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

Timothy A. Gonsalves

CSL-82-5 **March 1982** [P82-00034]

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 chosen 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.

A version of this paper has been submitted for presentation at the 3rd International Conference on Distributed Computing Systems, Miami, Florida, October 1982.

CR Categories: C.2.5, C.4

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

Author's Affiliation: Computer Systems Laboratory, Department of Electrical Engineering, Stanford University, Stanford, California 94305.

XEROX

Xerox Corporation
Palo Alto Research Centers
3333 Coyote Hill Road
Palo Alto, California 94304

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 computer 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 Centers, 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 Ethernet used for the experiments reported in this paper. We then summarize some of the results on measured performance of this Ethernet under artificially generated loads reported in [Shoch & Hupp, 80].

To avoid this problem, the Ethernet 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 ANDing 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 chosen to be X time units. The time unit should be no less than the round-trip end-to-end propagation delay. It is chosen 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 binary exponential backoff algorithm in two respects. Firstly, the retransmission interval is limited to 9.7 ms (255 $\times$ 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 is increased 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, under high loads the

Ethernet access algorithm is observed to be stable, it achieves a high channel utilization for large packets, and the utilized capacity is shared fairly among all contending hosts.

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.

3.1 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], p. 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.

greater than P_{min}. During the time taken to transmit P′ bytes the packet grows by (P′/C)×R bytes. Thus, we have:

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

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

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 information* 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. However, 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 Ethernet can support approximately N simultaneous conversations under similar conditions.

In most of the experiments reported, we chose R = 105 Kbps, higher than the usual coder rate in voice systems. With this value of R it is possible for 32 hosts to offer a total load of about 115%

Fig. 4.1 Delay vs. Throughput, variable P_{min}
$R = 105$ Kbps, $P_{max} = 1024$ Bytes

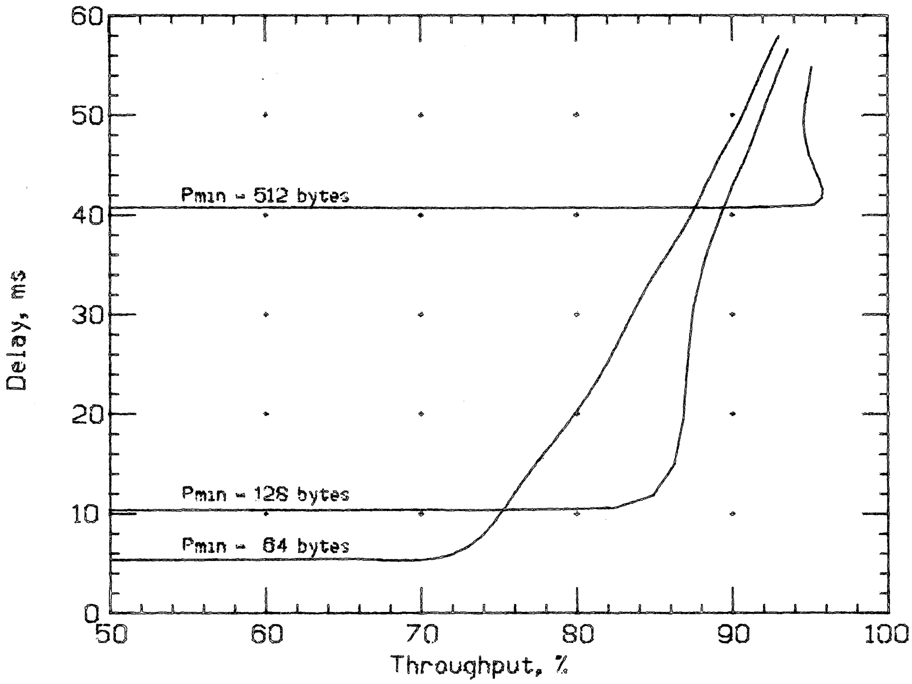
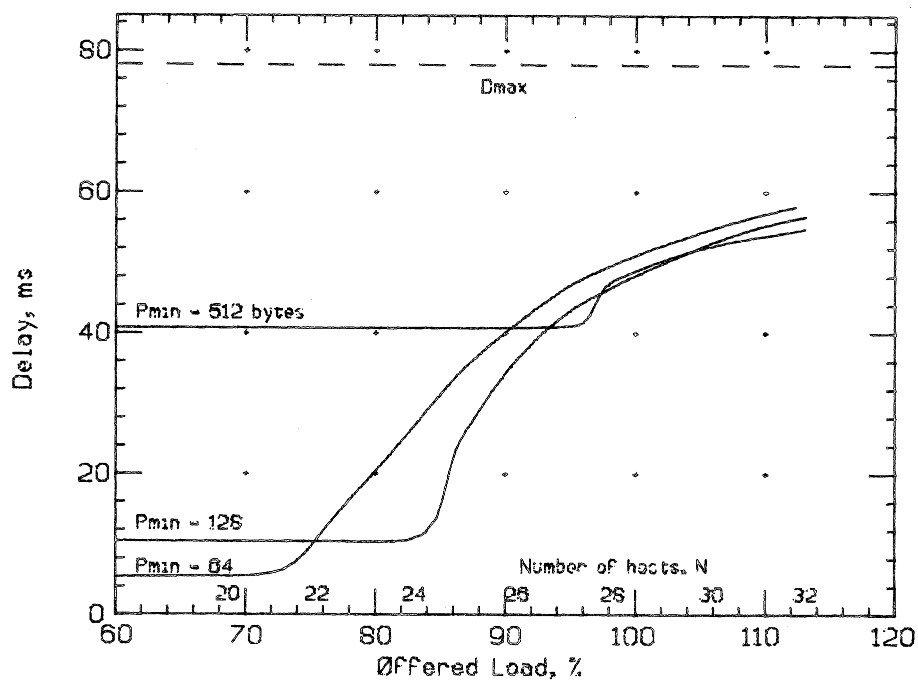


