COMPUTER SYSTEMS LABORATORY

DEPARTMENTS OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
STANFORD UNIVERSITY · STANFORD, CA 94305

Packet-Voice Communication on an Ethernet Local Computer Network: an Experimental Study

Timothy A Gonsalves

Technical Report No. 230

February 1982

This work was supported in part by IBM under the Distributed Systems Contract # SEL 47-79. The Xerox Palo Alto Research Center, provided the experimental and some other facilities.

Packet-Voice Communication on an Ethernet Local Computer Network: an Experimental Study

Timothy A Gonsalves

Technical Report No. 230

February 1982

Computer Systems Laboratory
Departments of Electrical Engineering and Computer Science
Stanford University
Stanford, California 94305

Abstract

Local computer networks have been used successfully for data applications such as file transfers for several years. Recently, there have been several proposals for using these networks for voice applications. This paper describes a simple voice protocol for use on a packet-switching local network. This protocol is used in an experimental study of the feasibility of using a 3 Mbps experimental Ethernet network for packet-voice communications. The study shows that with appropriately choosen parameters the experimental Ethernet is capable of supporting about 40 simultaneous 64-Kbps voice conversations with acceptable quality. This corresponds to a utilization of 95% of the network capacity.

Key Words and Phrases: CSMA/CD, Ethernet, local computer network, measurement, packet-voice, voice communication.

CR Categories: C.2.5, C.4

1. Introduction

In the past few years, local computer networks for the interconnection of computers and shared resources within a small area such as a building or a campus have rapidly increased in popularity. These networks support applications such as file transfers and electronic mail between autonomous computers, the shared use of large computers and expensive peripherals, and distributed computation. Typically, these networks have bandwidths in the range 0.1 - 10 Mbps and span distances of 0.1 - 10 km.

Many architectures have been proposed for local networks, and several have been implemented The Ethernet [Metcalfe & Boggs, 76] was one of the earliest to be implemented Use of a 3 Mbps *experimental Ethernet* at several installations by a large community of users over several years and an experimental study of its performance under varying conditions have proved its merit for a variety of data transfer applications [Shoch, 79]. This has led to the introduction of a 10 Mbps Ethernet Ethernet, 80] as a commercial product.

Recently, interest has focused on the use of local area networks for voice and other real-time applications. A local computer network can be used to support voice communications (in addition to data traffic), thereby supplementing or eliminating the internal telephone exchanges (PBX's) currently used in offices and campuses. New levels of functionality can be achieved by the integration of digitized voice into data systems such as text editors and electronic mail systems ([Shoch, 80a]). In this paper we present the results of some measurements aimed at evaluating the feasibility of a 3 Mbps experimental Ethernet, located at the Xerox Palo Alto Research Center, for packet-voice applications. These measurements show that, within certain bounds, the network can be used to carry voice traffic successfully. These results can also be used to validate theoretical models which can then be used to predict the performance of other implementations of Ethernet-like architectures.

In the next section we describe the principles of the Ethernet architecture and relevant details of the implementation used in this work, and we review prior work in this area. Section 3 goes into details of the experimental procedures. Results are presented and discussed in section 4. Section 5 is a summary of the paper. In the rest of this paper, for brevity, *we use* the term *Ethernet* to refer to the experimental Ethernet described in section 2.2 since our interest focuses on it.

2. Background and Previous Work

Here we review the Ethernet local network architecture and describe relevant details of the implementation of the Ethcmet uscd for the experiments reported in this paper. We then summarize some of the results on mcasured pcrformancc of this Ethcrnct under artificially gcneratcd loads reported in [Shoch & Hupp, 80].

2.1 The Ethernet Architecture

The Ethernet **[Metcalfe & Boggs, 76]** is an example of a packet-switching local network using a single shared bus or channel for communication among several hosts. The nature of the channel is such that only one packet can be transmitted at a time and every packet traverses the entire channel. This requires that all the hosts observe some protocol to gain control of the channel in an orderly and fair manner.

The Ethernet uses an access protocol called *CSMA/CD*. *CSMA*, or *carrier sense multiple-access*, is one of the distributed schemes proposed to efficiently utilize a broadcast channel [Kleinrock & Tobagi, 75]. In the CSMA scheme, a host desiring to transmit a packet waits until the channel is idle, i.e. there is no carrier, and then starts. transmission. If no other hosts decide to transmit during the time taken for the first bit to propagate to the ends of the channel, the packet is successfully transmitted (assuming no errors due to noise). - However, due to the finite propagation delay, some other hosts may also sense the channel to be idle and decide to transmit. Thus several packets may collide. To ensure reliable transmission, acknowledgements must be generated by higher level protocols. Unacknowledged packets must be retransmitted after some time.

In order to improve performance, the Ethernet protocol incorporates *collision detection into the* basic CSMA scheme, hence the abbreviation CSMA/CD. To detect collisions, a transmitting host monitors the channel while transmitting and aborts transmission if there is a collision. It then jams the channel for a short period to ensure that all other transmitting hosts abort transmission also. Each host then schedules its packet for retransmission after a random interval choosen according to some *retransmission or backoff algorithm*. The randomness is essential to avoid repeated collisions between the same set of hosts. The retransmission algorithm can affect such characteristics as the stability of the network under high loads and the fairness to contending hosts. References to proposals and studies of many variations of the basic CSMA scheme can be found in **[Shoch, 80]**.

2.2 An Ethernet Implementation

We summarize the relevant details of the Ethernet local network used in our experiments. This network has been described in detail in **[Metcalfe & Boggs, 76]** and [Crane & Taft, 80]. The Ethernet used in our experiments has a channel about 550 metres long with baseband transmission at 2.94 Mbps. The end-to-end propagation delay. is thus about 2.75 μs.

The retransmission algorithm implemented in the Ethemct hosts (for the most part, Alto minicomputers [Thacker *et. al.*, 82]) is an approximation to the ideal *binary exponential* backoff algorithm. In the binary exponential backoff algorithm, the mean retransmission interval is doubled with each successive collision of the -same packet. Thus, there can be an arbitrarily large delay before a packet is transmitted, even if the network is not heavily loaded.

