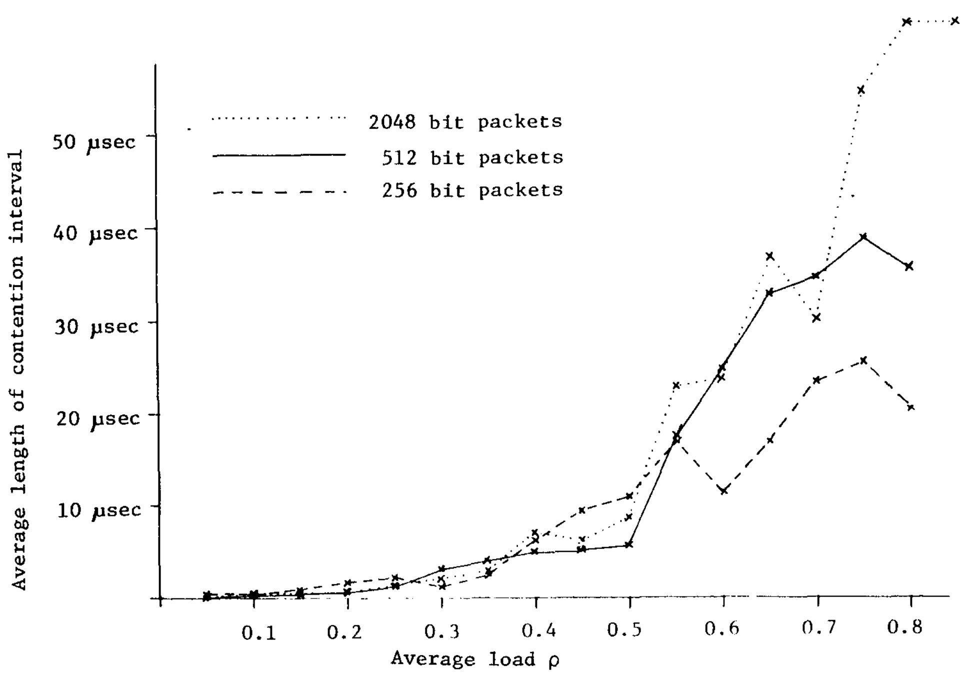

Figure 3-1: Contention Time for Three Lengths

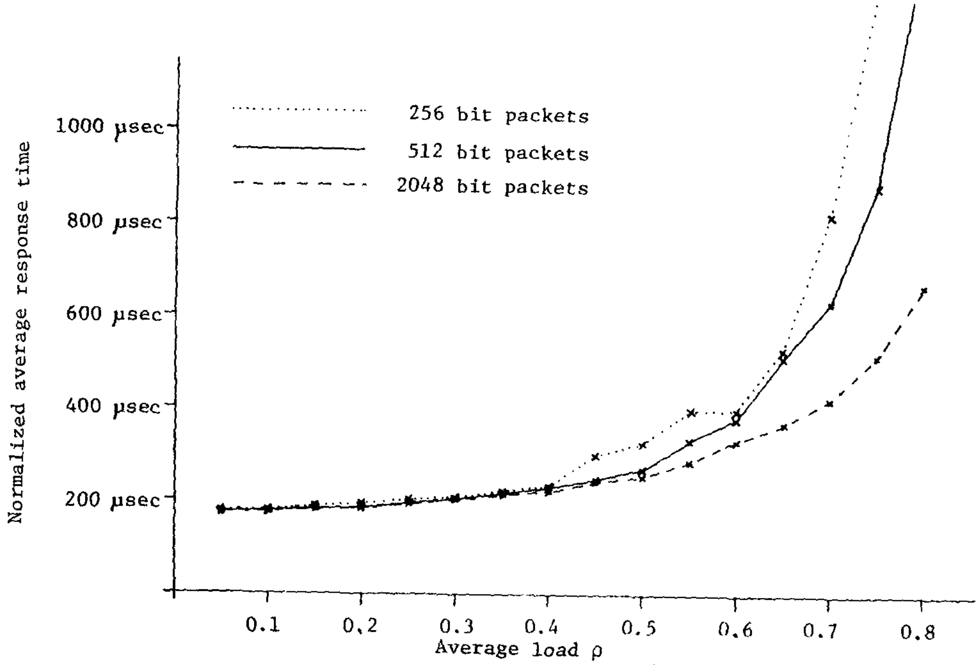

Figure 3-2: Response Time for Three Lengths

slowly at light load, then grow without bound as the average load approaches the appropriate asymptotic throughput efficiency. We also notice that small packets do have larger normalized response times than large packets. Qualitatively and quantitatively, these results are extremely close to those shown in Figure 2-3.

