AT&T BELL LABORATORIES TECHNICAL JOURNAL
Vol. 63, No. 1, January 1984
Printed in U.S.A.

A Simulation-Based Comparison of Voice Transmission on CSMA/CD Networks and on Token Buses

By J. D. DeTREVILLE*

(Manuscript received March 14, 1983)

Digitized speech can be transmitted over a variety of digital media. An interesting choice is the use of a Local-Area Network (LAN), for which digitized speech is packetized at the transmitter and depacketized at the receiver. Many local-area networks exhibit good throughput but poor delay characteristics; variable or excessive transmission delay can become noticeable and objectionable to the users of such a voice system. A number of simulations were performed to assess the delay characteristics of a Carrier Sense Multiple Access/Collision Detection (CSMA/CD) LAN and of a similar token bus LAN. A comparison of the results shows that the token bus performs somewhat better. The CSMA/CD LAN's performance was characterized by carrying voice well until a point of collapse is reached; the token bus's performance degraded more continuously. In either case, throughput close to the theoretical capacity of the LAN was found achievable with appropriate techniques.

I. CHARACTERISTICS OF DIGITAL VOICE

Human speech of telephone quality can be easily encoded into a 64-kb/s bit-stream containing 8000 8-bit speech samples per second; although much more efficient encodings are possible, this synchronous 64-kb/s speech encoding is assumed throughout this paper.† Speech

* AT&T Bell Laboratories.

† As a simple improvement, the use of delta-modulation to transmit only the differences between successive samples could produce savings of approximately 2:1. More extensive processing could result in more extensive savings by taking further advantage of the regular properties of human speech. Improvements in the encoding bit rate will

Table I—Speech sample delay breakdown

Type of Delay		Consisting of
Fixed	The (nominal) temporal length of the packet: the packet size measured by its acquisition time	The delay after the packet is acquired until the packet is completed and transmitted (the temporal length of the portion of the packet following this sample)
		Plus the delay after the packet is received until the sample is played back (the temporal length of the portion of the packet preceding this sample)
	Plus much smaller fixed delays (e.g., the transmission time)	
Variable	The delay in transmitting the packet	The delay in obtaining the transmission medium
	Plus (typically) smaller variable delays	

consists of *talkspurts* separated by *silences:* a speaker in a typical conversation talks about 40 percent of the time and is silent for the remainder, and an approach that transmits speech only during talkspurts can therefore be desirable. Silences, of course, are relative. Ideally, no speech should be lost by being considered silence, and no extraneous background sounds should intrude during silences. This ideal can be approached through the use of cutoff levels with memory.

Transmitting digital speech over a shared packet network entails packetizing the digital signal at the transmitter, transporting it over the network, and depacketizing it at the receiver; these operations can introduce delay. Table I gives a high-level breakdown of the delay in the transmission. The delay includes a fixed component and a variable component. Because of the variable component, if the receiving station begins playing a packet as soon as it is received, this can introduce artificial silences at some points (when a packet is delayed more than the one before it) and lost speech at others (when a packet is delayed less than the one before it), ultimately producing effects audible to the users. This variability of performance can be partially overcome by artificially delaying packets at the receiver, such that only those packets whose variable delay is greater than some threshold will cause anomalies; since speech is inherently real-time, arbitrary queueing of packets at the transmitter or receiver is not possible.

The user-level model of speech used in this paper is that of a typical two-way conversation, in which real-time constraints exist at both

result in improvements in the performance figures presented in this paper, but these performance improvements will typically not be linear, since a reduction in the bit rate will make other factors relatively more important. Similarly, although variable bit-rate encodings can produce further savings over fixed bit-rate encodings, they can lose many of the advantages shown for fixed bit rates in this paper, and will again have less of a total impact than might otherwise be expected.

ends. If either side of the voice conversation were to be a computer or similar device, knowledge of this fact could be used to ease the constraints somewhat, although this optimization is not considered in this paper. If both ends were known to be computers, speech could then be transmitted as a nonreal-time data transfer.

If packets are artificially delayed, the one-way voice sample delay from the transmitter to the receiver during successful transmission will roughly equal the packet size plus the threshold delay. Increasing the packet size will increase the effective bandwidth of the system (by reducing per-packet overhead); increasing the artificial delay will reduce the incidence of anomalies (by reducing the probability that a packet will have been delayed for longer than the threshold). Reducing the traffic speeds access to the shared network; reducing anomalies postpones the onset of overload. On the other hand, increasing these values increases the delay through the system, which will eventually become perceptible to the user; this suggests a compromise between the extremes. For example, the one-way delay on a single-hop synchronous-orbit satellite voice circuit is 270 ms, which many users view as disruptive; the double-hop delay of 540 ms is considered much worse. Considering that a system built on one LAN may frequently communicate with another system on another LAN (thereby at least doubling the end-to-end delay), this suggests that the one-way delay on a given LAN should be kept well below 270/2 = 135 ms. An alternative might be to treat inter-LAN connections differently from intra-LAN connections; this possibility is not considered here. In any case, the delay cannot be allowed to grow without bound. It should be noted that echo is perceived as being much more disruptive than simple delay, with the audible threshold occurring much earlier, but echos, where they might occur, can be controlled through the use of echo cancelers. The exact nature of this compromise depends upon the precise psychoacoustic characteristics of the importance of this delay compared to, say, the effect of the anomalies caused by variable delay; this trade-off is not well understood.

II. A TYPICAL CSMA/CD LAN

*Ethernet** is a typical Carrier-Sense Multiple Access/Collision Detection (CSMA/CD) LAN.[1] Data packets are transmitted bidirectionally over a coaxial cable with an acyclic branching topology. Access to the net is distributed ("multiple access") and statistical. A station wishing to transmit first listens to determine whether the net is in use ("carrier sense"); if it is, the station defers until the current user has finished transmitting its packet. If the net is not in use, the station

* *Ethernet* is a trademark of Xerox Corporation.

begins to transmit. Due to race conditions, two stations could begin to transmit simultaneously; when one station notices another transmitting ("collision detection"), it aborts its transmission, jams the net to ensure that other stations also notice the collision and abort their transmissions, and retries after a random amount of time, thereby statistically avoiding recollision.

An *Ethernet* CSMA/CD network is bit serial and runs at 10 Mb/s ; a bit-time is thus 0.1 μs. Assuming 64-kb/s speech, complete utilization of the bandwidth would result in carrying up to 195.3 two-person conversations (in which each person spoke 40 percent of the time). Such efficiency, however, can never be achieved in practice.

One reason is simple per-packet overhead. A transmission on an *Ethernet* CSMA/CD network begins with 64 sync bits, followed by the packet. A packet contains 112 bits of header, a 368- to 12,000-bit data field (thus between 5.75 ms and 187.5 ms of 64 kb/s speech) and a 32-bit CRC field. A station may begin to transmit when it has seen the net idle for 96 bit-times. Assuming (arbitrarily) that voice stations are uniformly distributed along a maximum-length linear CSMA/CD network, computations based on the *Ethernet* propagation delay budget give a worst-case mean one-way propagation time of about 10.06 μs; we can expect an arbitrary station to see the net go idle 100.6 bit-times after the arbitrary preceding station actually ceased to transmit. A linear CSMA/CD network is in ways a "best case," since the mean distance between stations will be less than in a more general topology. However, the limiting case in complex topologies is extremely unlikely. Similarly, uniform distribution is a "best case," but a more accurate characterization seems difficult to achieve.

Taking per-packet overhead into account, we see that, at the smallest packet size, the speech samples can occupy only 47.6 percent of the bandwidth, allowing a maximum of 93.0 conversations; at the largest packet size, 96.7 percent of the bandwidth can be speech samples, allowing 188.9 conversations.