To avoid this problem, the Ethcmet hosts use a *truncated binary exponential backoff algorithm*. Each host maintains an estimate of the number of hosts attempting to gain control of the network in its *load register*. This is initialized to zero when a packet is first scheduled for transmission. On each successive collision the estimated load is doubled by shifting the load register 1 bit left and setting the low-order bit to 1. This estimated load is used to determine the retransmission interval as follows. A random number, X, is generated by AND'ing the contents of the load register with the low-order 8 bits of the processor clock. Thus X is approximately uniformly distributed between 0 and 2^n-1, where n is the number of successive collisions, and has a maximum value of 255, which is the maximum number of hosts on the network. The retransmission interval is then choosen to be X time units. The time unit should be no less than the round-trip end-to-end propagation delay. It is choosen to be 38.08 μs for reasons of convenience. After 16 successive collisions the attempt to transmit the packet is abandoned and a status of *load overflow* is returned for the benefit of higher level software.

Thus, the truncated backoff algorithm differs from the ideal binary exponential backoff algorithm in two respects. Firstly, the retransmission interval is limited to 9.7 ms (255. X 38.08 μs). Secondly, the host makes at most 16 attempts. to transmit a packet.

2.3 Previous Work

The literature contains many theoretical and simulation studies of CSMA-based local networks with data, real-time and integrated data/real-time traffic ([Almes & Lazowska, 79], [Tobagi & Hunt, 80], [Johnson & O'Leary, 81], [Nutt & Bayer, 82], [Tobagi & Cawley-Gonzalez, 82]). However, there has been only one reported substantive study of measurements of the behaviour of such a local network under normal traffic and artificially generated high loads [Shoch, 79]. A concise treatment of part of that study can be found in [Shoch & Hupp, 80]. Since this work. is an extension of those measurements and was performed on the same Ethernet, we summarize here some of the results from that study.

The average load during a typical 24-hour period is about 0.6 - 0.8% of the network capacity, with most of the traffic occurring during daytime hours. However, the peak load over a 1 second interval is as high as 37%, over a 1 minute interval, 17%. The load at night rarely exceeds a small fraction of 1%. Thus it is possible to conduct controlled experiments with specific traffic patterns and loads on the Ethernet during the late night hours. Error rates due to causes other than collisions are of the order of 1 packet in 2,000,000. Hence, we ignore such errors in our experiments

Under artificially-generated loads the following features were observed: For a given packet size, as the offered load increases from 0% of the network capacity, initially the throughput is equal to the offered load. As the offered load approaches and exceeds 100%, the throughput levels off to some value inversely proportional to the packet size. Figs. 2.1 and 2.2 show this behaviour (reproduced from [Shoch & Hupp, 80], Figs. 9 and 10(b) respectively). Thus, undcr high loads the Ethcmet access algorithm is obscrved to be stable, it achievcs a high channel utilization for large packets and the utilized capacity is shared fairly among all contending hosts.

FIG, 2.1

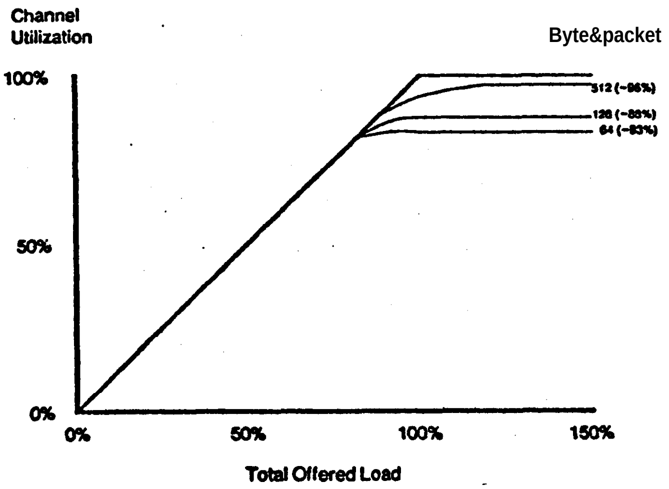
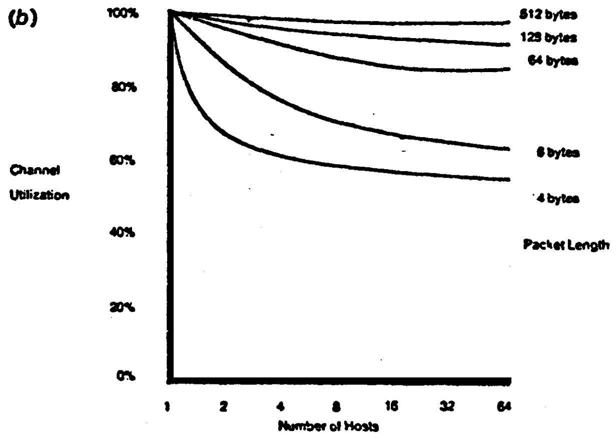


Fig. 9. Measured Utiliition of the Ethernet Network under High Load.

FIG, 2.2

Fig. 10. [illegible] . (b) Measured Utilization with Continuously Queued Sources.



(Reproduced from [Shoch & Hupp, 80] by permission of the ACM)

3. Experimental Environment

In this section we describe the techniques used to measure the performance of the Ethernet under varying load conditions. First, we describe the experimental setup and procedures. This is followed by a characterization of voice communication as it relates to packet networks. Finally, we describe the voice protocol modeled in our experiments and define notation used to represent traffic parameters and performance metrics.

3. I Experimental Setup and Procedures

To set up and run an experiment we use a special control program [Shoch & Hupp, 80] to find idle hosts on the network and to load a test program into each. The control program is then used to set parameters describing the traffic pattern to be generated by the test programs. Next the test programs actually generate traffic and record relevant statistics for the duration of the run. At the end of the run, the statistics are collected from the participating hosts by the control program. We ignore packets lost due to collisions which the transmitter cannot detect ([Shoch, 79], pg. 72) and due to noise. That is, we assume that if a packet is successfully transmitted it is also successfully received. Thus all the participating hosts in an experiment are transmitters of packets. In computing throughputs, we assume that the entire packet except for the 6-byte Ethernet header and checksum is useful data. Thus, our results represent upper bounds on performance since many actual applications will include additional protocol information in each packet.

The duration of each run is typically 60 seconds. Tests show that there is no significant variation in statistics for run times from 10 to 600 seconds. The control program can conveniently control up to 32 hosts. In order to control more hosts it is necessary to run the control program on two or more machines, each controlling up to 32 hosts. This considerably increases the effort required to set up and run an experiment and to collect and analyze the statistics. Hence, in most of our experiments we use at most 32 hosts.