Let us now consider how performance under a bimodal distribution of packet lengths compares with performance under a fixed packet length.[4] To study this, a simulation was run in which each packet had a length of 256 bits with probability 6/7 and 2048 bits with probability 1/7. The average packet length was thus equal to the case in which fixed length 512-bit packets were used. In Figure 3-3, we see that mean response time was noticeably worse for the bimodal distribution. Intuitively we can regard this as due to an irregular load--the ether will have intervals of low utilization (due to a series of 256-bit packets) and intervals of severe contention (due to numerous arrivals during a 2048-bit transmission). Many networks are expected to support a variety of applications, with a corresponding variety of packet lengths. The results shown in Figure 3-3 suggest that these networks should be designed with considerable care.

In many real-time environments the average load due to a single application is not very high, but fast, consistent response is required. In these environments the mean response time is an inadequate performance measure. In Figure 3-4 we graph both the mean and the standard deviation of response times and observe that the standard deviation grows more rapidly than the mean.

This high variability in response times suggests that Ethernet's ability to satisfy stringent real-time constraints might degrade severely as load increases. However, when the mean response time is small relative to the time constraint, this high variability can be tolerated. Consider the use of Ethernet to provide a number of voice

4Shoch and Hupp report such a distribution [13]: the majority of the packets are short (corresponding to terminal activity), but the majority of the bits are shipped in large packets (corresponding to file transfer operations).

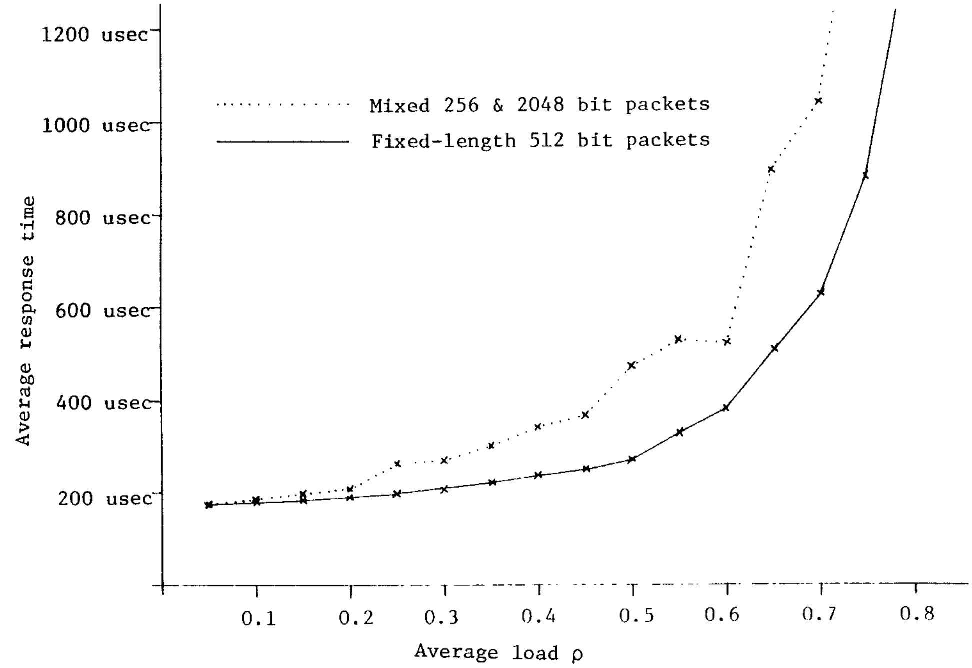

Figure 3-3: Response Time for Mixed Lengths

channels, each consisting of one 512-bit packet every 25 msec. with a 25 msec. response time requirement. (One such channel would utilize 0.68% of network carrying capacity.) Table 3-1 shows the response time achieved by 95% of all packets (the 95th quantile) for various average loads in an Ethernet devoted entirely to this application. Our conclusion is that Ethernet may be appropriate when such "soft" real-time constraints are permissible.