Studies of the *Ethernet* specifications under varying load conditions have typically shown *Ethernet* CSMA/CD networks to have very desirable throughput characteristics (e.g., see Ref. 2). Throughput tends to rise linearly with offered load until saturation is approached, and then levels off, with an asymptotic throughput within a few percent of maximum for large packets and within several percent for small packets. (For an experimental *Ethernet* CSMA/CD network described in Ref. 2, whose numerical parameters differed significantly from those of the specifications discussed here, measured throughput reached 96 percent for maximum-size packets, and 83 percent for minimum-size packets. The traffic in this study, as in the case considered here, was produced by a number of stations each offering a fraction of the total load.) Throughput may decrease under certain cases of extreme overload (for example, two stations each attempting to offer 100-percent load to the net would ultimately transmit less data together than either would individually, due to their contention), although this decrease evidently does not become pathological.

On the other hand, individual packets may experience significantly greater delays under heavy loads than under light loads. The nature of this increase in the delay has not been well characterized in past studies of data traffic, which is not as badly affected by variable delays as is real-time voice. Although we can be reasonably certain that the voice samples will make the journey from the transmitter to the receiver, almost up to the physical transport limits of the network, this might be inadequate if they require excessive time to do so.

III. A SIMULATION STUDY

To determine the performance characteristics of voice traffic on a CSMA/CD network, a computer simulation was prepared based on *Ethernet* specifications. The stations were assumed to be uniformly distributed over a maximum-length network. Both voice and data traffic were modeled.

The voice stations modeled typical two-person conversations. The stations were therefore paired, with an appropriate distribution and correlation of talkspurts and silences (adapted from Brady, as discussed in Ref. 3). Brady's study included filtering out very short silences and very short talkspurts, thereby increasing the mean length of silences and talkspurts and otherwise modifying their distribution. The exact type of voice filtering best suited for transmission over an LAN is still uncertain. Voice packets were not transmitted during silences. The simulations began with one conversation, after which an additional conversation was added every 0.5 second of simulated time, until the system had passed saturation. This staged introduction of conversations helps to eliminate anomalies associated with the start-up of several conversations at once. Although it is possible that the monotonically increasing number of conversations could produce history artifacts in the simulation results, none were observed in the CSMA/CD simulations; these did occur in the token bus studies outlined in Section IX.

Figure 1 shows the actual number of speakers over time, as a function of the number of conversations, for one particular voice traffic pattern; this pattern and another like it were used throughout the simulations to control the effect of differing traffic patterns in separate simulations. (As it turned out, the effect of a particular voice traffic pattern on observed behavior was less than anticipated, and is

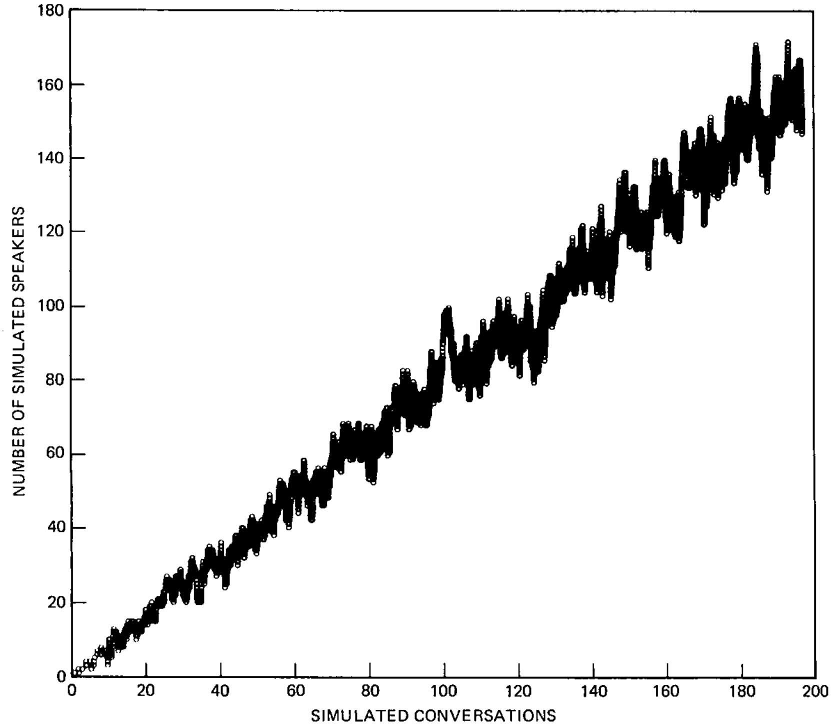

Fig. 1—Number of simulated speakers in a voice traffic pattern. This graph presents a voice traffic pattern used in most of the simulations presented in this paper, plotting the number of instantaneous speakers as a function of the number of conversations. One simulated two-person conversation is added every 0.5 second of simulated time; there are 0.8 expected instantaneous speakers per conversation. Speakers divide their time between talkspurts and silences. This simulation uses an empirically derived distribution of the lengths of talkspurts and silences and of the correlation between the states of the two potential speakers in a conversation.

easily compensated for. Thus, the use of the same voice traffic patterns throughout the simulations seems to have been unnecessary.)

The data stations presented a bimodal distribution of packet lengths, typical of data traffic on real nets, with 80-percent minimum-size packets and 20-percent maximum-size packets (giving approximately the opposite distribution when weighted by length). The packet arrivals were modeled by a Poisson process: the traffic generated by a Poisson process is not as bursty as real data traffic, but the difference was expected to be relatively unimportant in determining the effect of the data traffic upon the voice traffic. The simulations included between 0- and 10-percent steady data loading of the system, the latter value being well beyond the measured steady loadings of current *Ethernet* CSMA/CD networks.

IV. VOICE TRANSPORT ALGORITHMS

The simplest algorithm for transmitting voice would be to packetize the digital speech, dropping packets containing only silences, and to send them to an autonomous network interface to be transmitted asynchronously. The simplest algorithm for receiving voice would be to receive packets asynchronously from an autonomous network interface, and begin to play back the first packet of a talkspurt after some artificial delay, with subsequent packets of the talkspurt each immediately following its predecessor.

An important improvement on the transmission algorithm at the source deals with the case when packet transmission must be delayed until the net can be acquired. If, while the packet is waiting to be transmitted, more speech samples are being buffered, these can be appended to the old packet before it is transmitted instead of being used to start a new packet. This approach has three advantages:

1. It tends to transmit fewer packets under a heavy load, thereby applying a degree of negative feedback.

2. The varying length of a packet serves as a sort of time-stamp. Since the last speech sample in the packet was collected just before the packet was successfully transmitted, this allows the receiver to determine the exact age of the first speech sample, allowing more precise control over packet playback.

3. It produces an adaptive effect. In the simplest case, each station will begin to attempt to transmit a new packet one packet-time after beginning to attempt to transmit the previous packet; here, though, this will occur one packet-time after the last packet was successfully transmitted. In the first case, if two stations happen to collide with each other once, they will then collide with each other every packet-time afterwards until one of them begins a silence; in the second case, a collision once resolved creates a phase shift that persists thereafter.

At the receiver, we buffer packets to cope with their variable delay. If the speech samples are implicitly time-stamped by the variable packet size, it is possible to correct for the delay that the first packet of a talkspurt has already experienced in transmission.

The receiver implementation can be quite simple. The voice path is implemented as a first in first out (FIFO) buffer: packets are inserted as they are received while samples are extracted synchronously. The first packet of a talkspurt is preceded in the FIFO by the appropriate amount of artificial silence; the beginning of a talkspurt can be detected by the FIFO being empty. This scheme is easily extended to the case of connections with more than one other speaker, with multiple independent speech sources being merged together; each speaker is assigned a separate FIFO and summing is performed on the outputs of the FIFOs. The FIFOs can be implemented in hardware or software.

Samples that are too late are discarded; they will have been preceded by an artificial silence. Excessive delays can result in packets that are longer than the FIFO, in which case part of the packet can be discarded. For every amount of artificial silence we accidentally introduce, we lose an equivalent amount of speech, except when the last samples of a talkspurt are delayed excessively, in which case the artificial silence before playing them back is matched by losing part of the real silence elsewhere. With proper matching between transmitter and receiver, it is possible for the transmitter to predict which samples the receiver would discard, and simply not transmit these in the first place, thereby reducing net traffic under heavy load and avoiding a potential instability.