Fig. 4.2 Delay vs. Offered Load, variable P_{min}
$R = 105$ Kbps, $P_{max} = 1024$ Bytes



with P_{min} = 512, any additional load causes L to exceed 1%.

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

Fig. 4.7 Loss vs. Offered Load, variable P_{max}
$R = 105$ Kbps, $P_{min} = 32$ Bytes

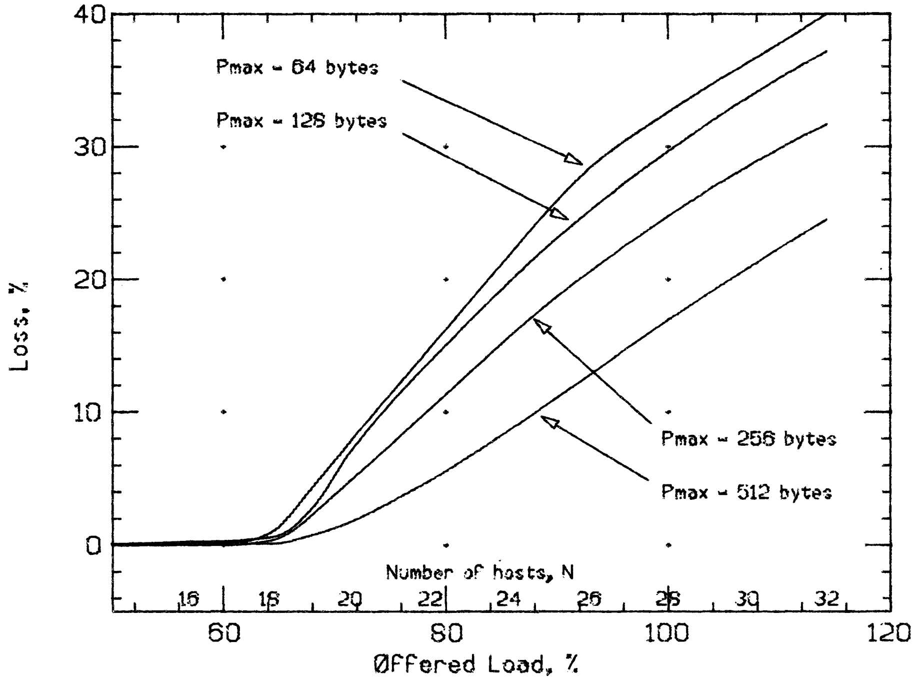


Fig. 4.8 Throughput vs. Offered Load, variable P_{max}
$R = 105$ Kbps, $P_{min} = 32$ Bytes

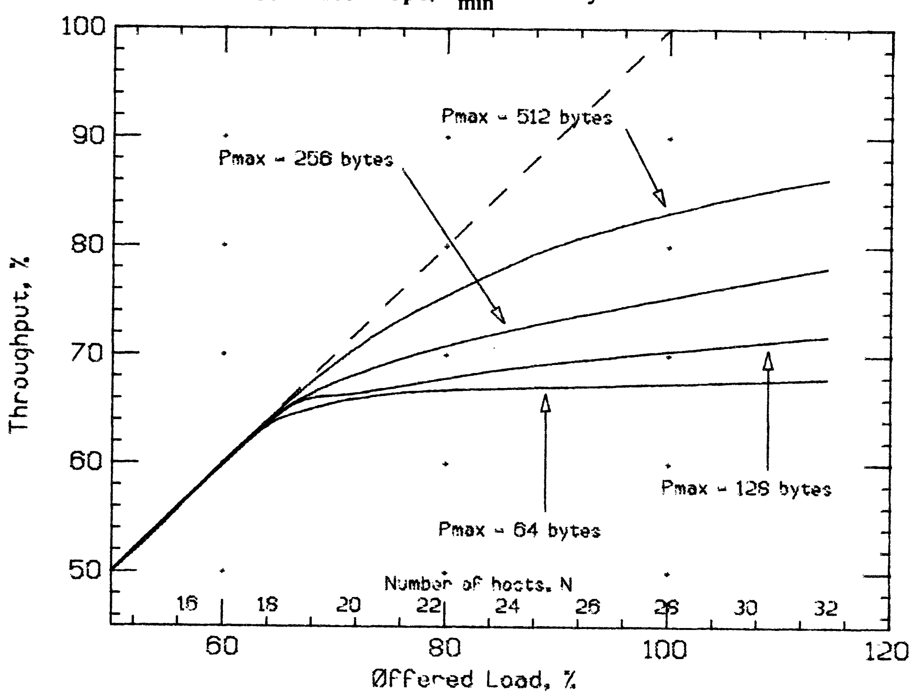


Fig. 4.9 Normalized Delay vs. Offered Load, variable R
P_{min} = 64 Bytes, P_{max} = 1024 Bytes