Our experiments differ from those reported in [Shoch & Hupp, 80] in two respects. Firstly, we measure packet delays in addition to throughput and the other quantities measured in the earlier experiments. . Delay is clearly an important performance metric particularly for real-time applications such as voice communications. Secondly, we use different traffic patterns. The traffic patterns used by Shoch and Hupp model typical data applications. We use traffic patterns which model voice conversations as described below.

3.2 Characteristics of Packet- Voice Communications

Here we describe the characteristics of voice traffic as they relate to packet-switching local networks. By voice we refer to two-way voice conversations or telephony. However, other real-time applications such as remote monitoring and control of equipment may have similar characteristics.

Packet-voice differs from data traffic in several respects. In the former, a continuous analog signal is digitized at a constant rate by a coder. For instance, a typical PCM encoder operating at 8 KHz produces one 8-bit word every 125 μs. From the standpoint of network efficiency, it is desirable to transmit several coder samples in one packet- Constructing such packets introduces some delay depending on the packet size choosen. Further delay is incurred in gaining access to the network and in the actual transmission of the packet. For the communication mechanism to be acceptable to the users, the total delay must be within some bound (typically 100 ms for voice [AT&T, 80]). If the delay exceeds this maximum, the packet is discarded thus resulting in some loss of information. Further loss may occur because typical voice protocols do not use acknowledgements ([Shoch, 80a], [Johnson & O'Leary, 81]). Owing to the nature of speech the signal can be reconstructed at the receiver with acceptable quality, provided that the information loss is less than some specified fraction of the samples generated by the coder, typically 1% [Johnson & O'Leary, 81]. Note that the definition of acceptable voice quality in terms of delay and loss is subjective and depends on the coding scheme used.

In contrast, a data application such as a file transfer generates bursts of packets at occasional intervals. The data must be transmitted with 100% reliability, usually requiring acknowledgements at some level of the protocol, but larger and more 'varying delays can be tolerated.

3.3 A Voice Communication Protocol

We now describe the protocol modeled in our experiments. In order to mitigate the deleterious effects of contention at high loads, we use variable-length packets instead of the fixed-length packets used in most proposed protocols. We assume that 'a connection is set up and terminated between a pair of hosts in some manner. This part of the protocol is not germane to our study. Each host emulates a coder with a *sample generation rate* of R bits/second (bps). The generated samples are accumulated in a packet buffer. When the length of the packet in the buffer reaches the *minimum packet length,* P_{min} bytes, the host attempts to transmit the packet using the CSMA/CD scheme described in section 2.2. Whilst the host is trying to gain control of the network, the packet continues to grow as new samples are generated When the packet has been successfully transmitted, the packet-buffer is emptied and a new cycle starts. Since the sample generation is independent of the packet transmission process, collisions and retransmissions do not affect the continuous growth of the packet.

While the packet is being transmitted at the *channel transmission speed* C bps, it continues to grow at the rate R bps. Thus, in the absence of contention, the length of the *shortest packet,* P', *is* greater than P_{min}. During the time taken to transmit P' bytes the packet grows by $(P'/C)\times R$ bytes. Thus, we have:

$$P' = P_{min} + (P'/C)\times R$$

i.e. $P' = P_{min}/(1 - R/C)$ bytes.

The *delay* is defined to be the time from the generation of the first sample to the successful transmission of the entire packet. The propagation delay over the network is negligible for the cases which we study. We denote by *D* the delay averaged over all packets in an experimental run. The minimum delay depends on P':

$$D_{min} = P' \times 8 / R \text{ seconds.}$$

In order to ensure that the delay is bounded, the packet buffer is limited in size to the *maximum packet length*, P_{max} bytes. If, due to contention, the host cannot transmit the packet before its length reaches P_{max}, the buffer is managed as a FIFO queue, with the oldest sample being discarded when a new one is generated. Thus, we have the maximum delay:

$$D_{max} = P_{max} \times 8 / R \text{ seconds.}$$

The resulting ***loss*** *of infomation* is denoted by L expressed as a percentage of R.

The number of hosts participating in an experiment *is N. The throughput,* T, of the network is the aggregate data successfully transmitted by all N hosts expressed as a percentage of the network capacity. We use the subscript *Ln* to denote quantities measured when the average rate of loss per host is n%. Thus, T_{L1} and D_{L1} represent the throughput and delay when the loss per host is 1% of the generated samples.

To summarize, each host accumulates "voice" samples at R bps until it has a packet of length P_{min}. It then attempts to transmit, continuing to build the packet until it is successfully transmitted. A maximum, P_{max} is imposed on the packet length to bound the delay. kowever, this can cause loss of information due to contention.

4. Experimental Results

We now present experimental results which show the effects of the parameters N, R, P_{min} and P_{max} on the performance of the network and the quality of the communication. If we assume that in a two-way telephone conversation only one party talks at a time, then one host continuously generating information is approximately equivalent to one two-way conversation. Measurements have shown that this assumption is reasonable [Brady, 68]. Thus, if certain experiments show that N hosts can transmit with acceptable performance, we are justified in claiming that the Ethcmet can support approximately N simultaneous conversations under similar conditions.

In most of the experiments reported, we choose R = 105 Kbps, higher than the usual coder rate in voice systems. With this value of R it is possible for 32 hosts to offcr a total load of about 115% of the network capacity. WC also have some rcsults for lower values of R. For accurate timing, the coder emulator is written in microcodc which can generate one 16-bit sample every n X 38.08 µs, for n = 1, 2, . . . Thus R can be any submultiple of 420 Kbps. For example, with n = 4, R = 105 Kbps, and n = 6 yields R = 70 Kbps.

In sections 4.1 and 4.2 we examine the effects of varying P_{min} and P_{max} respectively. Then we present the results of a few experiments with R as a parameter which we use to justify some approximate* extrapolations of our results to 64 Kbps systems.

4.1 The Effect of the Minimum Packet Length, P_{min}

P_{max} is set to 1024 bytes to yield a bound on the delay of D_{max} = 78 ms. For P_{min} = 64,128 and 512 bytes we plot several families of curves. Figs. 4.1 and 4.2 show the mean delay, D, as a function of throughput, T, and the total offered load respectively. Also shown on the x-axis of Fig 4.2 is the number of hosts, N, corresponding to various offered loads. Figs. 4.3 and 4.4 show the variation' of loss, L, and throughput, T, respectively with offered load.

Delay. From Figs. 4.1 and 4.2 we see that for low to medium offered loads, D $\approx D_{min}$, i.e. there is little queueing delay in gaining control of the network. Most of the packets are approximately P_{min} in length. Then, at some point, the delay increases sharply. The throughput at this knee point increases with P_{min} As the offered load is increased above 100%. the throughputs for all values of P_{min} converge to 95% and the delays to D_{max}. This is because the queueing delays become so large that the lengths of most of the packets are close to P_{max}.