V. BASIC CSMA/CD PERFORMANCE

Simulations were performed to measure the voice capacity and related characteristics of CSMA/CD LANs. For a simulation in which voice stations used (nominally) minimum-size voice packets (5.75 ms), and in which there was no data traffic, Fig. 2 shows the transmission delays that voice packets experienced. Note that the delay is essentially zero (i.e., less than the quantizing sample time of 125 μs) until the equivalent of approximately 60 conversations is reached, at which point the delay rises roughly linearly. [While the expected number of speakers at an arbitrary point in the simulation is 0.8 times the number of conversations, the actual number of speakers will vary from this, depending on the details of the traffic pattern. We define the *effective* number of conversations at a point in time as the actual number of speakers divided by 0.8 (the expected value of the effective number of conversations is the actual number of conversations). It was found that much smoother graphs were obtained by plotting transmission performance using the effective number of conversations rather than the actual number, and that the curves were thereby made much more similar across different traffic patterns. Most of the graphs in this paper are based on effective conversations rather than actual conversations; they may be converted to actual conversations by the addition of appropriate axial randomness.] Note that the standard deviation is several times the mean, due to the long tail of the distribution of delays; this is illustrated in Fig. 3, which shows the distribution of delays at a 50-conversations loading.

If we set a threshold of an additional 5.75-ms artificial delay of voice samples, we can bring the total delay through the system to 11.5 ms. (Given a desired total delay of 11.5 ms, it would be possible to allocate less of the total to variable delay and more to packet size; the reverse would also be possible. An extreme position in either direction can be

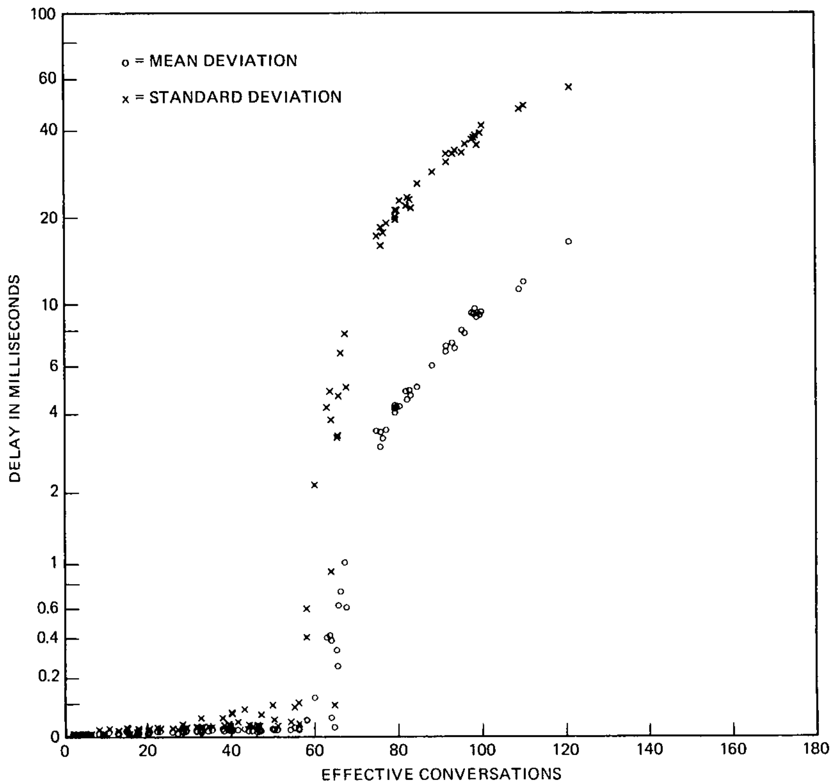

Fig. 2—CSMA/CD delay for 5.75-ms voice packets in the absence of data. This graph shows the mean and standard deviation of the delay experienced in the transmission of (nominally) 5.75-ms (i.e., minimum-size) voice packets in the absence of data traffic, as a function of the number of effective conversations. Note that both the mean and the standard deviation are essentially zero (i.e., less than the quantizing sample time, 125 μs) until about 60 effective conversations are reached, at which point they grow roughly linearly (distorted here by the logarithmic vertical scale) and become quite large; the standard deviation far exceeds the mean.

counterproductive, so an equal division is not totally unreasonable. However, as will be shown later in this paper, it seems more optimal, for CSMA/CD networks, to allocate significantly more delay to packet size than to variable delay.) At this delay we can expect to transmit up to about 60 conversations well, and to lose some speech samples past that point, as shown in Fig. 4. Here, the vertical axis measures the percentage of speech samples lost, which roughly models the degradation of the channel. Further study is needed to determine the effect of other parameters of the artificial silences and loss speech upon human users: for example, the number and length of these anomalies are probably important.

As a test of validity, the results of these simulations (as well as the

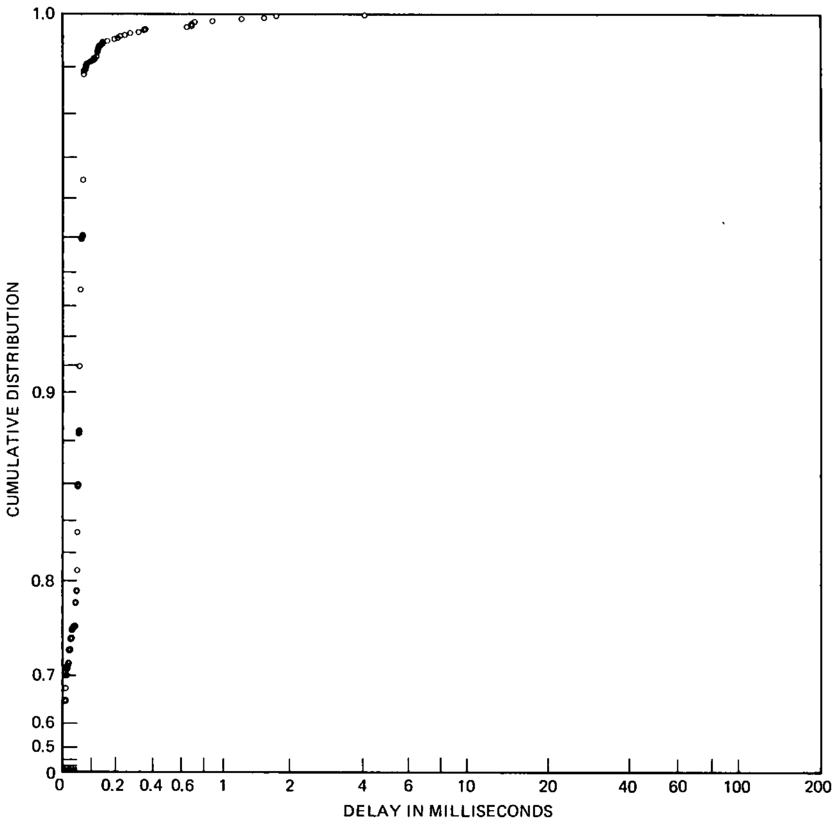

Fig. 3—CSMA/CD cumulative distribution of voice packet transmission delays for a loading of 50 conversations and 5.75-ms packets. This graph shows the cumulative distribution of the variable transmission delays experienced over a period of 0.5 second when the simulated CSMA/CD network was loaded with 50 conversations and no data traffic. The vertical axis is exponential; the horizontal axis is logarithmic. We see that about 65 percent of the packets experienced no delay, that over 99 percent were transmitted in less than 125 μs (the quantum phase shift possible using the adaptive algorithm), and one took over 4 ms. The shape of this curve causes the standard deviation to exceed the mean, as shown in Fig. 2.

ones following) were compared to a previous study of voice transmission on CSMA/CD networks[4]; the results were found to correspond closely.

As an example of the importance of the adaptive nature of the variable packet-size algorithm, Fig. 5 shows the effect of using fixed packet sizes; we see that the channel degrades much sooner.