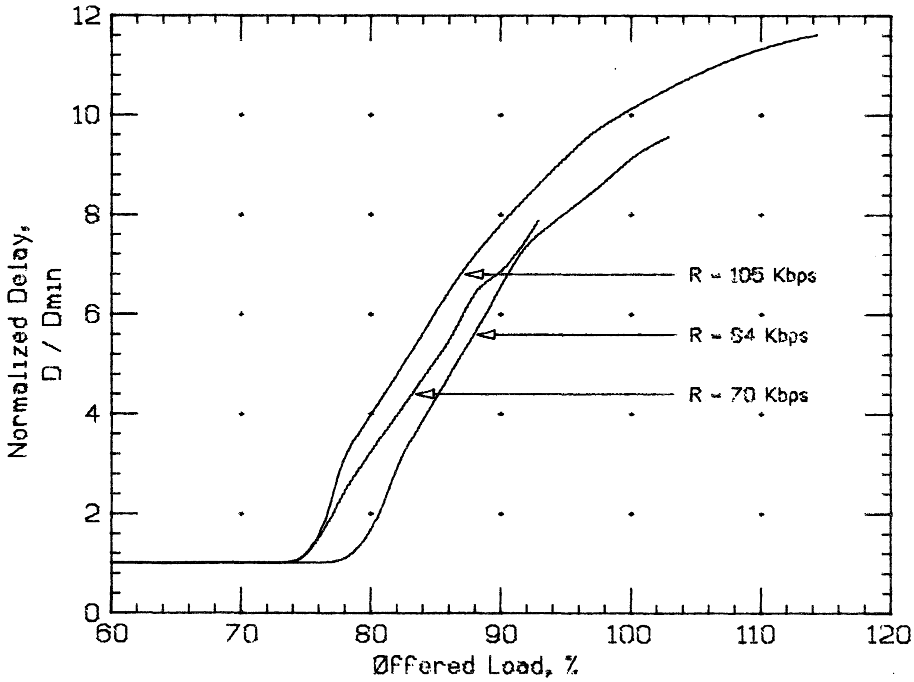
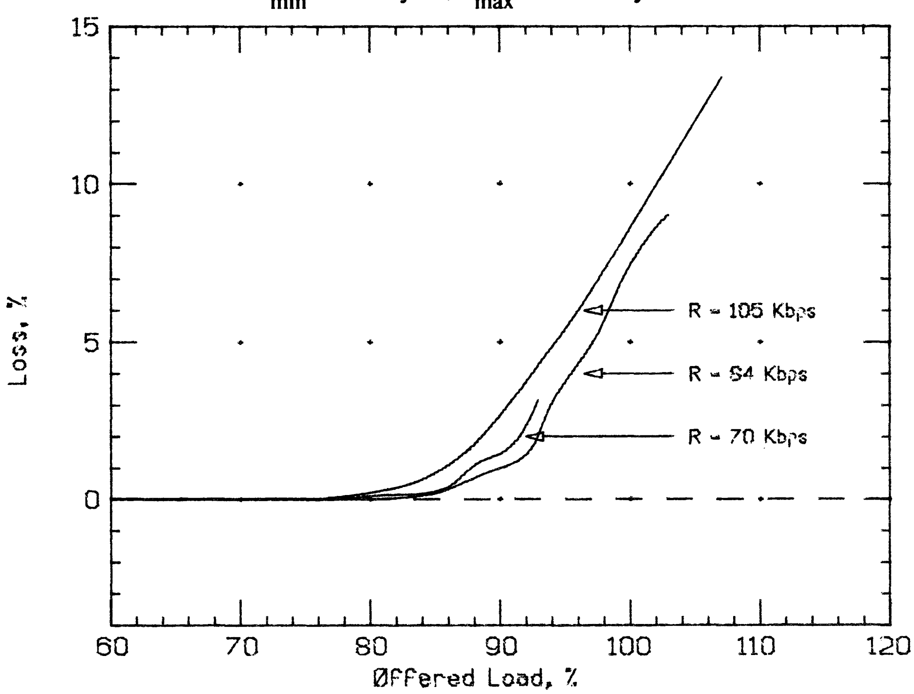


Fig. 4.10 Loss vs. Offered Load, variable R
P_{min} = 64 Bytes, P_{max} = 1024 Bytes



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 priority-less 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 performance 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.

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, California 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, Switzerland, 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," in *Proc. of INFOCOM '82*, Las Vegas, Nevada, Mar./Apr. 1982.