Fairness. Shoch and Hupp had observed that the Ethernet access strategy is fair in allocation of channel capacity among contending hosts even at high offered loads. We note that this fairness extends also to the mean delays suffered by individual hosts. For instance, with N = 32 and P_{min} = 64, the mean delays seen by individual hosts range from 97.5% to 103.2% of the delay averaged over all N hosts.

Loss. From Fig. 4.3 we see that for low to medium offered loads there is no loss. Then, from some value of offered load proportional to P_{min}, L increases almost linearly with offered load. Also, for a given acceptable loss rate, the maximum offered load and consequently throughput increase with P_{min}

From this discussion it would appear that the larger the value of P_{min} choosen the better, subject to any limits on D_{min}. However, consider a system operating near the knee-point of one of the curves. Any increase in offered load causes increased delay and loss. The rate of this performance degradation increases with P_{min}. From this consideration a low value of P_{min} is desirable. This trade-off is illustrated in Tables 4.1a and b which show T and N at the knee-point and 1% and 5% loss points respectively for the different values of P_{min}. Thus if the system is opcrating at thc knee-point, with P_{min} = 64, three additional hosts can start transmitting before L reaches 1% whilst with P_{min} = 512, any additional load causes L to exceed 1%.

Fig. 4.1 Delay vs. Throughput, variable Pmin
R = 105 Kbps, Pmax = 1024 Bytes

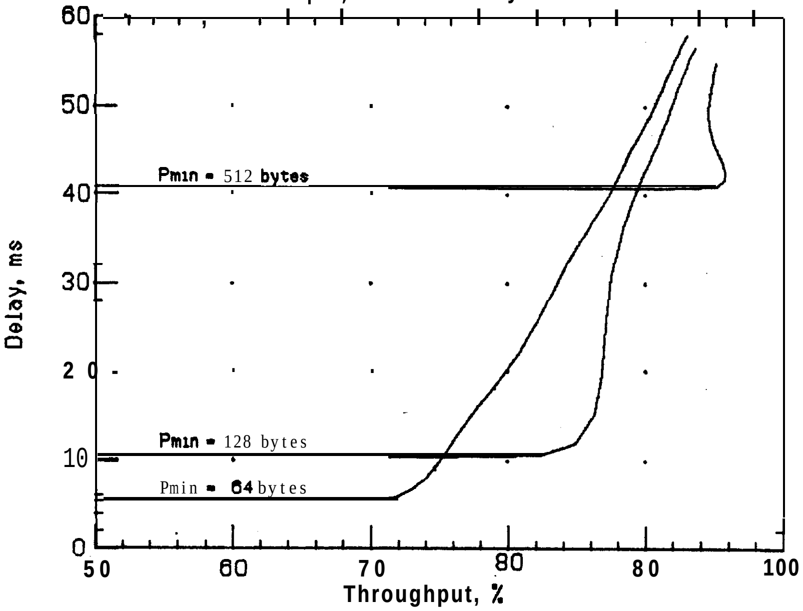
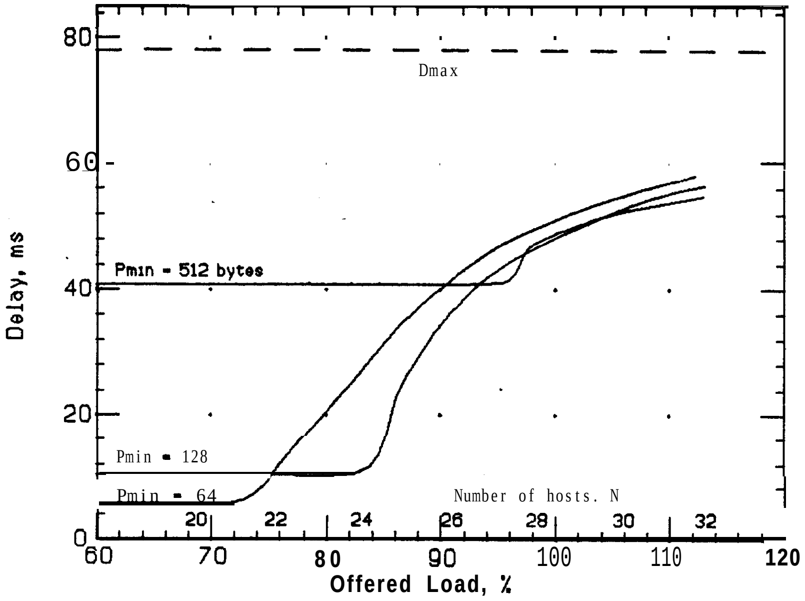