VI. EFFECT OF DATA TRAFFIC ON CSMA/CD CAPACITY

As we have seen, the natural synchronous nature of voice traffic in conjunction with an adaptive algorithm enables the transmitters, in

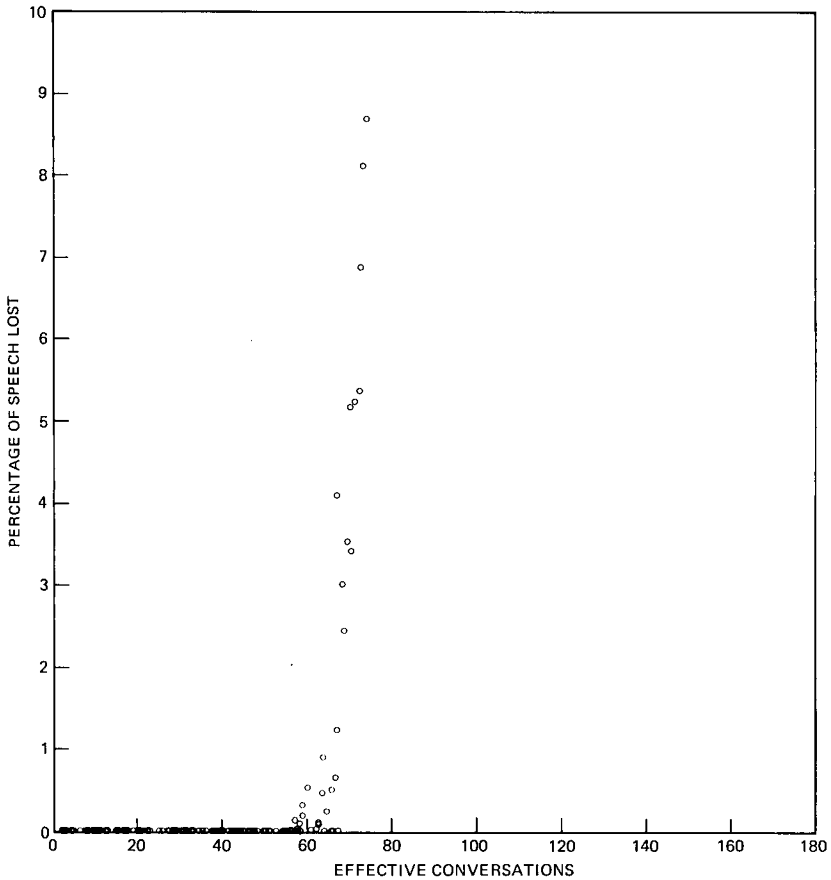

Fig. 4—CSMA/CD voice channel degradation with increasing load with 5.75-ms packets and 5.75-ms artificial delay. This graph shows the voice signal degradation experienced on a simulated CSMA/CD network with 5.75-ms packets and an additional 5.75-ms artificial delay at the receiver. Two simulations with different voice traffic patterns were performed and their results superimposed. Degradation is measured as the percentage of voice samples that are discarded (here at the transmitter). There is no degradation until about 58 effective conversations, soon after which the degradation rises roughly vertically: the network is saturated and each new conversation causes a conversation's worth of speech samples to be lost.

effect, to slot themselves and thereby interfere only minimally with each other. As the traffic increases, though, talkspurts begin to arrive faster than they can settle in and this structure begins to disintegrate. It is therefore to be expected that the addition of data traffic, with its inherent asynchronous nature, will interfere with the voice traffic more than its share, so that the addition of some amount of data traffic will eliminate more than an equivalent amount of voice traffic capacity.

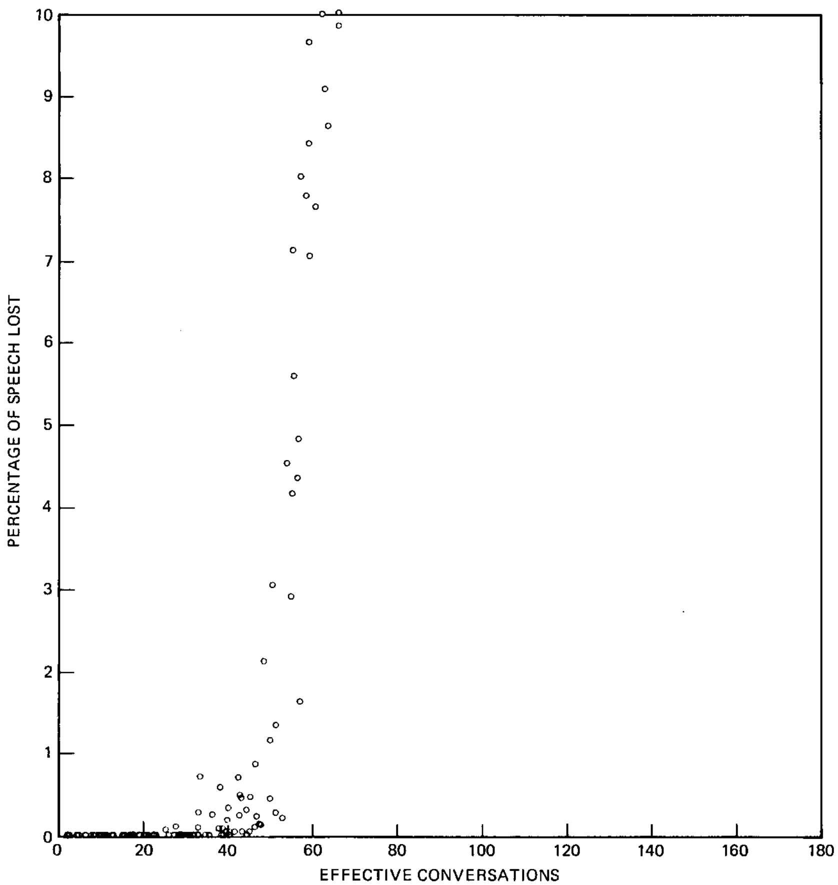

Fig. 5—CSMA/CD voice channel degradation with increasing load with 5.75-ms packets and 5.75-ms artificial delay, with a nonadaptive algorithm. This graph shows the same relation between simulated network load and voice channel degradation as does Fig. 4, except that it uses a simple nonadaptive transmission algorithm. As we see, the expected performance of such a system is significantly poorer than one with the adaptive algorithm, in regard both to the point at which degradation begins and the point at which the curve becomes essentially vertical.

This phenomenon does in fact occur. Figure 6 shows the delay experienced by voice packets on an CSMA/CD system with 5-percent data loading. Note that there is no longer any region of essentially zero delay, as in Fig. 2 without data loading, and that the knees of the curves, although significantly less well-defined here, certainly occur more than 5 percent sooner than earlier. Figure 7 shows the channel degradation allowing 5.75-ms buffering at the receiver.

Additional simulation results, not shown here, were obtained for 10-percent data loading; they basically extend this trend.