Let us now consider the source of the large standard deviation in response times. Is it an artifact of the 1/Q model of Section 2, or of the particular implementation of this model in

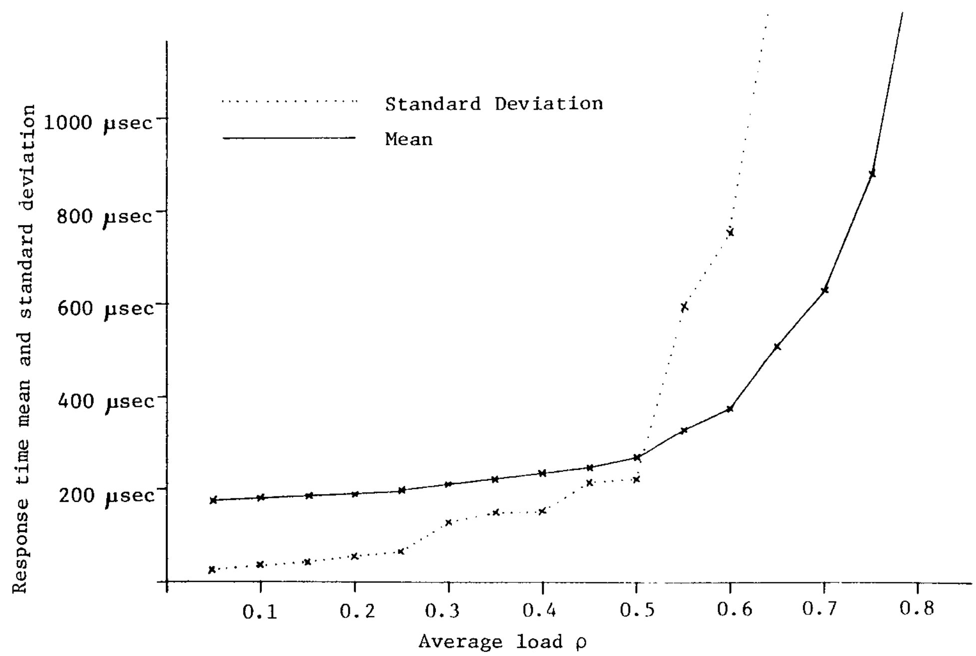

Figure 3-4: Response Times; Mean and Stand. Dev.

Load, ρ	95th Quantile msec.
0.1	.075
0.2	.140
0.3	.165
0.4	.300
0.5	.435
0.6	.825
0.7	7.8% > 1.0 msec.
0.8	15.9% > 1.0 msec.

Table 3-1: 95th Quantile of Response Times vs.

Ethernet? Intuitively, we can find some reason to suspect the implementation. Ethernet achieves stability by means of a backoff algorithm executed by stations that fail in their attempts to send packets. This means that a station on its third or fourth attempt will usually wait longer before retransmitting than a station that has only recently decided to send a packet.[5]

In order to measure this effect, we simulated two variations of the Ethernet backoff scheme. The first is called Pseudo-1/Q. Under it, after each collision, the simulated station retransmits immediately with probability 1/Q. This perfect knowledge of Q, though easy to simulate, cannot be achieved in practice. The second, called Short Backoff, is similar to Ethernet, except that (a) 15 usec. ticks are used instead of 38.08 usec. ticks and (b) only four bits of the clock are used instead of eight. The 15 usec. tick more closely approximates a slot time; the four-bit clock stops doubling the mean retransmission time sooner. In Figures 3-5 and 3-6, the means and standard deviations of response times for Ethernet and these two variants are compared. In Figure 3-5 we see that both of our new schemes have significantly better means than the standard backoff scheme, that the Short Backoff scheme comes very close to the Pseudo-1/Q scheme, and that all three schemes are quite close below average loads of 0.3. In Figure 3-6 we see that our new schemes show even greater improvement in standard deviation than in mean, that the Short Backoff scheme is not as close to Pseudo-1/Q as before, and that all three schemes are close only below average loads of 0.25.