Fig. 4.2 Delay vs. Øffered Load, variable Pmin
R = 105 Kbps, Pmax = 1024 Bytes



F19. 4.3 Loss vs. ØFfered Load, varıable Pmın

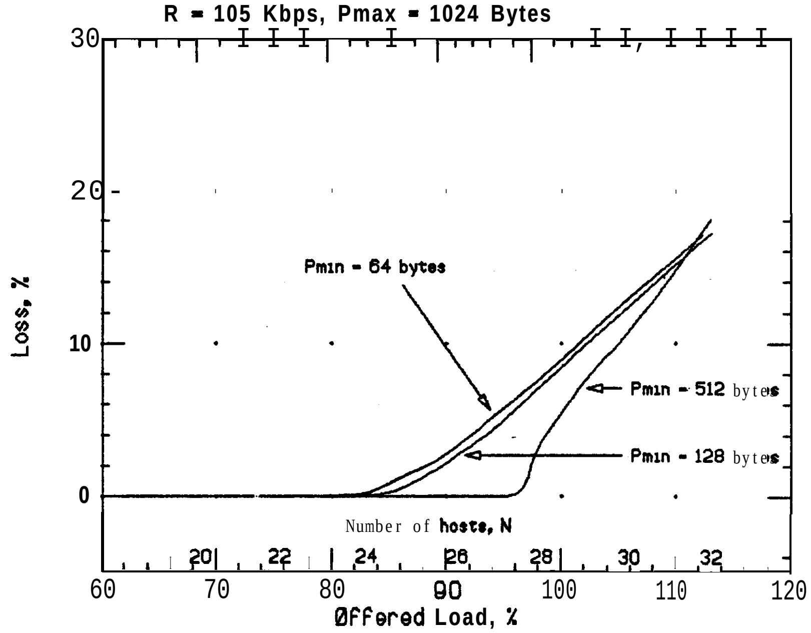


Fıg. 4.4 Throughput vs. ØFfered Load, variable Pmın

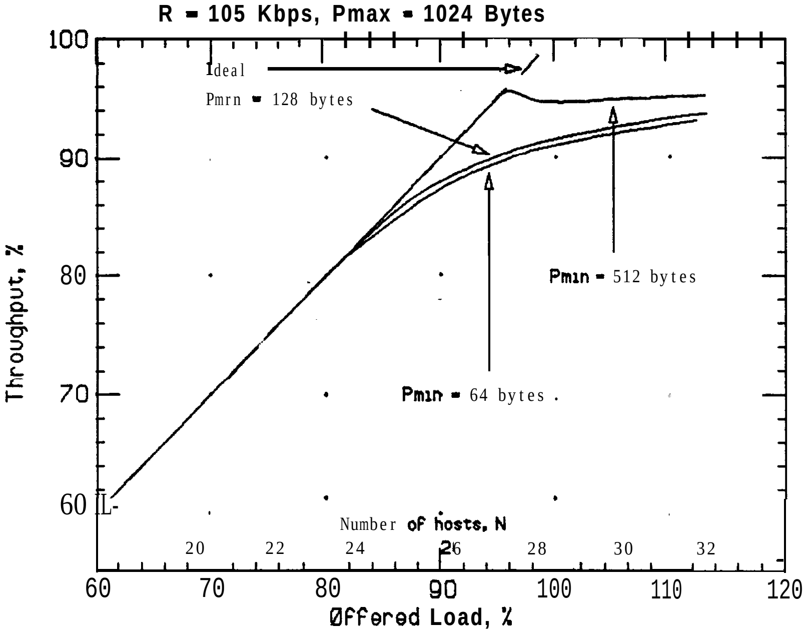


Table 4.1a

Throughput and delay at the knee-points

R = 105 Kbps, P_{max} = 1024 bytes

P_{min} (bytes)	D_{min} (ms)	T_{Knee} %	N_{Knee} (hosts)
64	5.4	72	-22
128	10.4	82	-24
512	40.7	95	-27

Table 4.1b

Throughput and delay with L = 1%, 5%

R = 105 Kbps, P_{max} = 1024 bytes

P_{min} (bytes)	D_{L1} (ms)	D_{L5} (ms)	T_{L1} %	T_{L5} %	N_{L1} (hosts)	N_{L5} (hosts)
64	33	45	84	89	-25	-27
128	29	43	87	90	-26	-27
512	45	48	95	95	-27	-28

It is clear that some form of higher-level access control, either distributed or centralized, is essential to avoid overloading of the network, which would cause degradation of performance for all the voice connections (a consequence of the fairness of the Ethernet protocol). The trade-off discussed above also manifests itself in the response time of the access control algorithm. As P_{min} increases, the algorithm must be able to react to changes in load more rapidly to maintain acceptable voice quality.

4.2 The Effect of the Maximum Packet Length, P_{max}

For convenience we fix P_{min} at 32 bytes and R at 105 Kbps. Figs. 4.5 to 4.8 show the effects of varying P_{max} (with axes corresponding to those of Figs. 4.1 to 4.4 respectively). Each set of curves is for P_{max} = 64, 128, 256 and 512 bytes, which correspond to D_{max} = 4.9, 9.8, 19.5 and 39.0 ms respectively. Again, we see that for low to medium offered loads D $\approx$ D_{min} and there is no loss. Each curve has a fairly well-defined knee-point, above which D increases rapidly to D_{max} and L increases roughly linearly with offered load. T tends asymptotically to some value dependent on P_{max}.