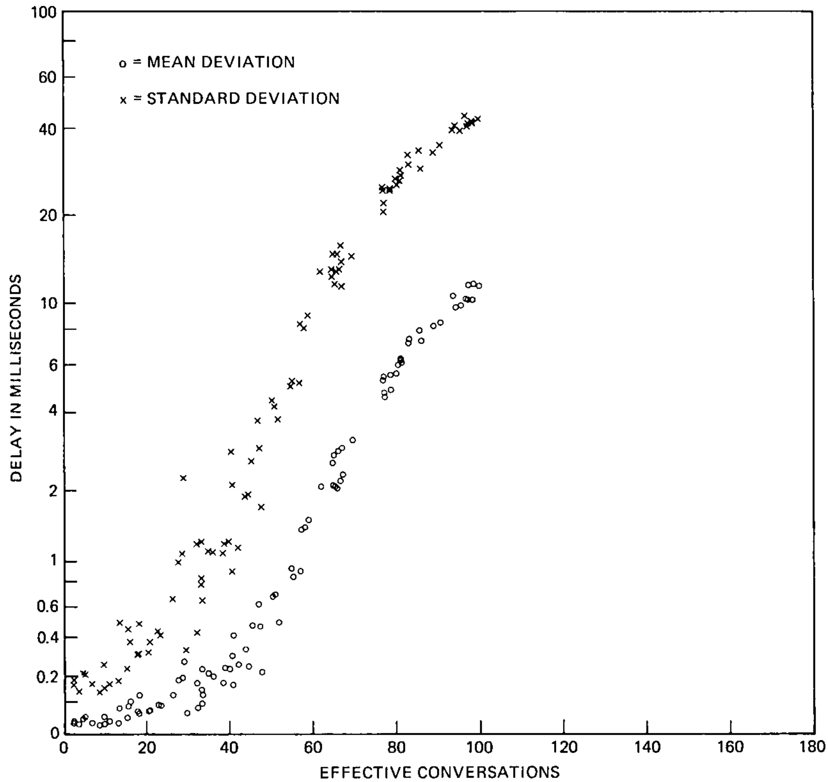

Fig. 6—CSMA/CD delay for 5.75-ms voice packets with 5-percent data loading. This graph shows the effect of a 5-percent data traffic loading on the mean and the standard deviation of the voice packet delay on the simulated CSMA/CD network; it should be compared with Fig. 2, where no data loading was assumed. Notice that there is no longer any large region of essentially zero delay, and that the 5-percent data loading has shifted the curves to the left far more than 5 percent.

VII. INCREASING THE DELAY IN A CSMA/CD SYSTEM

We can increase the effective bandwidth of the system by increasing the packet size at the transmitter or by increasing the variable delay threshold at the receiver. If we choose a relatively large value of 50 ms for each, resulting in a 100-ms total delay through the system, we find that, as shown in Fig. 8, about 150 effective conversations can take place on the *Ethernet* CSMA/CD network in the absence of data. Assuming 5-percent data loading reduces this number to about 125 effective conversations, as shown in Fig. 9.

It seems likely that the point of diminishing returns has been reached at the 50-ms level; further increases in the packet size or receiver delay cannot produce any great increase in the capacity of the network, but they could subjectively degrade the channel by increasing its delay.

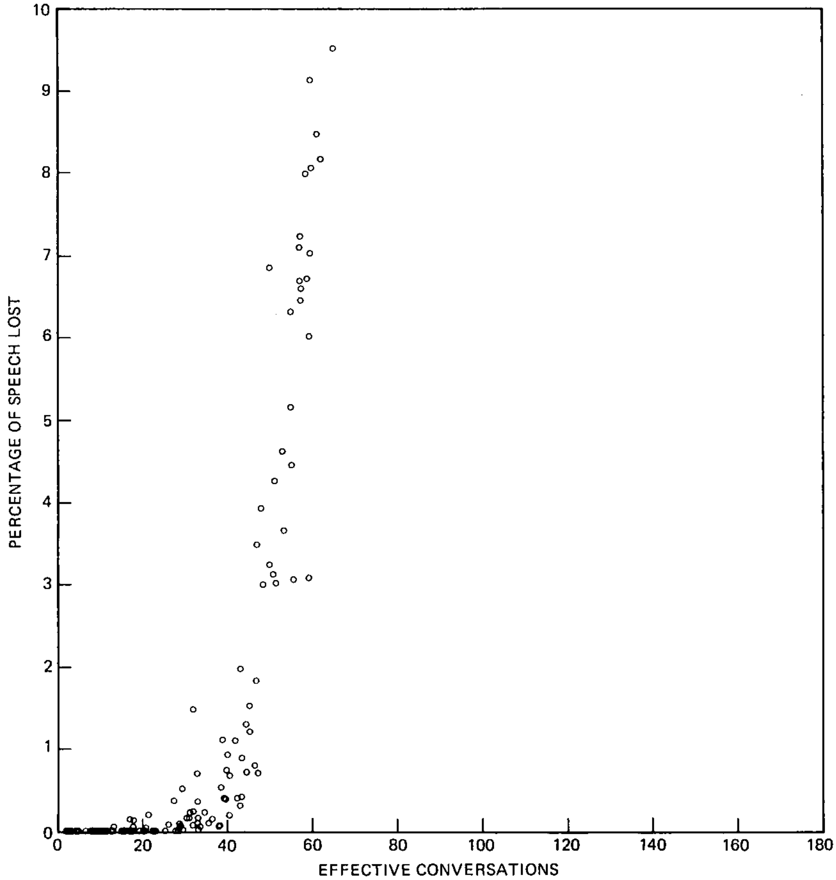

Fig. 7—CSMA/CD voice channel degradation with increasing load with 5.75-ms packets and 5.75-ms artificial delay, with 5-percent data loading. This graph shows the signal degradation on simulated voice channels over a CSMA/CD network in the presence of 5-percent data loading; it should be compared with Fig. 4, in which there was no data loading. Again, two simulations were performed. Degradation rises significantly earlier than with no data loading; there is more than a 5-percent degradation in the effective bandwidth. The effect of a 5-percent data loading on a system with a nonadaptive fixed packet size (not shown here) is comparatively less, since it does not take advantage of the synchronous nature of the voice packets.

VIII. A TOKEN BUS

An additional simulation study was performed to determine the suitability of a *token-passing* LAN for carrying voice. In a token-passing LAN, contention is resolved through use of a conceptual circulating token. A station may transmit only if it has possession of the token, and must then pass the token to the next station in logical sequence. A *token ring* is a token-passing LAN with a physical ring

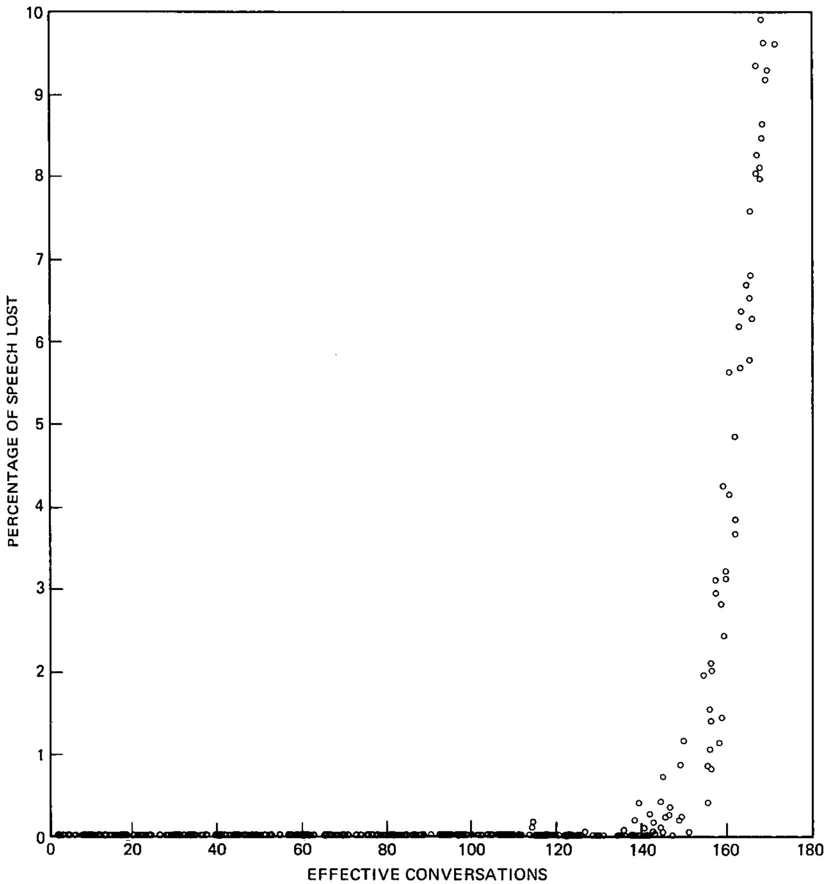

Fig. 8—CSMA/CD voice channel degradation with increasing load with 50-ms packets and 50-ms artificial delay. This graph shows the signal degradation experienced on simulated voice channels over a CSMA/CD network with 50-ms packets and an additional 50-ms artificial packet delay at the receiving station; it should be compared with Fig. 4, which assumes smaller numerical values. As in Fig. 4, two simulations were performed and their results superimposed. We see that a large increase in the delay through the system can produce a significant increase in its effective bandwidth.

topology; the logical sequence is typically the same as the physical sequence of stations on the ring. A *token bus* is a token-passing LAN with a physical bus topology (linear or acyclic branching); the logical sequence can often be arbitrary but is most efficient if it corresponds to the physical sequence.