We now suggest a rationale for the improvement

[5] A simple analogy could be made to a time-sharing scheduler where requests for service are stacked instead of queued: if service times have relatively low variability, then the mean response time will be acceptable, but the standard deviation of response times may not be.

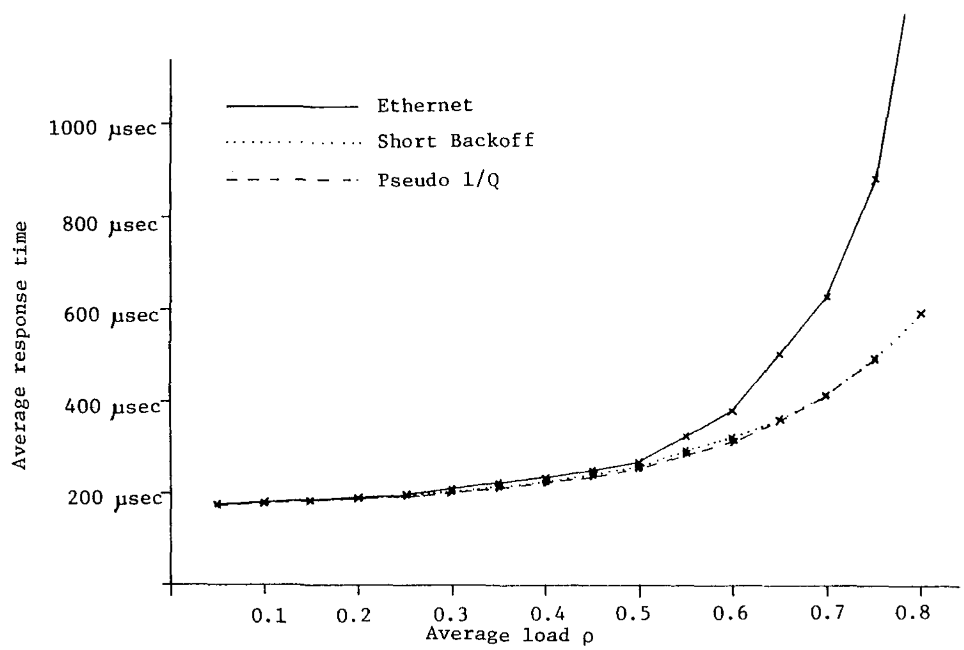

Figure 3-5: Response Time Mean for Variants

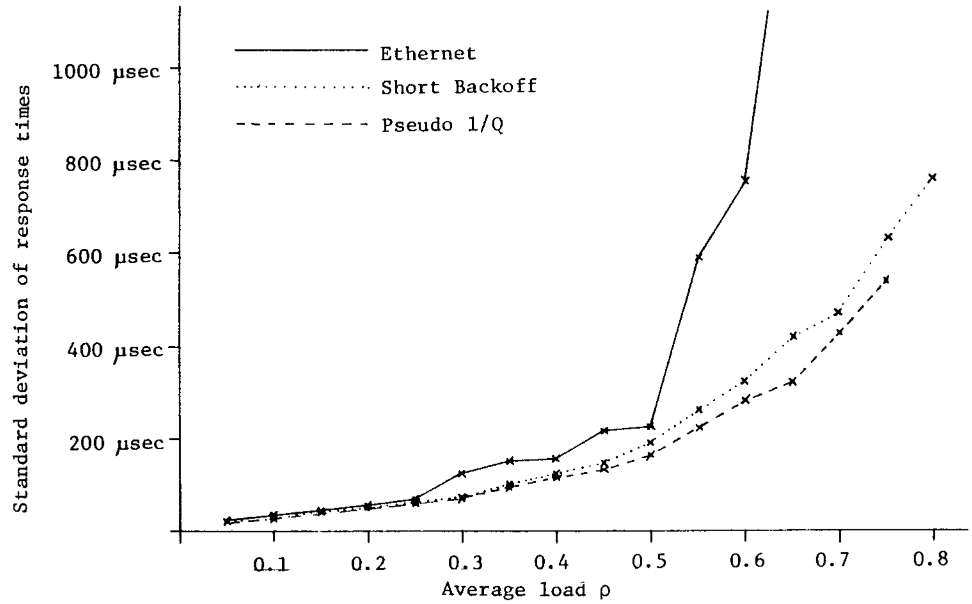

Figure 3-6: Response Time Stand. Dev. for Variants

of the two new schemes over Ethernet. First, the reduction of backoff grain from 38.08 usec. to 15 usec. seems to have helped; this explains much of the improvement in mean response time. Second, the Pseudo-1/Q scheme does not discriminate against stations that are making a third or fourth attempt, as does the Ethernet policy; this shows up especially in the superiority of Pseudo-1/Q with respect to standard deviation of response time. Finally we note that the Short Backoff scheme discriminates, but less than Ethernet; in effect it never estimates Q at more than about 16, while Ethernet continues to double its estimate. In our simulations, in fact, we never observed values of Q greater than 16. This suggests, however, that if Q ever did exceed the maximum value estimated by Short Backoff, then stability might be compromised.

4. Conclusions

In this paper we have used an analytic model to study the behavior of Ethernet-like computer communications networks and a simulation model to study the performance of Ethernet itself. Among our more interesting observations are the following:

- The behavior of Ethernet is close to that of the optimal 1/Q control policy. This observation has two significant implications:

  The results of simple analytic models are applicable (although they certainly do not answer all of the interesting questions), and

  vastly improved implementations of the 1/Q policy are not likely to be developed.

- Ethernet and other networks adequately grounded in the 1/Q model are stable: throughput is a non-decreasing function of load.

- The fact that perceived efficiency is equal to 1 minus relative load provides a succinct characterization of many of the important properties of Ethernet-like systems.