Fig. 4.5 Delay vs. Throughput, variable Pmax

R = 105 Kbps, Pmin = 32 Bytes

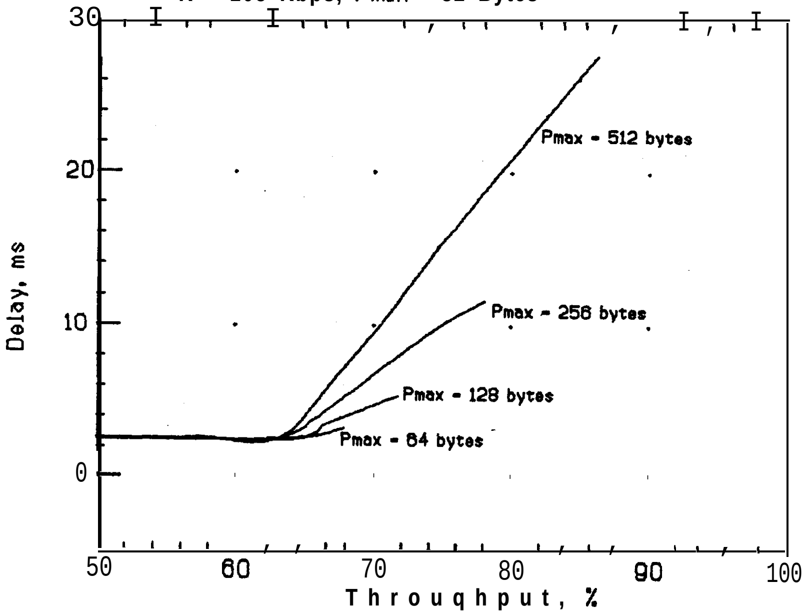


Fig. 4.6 Delay vs. Offered Load, variable Pmax

R = 105 Kbps, Pmin = 32 Bytes

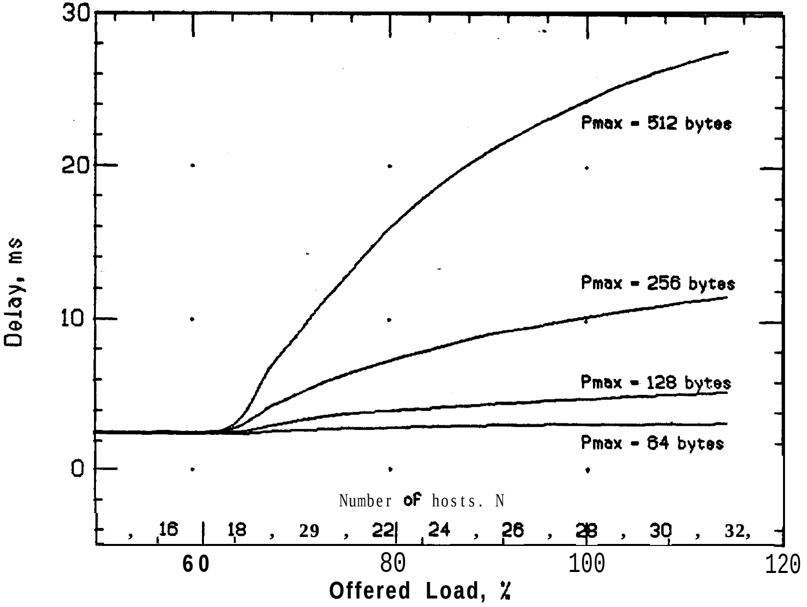


Fig. 4.7 Loss vs. ØFFered load, variable Pmax

R = 105 Kbps, Pmin = 32 Bytes

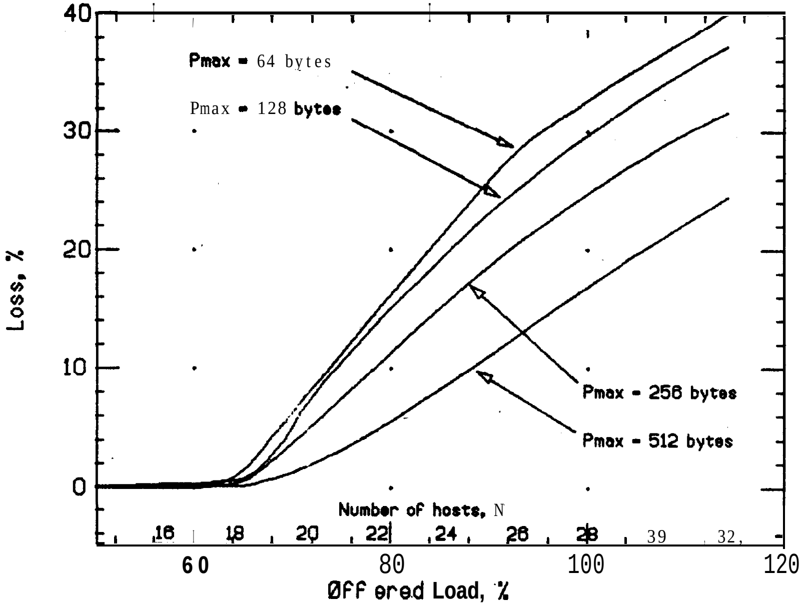


Fig. 4.8 Throughput vs. ØFFered Load, variable Pmax

R = 105 Kbps, Pmin = 32 Bytes

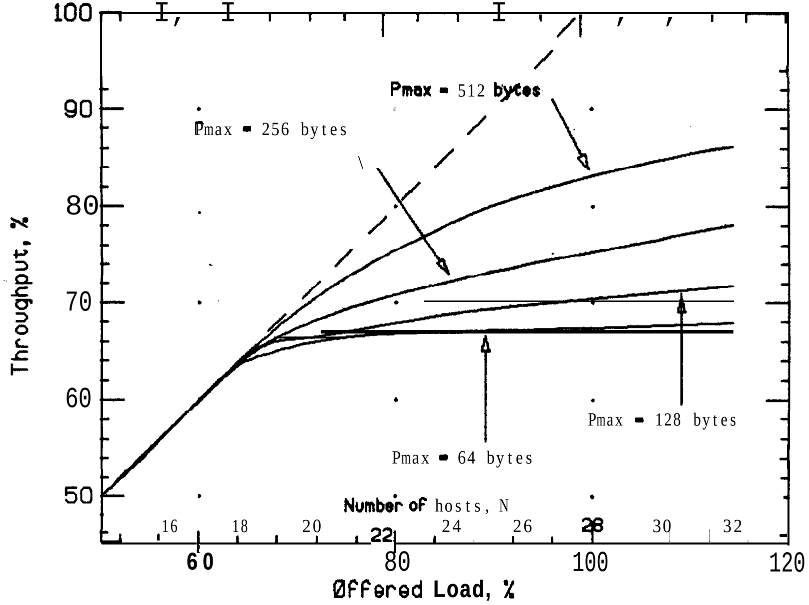


Table 4.2 shows some performance metrics for L = 1% and 5% which illustrate two noteworthy points. First, for a given loss rate, throughput increases with P_{max}. Second, the increase in throughput achieved by allowing a higher loss rate is larger for larger values of P_{max}. Thus, as the allowable maximum delay increases there is more incentive for the designer to accept a higher loss rate, thus increasing throughput and correspondingly the maximum number of hosts which can use the network simultaneously. Alternatively, if the acceptable loss rate is fixed, at higher loss rates there is more incentive to tolerate a higher maximum delay.

Table 4.2

Throughput and delay with L = 1%, 5%

R = 105 Kbps, P_{min} = 32 bytes

P_{max} (bytes)	D_{max} (ms)	T_{L1} %	T_{L5} %	N_{L1} (hosts)	N_{L5} (hosts)
64	4.9	63.7	65.5	~18	-19
128	9.8	64.5	66.0	~18	-19
256	195	66.0	68.0	-19	~20
512	39.0	68.0	75.0	-19	-22

4.3 *The Effect of the Coder Rate, R*

In this section we present some results of experiments conducted with various values of R. We make an important observation which enables us to extrapolate the results of the previous two sections to coder rates lower than 105 Kbps, specifically, 64 Kbps.

Figs. 4.9 and 4.10 show the *normalized delay, D/D_{min}* and loss respectively as functions of offered load for coder rates, R = 70, 84 and 105 Kbps. P_{min} = 64 and P_{max} = 1024. From these figures we observe that changing the coder rate has little effect on performance. The offered load at which the normalized delay and loss rate reach specific values varies by at most 5% for the various values of R. Further, for a given rate of loss, the normalized delay and throughput do not change much with R. This is shown in Table 4.3.

Thus, we can make the claim that, other parameters being constant, the performance curves with R = 64 Kbps are similar in shape and value to those for R = 105 Kbps *provided we consider normalized delays.* Thus, the effects of P_{min} and P_{max} which we observed in the previous two subsections will also be observed in a systcm operating with 64 Kbps codcrs. Note that if the limits on packet lcngths are the same, the limits on the absolute delay are invcrscly proportional to R. For instance, with P_{max} = 1024 bytes, D_{max} = 78 ms when R = 105 Kbps but it increases to 122 ms when R decreases to 64 Kbps.