A token bus was chosen as the token-passing LAN most directly comparable with a CSMA/CD LAN, and the numerical parameters of the token bus were chosen to be as similar as possible to those of the *Ethernet* specifications and the choices of the above CSMA/CD sim-

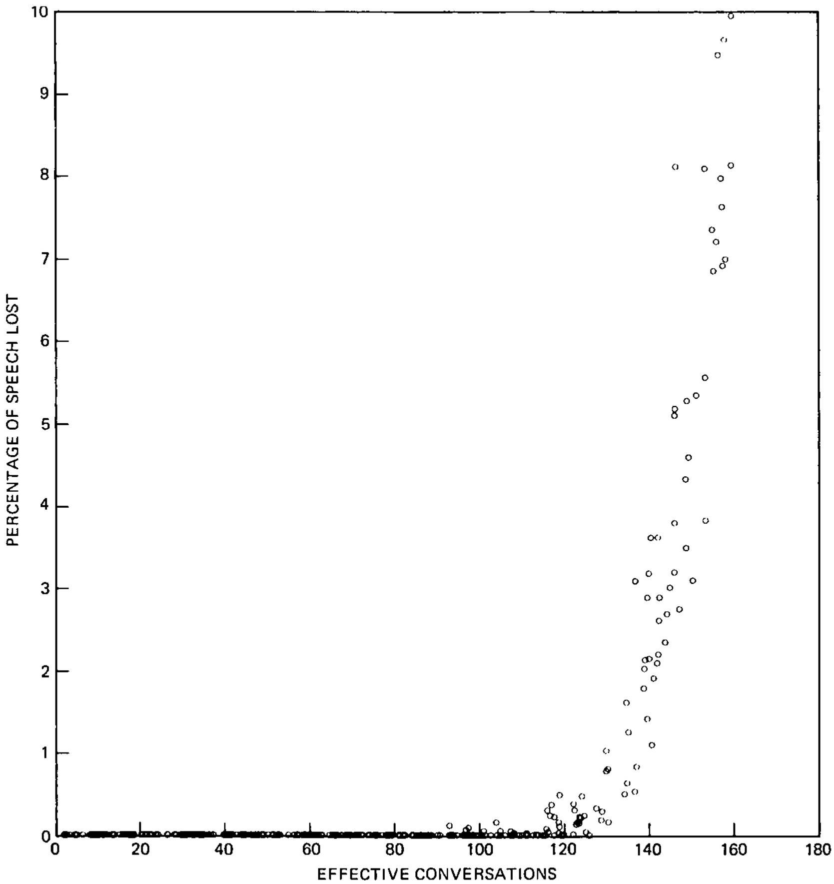

Fig. 9—CSMA/CD voice channel degradation with increasing load with 50-ms packets and 50-ms artificial delay, with 5-percent data loading. This graph shows the signal degradation experienced on simulated voice channels over a CSMA/CD network with a long delay through the system, in the presence of 5-percent data loading; it should be compared with Fig. 8, in which there was no data loading. Again, two simulations were performed and their results superimposed. Note that the proportionate drop in effective bandwidth caused by the data traffic is much less than when small delays were considered, as in the difference between Figs. 4 and 7.

ulations. These choices are quite possibly far from optimal for a token-passing LAN, but they allow for a simple comparison with the CSMA/CD results; there is no typical design for token-passing systems that corresponds to *Ethernet* among CSMA/CD systems.

The simulated token bus LAN has a single token circulating; when a station receives the token, it either transmits a packet, which implicitly passes the token to the next station in logical sequence, or it transmits an abbreviated packet, containing only a header, to explicitly pass the token. The bus is linear, of maximum *Ethernet* length, with stations uniformly distributed. No attempt is made to match the token-passing sequence to the physical sequence of stations on the bus, or to model the (small) control traffic needed to expand the sequence when new conversations, and their associated stations, are added.

The voice packet delays in the token bus simulation are shown in Fig. 10. [To understand the strange shape of the curves in Figure 10, consider a simplified case. We transmit (nominally) 5-ms packets. There are enough stations in the ring for the token to require 2 ms to circulate in the absence of any transmissions. Assume that for each

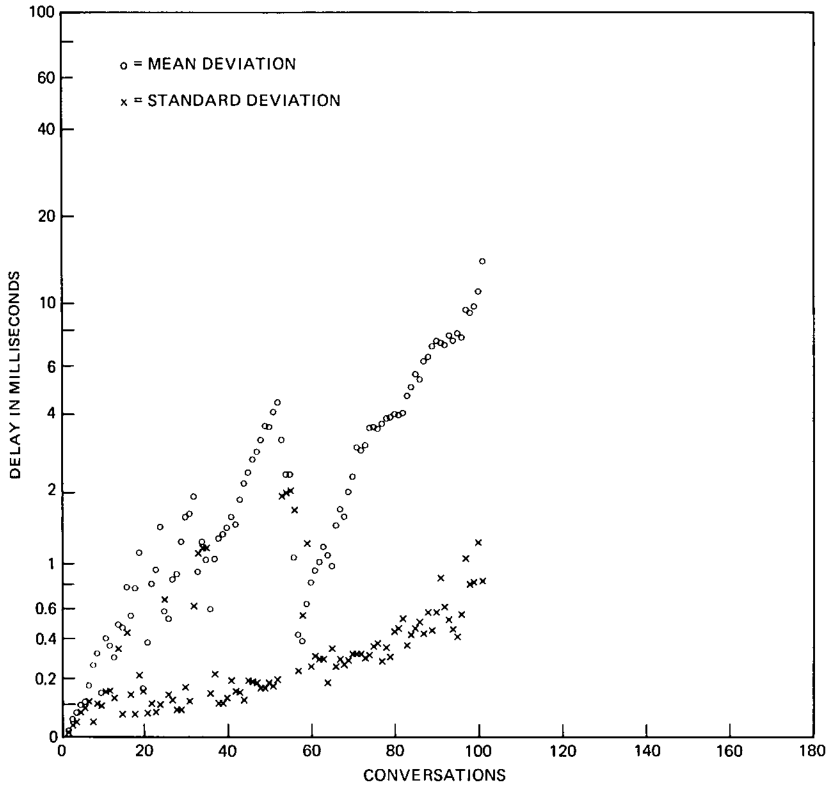

Fig. 10—Token bus delay for 5.75-ms voice packets in the absence of data. This graph shows the mean and standard deviation of the transmission delay on a simulated token bus with no data; it should be compared with Fig. 2, which shows the corresponding delay for a CSMA/CD network. The mean delay is never less than for the CSMA/CD case; the standard deviation under heavy load is much less than for the CSMA/CD case but greater under light load. The sawtooth shape of the curves show that these figures are nonunique and depend on the transmission history. The horizontal axis measures actual conversations instead of effective conversations since even silent stations take part in token circulation.

active station (one associated with an active speaker) to transmit its 5-ms packet each time around would require an additional 4 ms total. An active station will be ready to transmit 5 ms after it has last transmitted, but if every station transmits every time around, the token will take (over) 6 ms to circulate, and so every station will experience the same (over) 1-ms delay; this can remain as constant as the load on the net. On the other hand, if the token were circulating faster, so that it needed only 4 ms for its transit, then a station would transmit only every other time around and experience a delay of 3 (4+4-5) ms. If stations transmitted only every other time around, the time needed under the original assumptions for a token cycle will be 2+4/2 = 4 ms, as assumed; this shows that the performance of a