- Due to Ethernet's implementation of the 1/Q model, it has considerable variance in response times. This variance does not, however, seem to make it unsuitable for "soft real-time" applications at moderate average loads.

- The performance of these systems is quite sensitive to the distribution of packet sizes.

- System performance is also quite sensitive to the slot length. The existing control mechanism will not be effective if network length is increased substantially. Choosing a basic retransmission interval for reasons of convenience, e.g., the granularity of an existing clock, may significantly impact performance.

- Higher bandwidth technologies, e.g., fiber optics, will provide greatest benefit for applications that can use large packet lengths.
- When designing or evaluating a control mechanism based on the 1/Q model, it may be useful to decompose it into two parts:

  A mechanism for estimating Q from information available to a station, and

  a backoff mechanism that closely approximates the 1/Q model for the estimated value of Q.

We would also like to suggest one important area for future research. Ethernet seems to obtain its stability at the cost of exhibiting a degree of "last come first served" scheduling behavior. Is this necessary, or can implementations of the 1/Q model avoid this behavior, and thus give more consistent response, while retaining Ethernet's simplicity, stability and good mean response?

Finally, we would like to emphasize the close relation between the particular performance issues raised here and the needs of system designers. If simplicity, stability and throughput are important, then the original Ethernet control policy looks surprisingly good. If tight real-time response is needed, then a modified policy with more consistent response may be necessary. If a physically larger network or a very high bandwidth medium is anticipated, give careful thought to ways of reducing mean contention time or, if compatible with application needs, increasing average packet length.

Acknowledgements

We should like to thank David Boggs and Ed McCreight of the Xerox Palo Alto Research Center for their help in deciphering the details of Ethernet, and to credit Mariano Landicho for his support of the Simula 67 implementation at the University of Washington. Forest Baskett, Peter Denning, John Shoch, Fouad Tobagi, and the referees provided a number of helpful suggestions.