Fig. 4.9 Norm. Delay vs. ØFFered Load, variable R

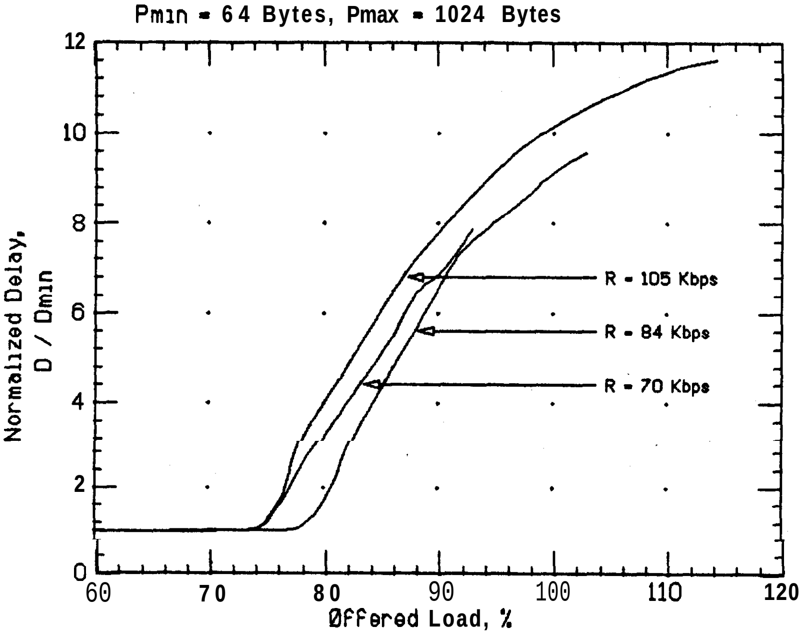


Fig. 4.10 Loss vs. ØFFered Load, variable R

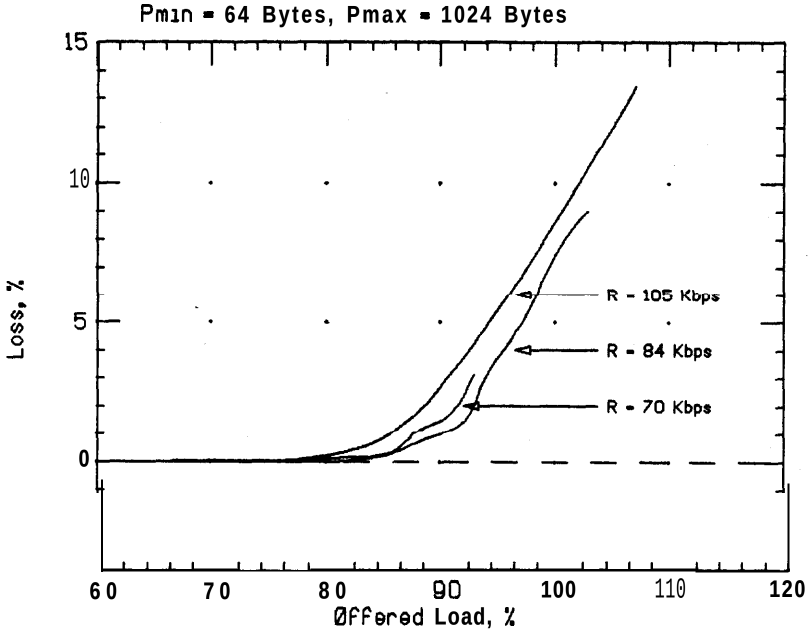


Table 4.3

Throughput and normalized delay with L = **1%, 3%**

P_{min} = 64 bytes, P_{max} = 1024 bytes

R	Normalized Delay		T_{L1}	T_{L3}
(Kbps)	**L1**	**L3**	**%**	**%**
m-m--	- - m -	--------	--------	--------
70	*6.3*	*7.8*	*86.8*	*92.6*
84	*6.6*	*7.8*	*88.5*	*90.8*
105	*6.3*	*7.9*	*84.8*	*88.0*

In this context we recall the observation made by **Shoch** and Hupp that, with hosts continuously attempting to transmit fixed-length packets, the throughput was mainly **a function** of the total offered load and the packet length. The number of hosts attempting to transmit had little effect of the throughput. This adds some weight to our claim above.

4.4 Summary

To summarize the results of this section, within limits determined by the parameters **choosen, the** 3 Mbps Ethernet is capable of supporting voice applications with minimal queueing delay and no loss of information. This is true also of other real-time applications with similar characteristics. . Above some offered load the delay increases and there is loss of information. The Ethernet protocol is fair in that the delay and loss suffered by all contending hosts are approximately equal. Also, the utilized throughput is shared equally among these hosts.

Subject to delay constraints, it is desirable to choose a fairly large value of P_{min}. Also, given a maximum delay constraint and hence a corresponding value of P_{max}, the throughput increase achieved by accepting a higher loss rate increases with P_{max}. Thus, as the maximum delay constraint increases there is more incentive for the designer to accept a higher loss rate, and, perhaps, to mitigate the effect of this higher loss by mechanisms in other parts of the system.

As a consequence of the fairness of the Ethernet protocol, it is essential to restrict the number of simultaneously active connections to avoid overloading the network. **I.1** a system devoted solely to voice, this can be **implemcnted** easily in either a centralized or distributed manner. To implement centralized control, a host wishing to open a new connection would have to request permission from some server which would refuse permission if the existing number of connections was above some threshold determined by parameters such as the minimum and maximum packet **lcngths. Dcccntralized** control could be implemented analogously. Before opening a new connection a host would monitor the traffic for a short period to determine the number of existing connections and would make a decision based on the threshold.

The case of an integrated voice/data network is more interesting. Since data traffic is not necessarily connection-based and is bursty the above control algorithms would not succeed entirely in eliminating spikes of high load. Whether this problem can be satisfactorily solved in a priorityless Ethernet is a topic for further investigation. One possible solution is to allow only connection-based traffic. Voice and data connections would be accorded similar treatment. This problem has also been addressed by the introduction of packet-level priorities in the basic CSMA scheme [Tobagi, 82].