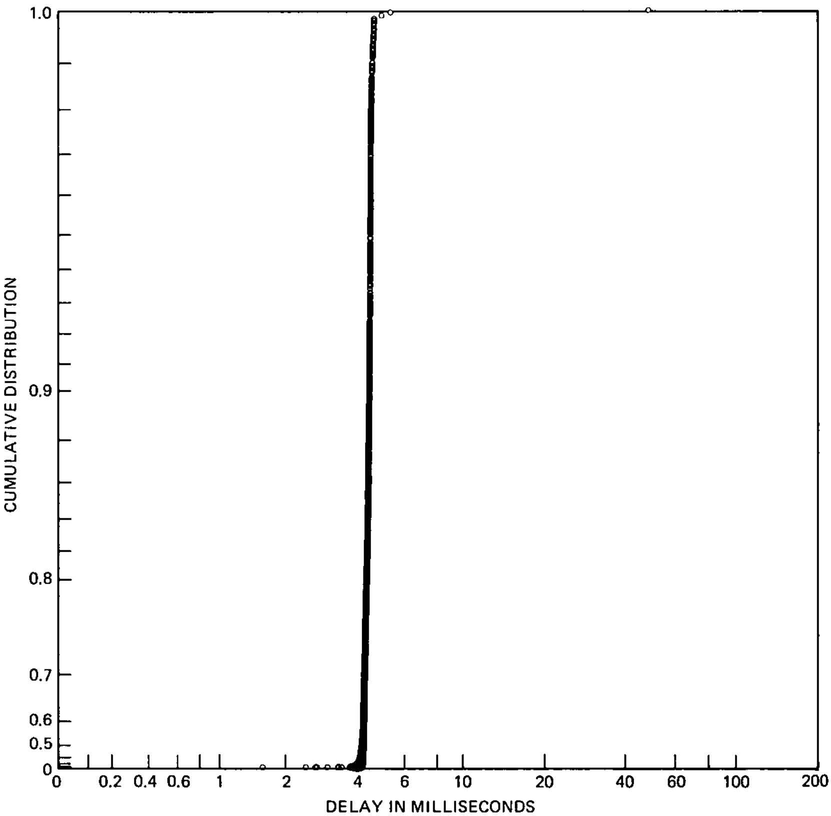

Fig. 11—Token bus cumulative distribution of voice packet transmission delays for a loading of 50 conversations and 5.75-ms packets. This graph shows the cumulative distribution of the variable transmission delays experienced on a simulated token bus loaded with 50 conversations and no data traffic; it should be compared with Fig. 3, which shows the equivalent case for a simulated CSMA/CD network. We note that almost all packets are delayed essentially the same amount of time, which reflects an essentially constant token circulation rate during this period.

token-passing system can be nonuniquely determinable from the load, and can therefore depend upon history.] We note that the mean delay is never less than the mean delay for the corresponding CSMA/CD case shown in Fig. 2. However, the standard deviation for a token ring under sufficient load is much smaller than for the CSMA/CD network: all packets experience very similar delays, as shown in Fig. 11.

Adding 5-percent data loading to a token bus increases the delays to those shown in Fig. 12. Again, the mean is never less than the mean for the CSMA/CD case shown in Fig. 6, but the standard deviation is much smaller under heavy load.

To allow a more direct comparison, Fig. 13 shows the capacity of a CSMA/CD network as a function of the amount of buffering at the receiver, with 5.75-ms packets and no data loading, and allowing 1-

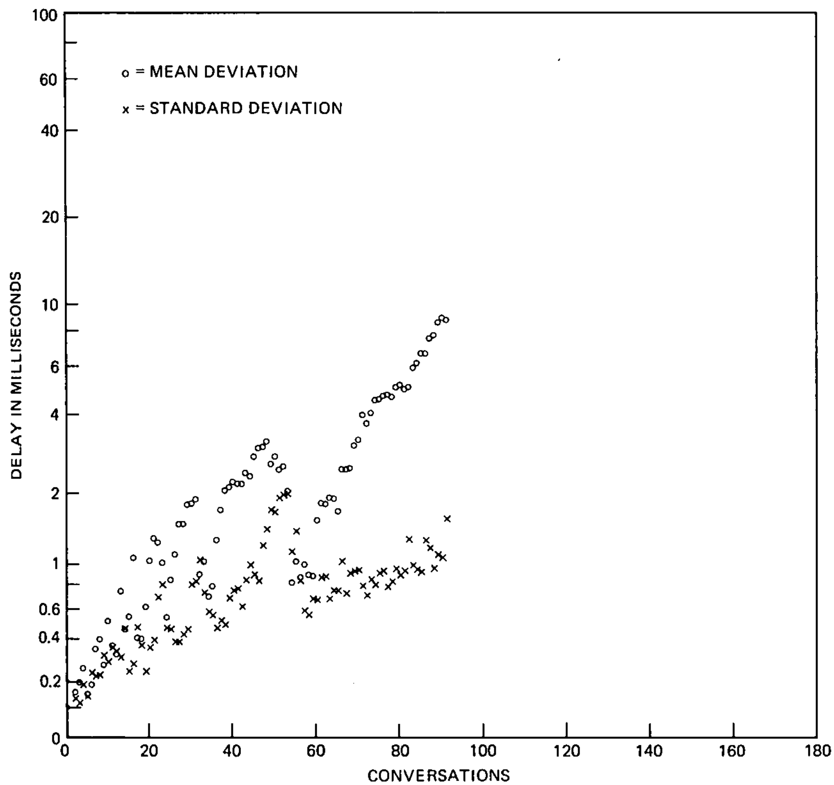

Fig. 12—Token bus delay for 5.75-ms voice packets with 5-percent data loading. This graph shows the mean and the standard deviation of the delay experienced in the transmission of voice packets on a simulated token bus with 5-percent data loading; it should be compared with Fig. 6, which shows the corresponding delay for a CSMA/CD network. Note that the mean delay is never less than for the CSMA/CD case; the standard deviations for a token bus are significantly less than for CSMA/CD under heavy load, although they are greater under light load.

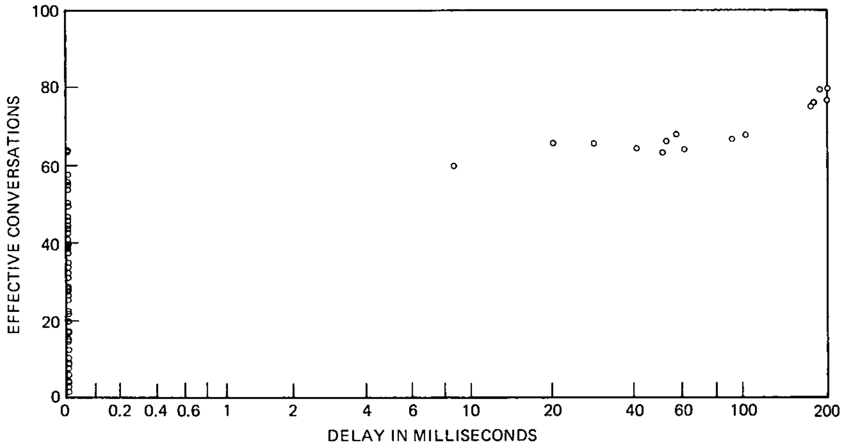

Fig. 13—CSMA/CD capacity as a function of receiver buffering delay, with 5.75-ms packets, 1-percent sample loss, and no data loading. This graph shows the capacity, measured in effective conversations, of a simulated CSMA/CD network as a function of the buffering delay at the receiving station, with (nominally) 5.75-ms packets and allowing up to 1-percent of the speech samples to be lost (at the receiver), in the absence of data. We note that the capacity depends very little on the buffering at the receiver; this suggests that, of some total allowable delay through the system, more delay should be allocated to packet length than to receiver buffering.

percent speech sample loss. No compensation at the transmitter for the buffering at the receiver, in the form of locally discarding samples that would otherwise simply be discarded remotely, was performed in these simulations. Figure 14 shows CSMA/CD capacity with 5-percent data loading. By contrast, Figs. 15 and 16 show the corresponding relations for the token bus with no data loading and with 5-percent data loading, respectively. Figures 13 through 16 show the token bus to offer significantly more capacity than the CSMA/CD network, suggesting that a token-passing network is superior to an CSMA/CD network for voice transmission. We can see that the CSMA/CD LAN's performance is much less dependent on the receiver delay than is that of the token ring, suggesting that the increase in the overall delay caused by increasing the receiver buffering would be better spent in increasing the packet length while keeping the receiver delay relatively small. On the other hand, a token bus can efficiently keep a relatively small nominal packet size and benefit directly from an increase in receiver buffering, as shown.