THE APPENDICES

I. The Analytic Model

The analytic model is extremely straightforward. The state of the system is denoted by Q, the instantaneous load. Let the system be in some state $q > 0$. The system moves to state $q+1$ at a rate equal to the rate at which packets arrive when the instantaneous load is q. We assume that this rate, which by definition has an average value of $\rho C/P$, is independent of q (an <u>infinite source</u> assumption). The system moves to state $q-1$ at a rate equal to the rate at which packets are delivered when the instantaneous load is q. We assume that this rate is equal to the network carrying capacity in packets per second, C/P, multiplied by the instantaneous throughput efficiency when the instantaneous load is q, calculated from Equation 4. (This use of a low-level performance measure (in this case, instantaneous throughput efficiency) as input to a high-level model is a typical application of the principle of <u>decomposability</u>.) When the system is in state 0 it is idle and packet deliveries cannot occur; packet arrivals occur at rate $\rho C/P$.

Solution of this model yields the proportion of time the system spends in each state, i.e., at each instantaneous load. The proportion of time devoted to contention is equal to the proportion of time the instantaneous load is greater than zero, minus ρ. The average response time can be determined from Little's equation by taking the quotient of the average instantaneous load and the average packet arrival rate, $\rho C/P$.

We note that one might plausibly argue in favor of either an infinite source model, as used here, or a finite source model, in which the packet arrival rate depends on the system state. As a practical matter, however, the choice is irrelevant; their predictions are indistinguishable for systems with large numbers of stations.

II. The Simulation Program

```
Simulation
  begin
  comment Lengths in meters, times in usec,
             information in bits;

  integer PacketLength, Seed;
  real SlotTime, HalfSlotTime, Capacity,
             CPrime, Rho, Lambda;
  ref(Ether) Net;


Link class Packet(Owner);
  ref(Node) Owner;
  begin
  real TNaught, TFinal, TWakeup, Source;
  boolean Collided;
  TNaught := Time;
  TWakeup := TFinal :=
        TNaught+Owner.PacketSize/Capacity;
  Source := Owner.Locus;
  Net.Assert(this Packet);
  Hold(TWakeup-Time);
  Collided := TWakeup  = TFinal;
  TFinal := TWakeup;
  activate new Killer(this Packet)
        delay HalfSlotTime
  end Packet;


Process class Killer(Pack);
  ref(Packet) Pack;
  begin
  Pack.Out
  end Killer;


Head class Ether;
  begin

  boolean procedure  Busy(Locus);
    real Locus;
    begin
    ref(Packet) x;
    Busy := false;
    x :- First;
    while x =/= none do
      inspect x do
        begin
        real TCrit;
        comment Pulse at Locus leaves x;
        TCrit :=
            Time-abs(Locus-Source)/CPrime;
        if TNaught<TCrit and
            TCrit<TWakeup+1.5/Capacity then
                          Busy := true;
        x :- Suc
        end
    end Busy;

 procedure Assert(Pack);
   ref(Packet) Pack;
   begin
   ref(Packet) x;
   x :- First;
   while x =/= none do
     inspect x do
       begin
       real TCrit1;
       comment Time when my pulse hits x;
       real TCrit2;
       comment Time when x pulse hits me;
       TCrit1 := Time +
         abs(Pack.Source-Source)/CPrime;
       if TCrit1<TWakeup then
         begin
         TWakeup := TCrit1;
         reactivate Owner at TCrit1
         end;
       TCrit2 := TNaught +
         abs(Pack.Source-Source)/CPrime;
       if Time<TCrit2 and
               TCrit2<Pack.TWakeup then
                     Pack.TWakeup := TCrit2;
       x :- Suc
       end;
   Pack.Into(this Ether)
   end Assert;
 end Ether;


Process class Node;
  begin
  integer PacketSize, Mask, NTries;
  real Locus;
  ref(Packet) CurrentP;

  boolean procedure Collision;
    if NTries<=16 then
      begin
      while Net.Busy(Locus) do Hold(0.001);
      CurrentP :- new Packet(this Node);
      Collision := CurrentP.Collided
      end Collision;

 NTries := 1;
 Mask := 0;
 PacketSize := PacketLength;
 Locus := Uniform(0,1000,Seed);

 while Collision and NTries<=16 do
   begin
   Mask := Mask*2+1;
   comment Shift Mask left, one filled;
   Hold( mod(RandInt(0,255,Seed),Mask+1)
             * 38.08 );
   comment i.e. Clock<8:0> AND Mask;
   NTries := NTries+1
   end;

 comment if CurrentP.Collided
      then Packet was Aborted;
 end Node;



 comment Initialization of main program;
 Net :- new Ether;
 CPrime := 200;
 HalfSlotTime := 1000/CPrime;
 SlotTime := 2*HalfSlotTime;
 Capacity := 3;
 Seed := 1;
 PacketLength := 512;
 Rho := 0.25;
 Lambda := Rho*Capacity / PacketLength;

 while true do
   begin
   activate new Node;
   Hold(NegExp(Lambda,Seed))
   end Arrival Loop
 end EtherNet
```