5. Conclusions

We have proposed a simple protocol for voice communication on packet-switching local computer networks. Measurements conducted-on an experimental 3 Mbps Ethernet using this protocol as a model have shown that the network is capable of supporting voice traffic with acceptable perfromance up to substantial fractions of its capacity. For instance, from the curve for R = 70 Kbps in Fig. 4.7 we can make the following prediction: Using 8-bit PCM coders and a sampling rate of 8 KHz (i.e. R = 64 Kbps), with P_{min} = 64 bytes and P_{max} = 1024 bytes, the network can support approximately 35 conversations (i.e. 70 users) with typical delays of less than 10 ms (>99% of the samples), loss of less than 0.1% of the coder samples and a maximum delay of 120 ms. At any given time only a fraction of the installed telephones in a system are in use. Assuming a peak usage of 20% of the telephones, we could attach 350 telephones to the Ethernet If we allow a loss rate of 1%, the corresponding numbers are 40 simultaneous conversations and 400 telephones attached to the Ethernet. The use of sophisticated, lower-bandwidth vocoders could increase these numbers appreciably.

At high offered loads the fairness of the Ethernet protocol causes degradation of performance of all voice connections. Thus, to ensure acceptable quality, some higher-level access control algorithm is necessary to avoid peaks in the offered load This is especially important in an integrated voice/data network in the absence of packet-level priorities. This is an area for further study.

An important contribution of this research is that we have substantially increased the base of experimental data available on the performance of the 3 Mbps experimental Ethernet under diverse conditions. This database' can be used to develop and. validate theoretical models for the purpose of studying other implementations of the Ethernet architecture, for instance [Ethernet, 80], for various applications. This is currently under investigation.

Although our results were obtained using a specific voice protocol, we feel that using other protocols would not improve performance appreciably. In fact, since we have not considered software and protocol overhead, our results may represent an upper bounds on the performance of the 3 Mbps Ethernet for voice applications.

While it is tempting to extrapolate our results to the 10-Mbps Ethernet, caution must be exercised in doing so. A simple model [Metcalfe & Boggs, 76] indicates that, other parameters being constant, as the channel capacity of an Ethernet is increased linearly the resulting increase in throughput is less than linear.

6. Acknowledgements

I wish to thank several people for their contributions to this paper. Forest Baskett introduced me to this avenue of research and provided helpful advice along the way. Some of the ideas in this paper, including the voice protocol, are his. John Shoch made available to me the measurement tools developed by himself and Jon Hupp, thus considerably reducing my programming load Yogen Dalal's incisive technical and editorial comments occasioned substantial revisions of several drafts of this paper.

References

[Almes & Lazowska, 79]
G. T. Almes, and E. D. Lazowska, "The Behavior of Ethernet-like Computer Communication Networks," in *Proc. of 7th Symp. on Operating Sys. Prins.,* Asilomar, California, pp. 66-81, Dec. 1979.

[AT&T, 80]
AT&T., Notes on *the* Network, American Telephone and Telegraph Company, Network Planning Div., Fundamental Network Planning Section, 1980.

[Brady, 68]
P. T. Brady, "A statistical analysis of on-off patterns in 16 conversations," *Bell System Tech. Journal,* vol. 47, no. 1, pp. 73-91, Jan. 1968.

[Crane & Taft, 80]
R. C. Crane , and E. A. Taft, "Practical Considerations in Ethernet Local Network Design," in *Proc. of the 13th Hawaii International Conf. on System Sciences,* Honolulu, pp. 166-174, Jan. 1980.

[Ethernet, 80]
"The Ethernet, A Local Area Network: Data Link Layer and Physical Layer Specifications," DEC, Intel & Xerox Corps., Version 1.0, 1980.

[Johnson & O'Leary, 81]
D. Johnson, and G. O'Leary, "A Local Access Network for Packctized Digital Voice Communication," *IEEE Tr. Comm.,* vol. COM-29, no. 5, pp. 679-688, May 1981.

[Klcinrock & Tobagi, 75]
L. Kleinrock, and F. Tobagi, "Packet Switching in Radio Channels: Part I -- Carrier Sense Multiple-access Modcs and thcir Throughput-delay Charactcristics," *IEEE Trans. Comm.,* vol.

COM-23, no. 12, pp. 1400-1416, Dec. 1975.

[Metcalfe & Boggs, 76]
R. M. Metcalfe, and D. R. Boggs, "Ethernet: Distributed Packet Switching for Local Computer Networks," *Commun Ass. Comput. Mach,* vol. 19, no. 7, pp. 395-404, Jul. 1976.

[Nutt & Bayer, 82]
G. J. Nutt, and D. L. Bayer, "Performance of CSMA/CD Networks Under Combined Voice and Data Loads," *IEEE Trans. Comm,* vol. COM-30, no. 1, pp. 6-11, Jan. 1982.

[Shoch, 79]
. J. F. Shoch, "Design and Performance of Local Computer Networks," Ph.D. Thesis, Dept. of Computer Science, Stanford University, Stanford, CA-94305, 204 pp., Aug. 1979.

[Shoch, 80]
J. F. Shoch, "An Annotated Bibliography on Local Computer Networks (3rd edition)," Xerox Parc TR SSL-80-2, and IFIP Working Group 6.4 Working Paper 80-12, Apr. 1980.

[Shoch, 80a]
J. F. Shoch, "Carrying voice traffic through an Ethernet local network -- a general overview," presented at *IFIP WG* 6.4 *Intl. Workshop on Local-Area Computer Networks,*. Zurich, Aug. 1980.

[Shoch & Hupp, 80]
J. F. Shoch, and J. Hupp, "Measured Performance of an Ethernet Local Network," *Commun Ass. Comput. Mach.,* vol. 23, no. 12, pp. 711-721, Dec. 1980.

[Thacker et. *al.,* 82]
C. P. Thacker, E. M. McCreight, B. W. Lampson, R. F. Sproull, and D. R. Boggs, "Alto: A Personal Computer," in D. P. Siewiorek, C. G. Bell, and A. Newell (Eds.), *Computer Structures= Principles and Examples,* McGraw-Hill Book Co., pp. 549-572, 1982.

[Tobagi & Hunt, 80]
F. A. Tobagi, and V. B. Hunt, *'Performance Analysis of Carrier Sense Multiple Access with Collision Detection," *Computer Networks,* vol. 4, no. 5, pp. 245-259, Oct/Nov. 1980.

[Tobagi, 82]
F. A. Tobagi, "Carrier Sense Multiple Acess with Message-Based Priority Functions," *IEEE Trans. Comm,* vol. COM-30, no. 1, pp. 185-200, Jan. 1982.

[Tobagi & Cawley-Gonzalez, 82]
F. A. Tobagi, and N. Cawley-Gonzalez, "On CSMA-CD Local Networks and Voice Communication,," to be presented at *ZNFOCOM '82,* Las Vegas, Nevada, Mar./Apr. 1982.