IX. CONCLUSIONS

It is possible to transmit a large number of voice conversations on either a CSMA/CD LAN or a token-passing LAN in the presence of

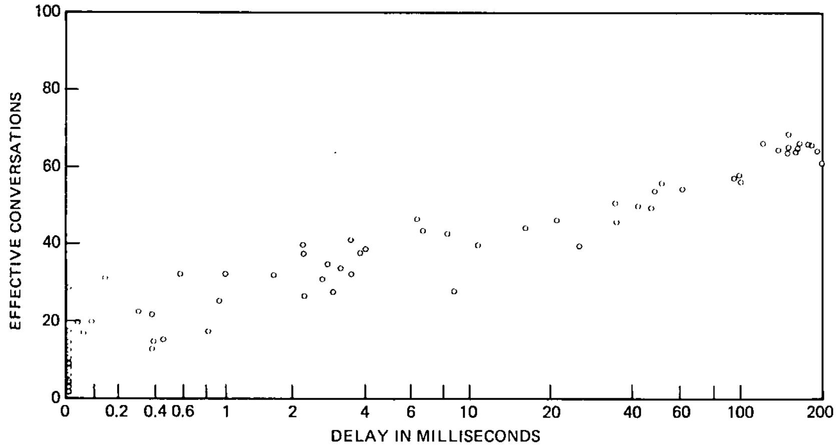

Fig. 14—CSMA/CD capacity as a function of receiver buffering delay, with 5.75-ms packets, 1-percent sample loss, and 5-percent data loading. This graph shows the capacity, measured in effective conversations, of a simulated CSMA/CD network as a function of the buffering delay at the receiving station, with (nominally) 5.75-ms packets and allowing up to 1-percent of the speech samples to be lost (at the receiver), with 5-percent data loading. As in Fig. 13, which considered the corresponding case with no data loading, the capacity depends fairly little on the amount of buffering at the receiver, again suggesting that receiver buffering should be kept fairly small and its share of the overall delay used in allowing the packet size to grow.

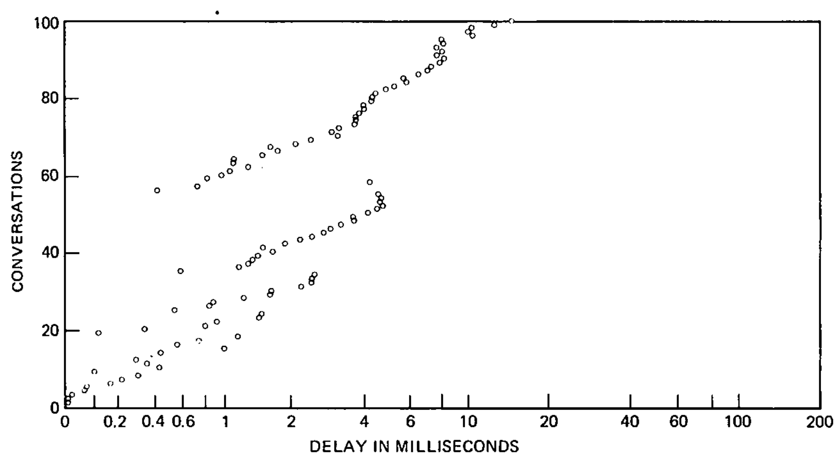

Fig. 15—Token bus capacity as a function of receiver buffering delay, with 5.75-ms packets, 1-percent sample loss, and no data loading. This graph shows the capacity, measured in actual conversations, of a simulated token bus as a function of the buffering delay at the receiving station, with (nominally) 5.75-ms packets and allowing up to 1-percent of the speech samples to be lost (at the receiver), in the absence of data. The significant increase in capacity with increased receiver buffering, plus some reasoning on the nature of token-passing, suggest that the total delay through a token bus system should be allocated predominantly to receiver buffering, with relatively small nominal packet sizes.

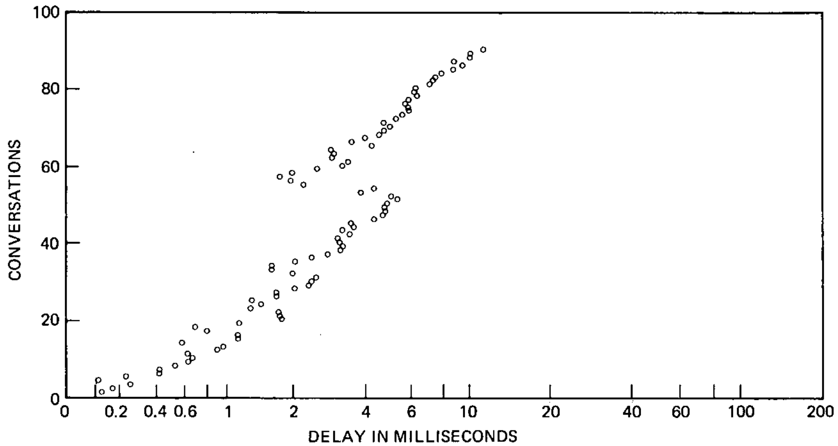

Fig. 16—Token bus capacity as a function of receiver buffering delay, with 5.75-ms packets, 1-percent sample loss, and 5-percent data loading. This graph shows the capacity, measured in actual conversations, of a simulated token bus as a function of the buffering delay at the receiving station, with (nominally) 5.75-ms packets and allowing up to 1-percent of the speech samples to be lost (at the receiver), with 5-percent data loading. As in Fig. 15, which considered the corresponding case with no data loading, the capacity depends significantly on the amount of buffering at the receiver, again suggesting that receiver buffering in a token bus system should be kept fairly large and the nominal packet size fairly small.

reasonable data loading. The performance of token-passing seems superior to that of CSMA/CD.

There are even better mechanisms for transmitting digital voice: time-division multiplexing schemes, for example, can do an excellent job for voice, but are not exceptional for carrying data because of their inherently synchronous nature. Similarly, CSMA/CD LANs can be superior to token-passing for many data applications. It is still a research problem to find an LAN that can carry both voice and data "optimally", or to identify more exactly the appropriate trade-offs.

One significant unanswered question is the potential of such a system for serving large numbers of users; there is an inherent limit of the number of users on one LAN. It is uncertain to what extent internetworking can help, since internetworking would increase the mean and standard deviation of the delays.

REFERENCES

1. *The* Ethernet/*A Local Area Network/Data Link Layer and Physical Layer Specifications,* Digital Equipment Corporation, Intel Corporation and Xerox Corporation, Version 1.0, September 30, 1980.
2. J. F. Shoch and J. A. Hupp, "Measured Performance of an Ethernet Local Network," Comm. ACM, *23,* 9 (December 1980), pp. 711–21.
3. P. T. Brady, "A Statistical Analysis of On-Off Patterns in 16 Conversations," B.S.T.J., *47,* 1 (January 1968), pp. 73–91.
4. G. J. Nutt and D. L. Bayer, "Performance of CSMA/CD Networks Under Combined

Voice and Data Loads," IEEE Trans. Commun., *COM-30,* 1 (January 1982), pp. 6–11.

AUTHOR

John D. DeTreville, B.S. (Mathematics), 1970, University of South Carolina; S.M. (Computer Science), 1972, Massachusetts Institute of Technology; Ph.D. (Computer Science), 1978, Massachusetts Institute of Technology; AT&T Bell Laboratories, 1978—. Mr. DeTreville's first work at AT&T Bell Laboratories was on 5*ESS*™; he moved to Murray Hill in 1980, where his work has incorporated topics in distributed systems and program synthesis. Member, Association for Computing Machinery.