References

[1] Norman Abramson.
The Aloha System--Another alternative for computer communications.
In Proc. 1970 Fall Joint Computer Conference, pages 281-285. American Federation of Information Processing Societies, 1970.

[2] A. K. Agrawala, R. M. Bryant, and J. Agre.
Analysis of an Ethernet-like protocol.
In Proc. Computer Networking Symposium, pages 104-111. IEEE Computer Society and National Bureau of Standards, December, 1977.

[3] Robert Carpenter, Joseph Sokol, Jr., and Robert Rosenthal.
A microprocessor-based local network node.
In Proc. CompCon Fall 78, pages 104-109. IEEE Computer Society, 1978.

[4] M. A. Crane and A. J. Lemoine.
An introduction to the regenerative method for simulation analysis.
In Lecture Notes in Control and Information Sciences 4, . Springer-Verlag, 1977.

[5] James E. Donnelley and Jeffry W. Yeh.
Interaction between protocol levels in a prioritized CSMA broadcast network.
In Proc. Third Berkeley Workshop. Lawrence Radiation Laboratory, 1978.

[6] David Farber, Julian Feldman, Frank Heinrich, Martha Hopwood, Kenneth Larson, Donald Loomis, and Lawrence Rowe.
The distributed computing system.
In Proc. CompCon 73, pages 31-34. IEEE Computer Society, February, 1973.

[7] L. H. Gerhardstein, J. O. Schroeder, and A. J. Boland.
The Pacific Northwest Laboratory minicomputer network.
In Proc. Third Berkeley Workshop. Lawrence Radiation Laboratory, 1978.

[8] Leonard Kleinrock and Simon Lam.
Packet switching in a multiaccess broadcast channel: performance evaluation.
IEEE Trans. Communications COM-23(4):410-423, April, 1975.

[9] Leonard Kleinrock and Fouad Tobagi.
Packet switching in radio channels: Part I--Carrier sense multiple-access models and their throughput-delay characteristics.
IEEE Trans. Communications COM-23(12):1400-1416, December, 1975.

[10] Leonard Kleinrock and Y. Yemini.
An optimal adaptive scheme for multiple access broadcast communication.
In Proc. ICC 78, pages 7.2.1-7.2.5. IEEE, 1978.

[11] Robert Metcalfe and David Boggs.
Ethernet: Distributed packet switching for local computer networks.
Comm. of the Assoc. for Computing Machinery 19(7):395-404, July, 1976.

[12] R. H. Sherman, M. G. Gable, and G. McClure.
Concepts, Strategies for Local Data Network Architectures.
Data Communications 7(7):39-49, July, 1978.

[13] John Shoch and Jon Hupp.
Performance of an Ethernet local network--A preliminary report.
In Proc. Local Area Communications Network Symposium. National Bureau of Standards and The MITRE Corporation, Boston, May, 1979.

[14] Fouad Tobagi and Bruce Hunt.
Performance Analysis of Carrier Sense Multiple Access with Collision Detection.
Technical Report 173, Computer Science Laboratory, Stanford University, June, 1979.

This paper was presented at the Local Area Communications Network Symposium, Boston, May, 1979.