

Voice Integration in a TCP/IP/Ethernet based LAN

Joaquim Azevedo, Joaquim Arnaldo Martins, Pedro Guedes de Oliveira

Resumo- Neste trabalho é feito um estudo experimental de um sistema de comunicação de voz a 64 kbit/s, com base em computadores pessoais, sobre a rede local Ethernet. A integração de voz e dados é alcançada sem necessidade de recorrermos a qualquer novo protocolo para comutação de pacotes e sem quaisquer alterações às especificações do protocolo CSMA/CD. O desempenho do sistema é avaliado segundo o atraso médio de transmissão e percentagem de pacotes perdidos em função da carga na rede, do tamanho dos pacotes, do tamanho dos *buffers* e das características do tráfego. Apresentamos, ainda, uma estimativa do número máximo de canais de voz, sem degradação significativa da inteligibilidade do sinal reconstruído. É feita uma referência ao problema da sincronização em sistemas de voz com comutação de pacotes sendo sugerido um método para eliminar o *clock drift*. É proposto um algoritmo de reconstrução sem perdas para um sistema sem detecção de silêncio. Finalmente, é apresentada uma implementação de um canal de voz sobre o protocolo TCP/IP.

Abstract- This work presents an experimental study on a personal computer based implementation of a 64 kbit/s voice communication system over the Ethernet local network. Voice/data integration is achieved without resorting to any new packet switching protocol and without any changes to the CSMA/CD protocol specifications. System performance is evaluated according to its average transmission delay and percentage of packet loss as a function of network load, packet size, buffers size, and traffic characteristics. An estimate of the maximum number of voice channels that can be supported without significant degradation of the reconstructed signal is also presented. A reference is made to synchronisation problems in packet voice systems and a method to eliminate clock drift is suggested. A non packet discarding reconstruction algorithm for a system with no silence detection is proposed. Finally, a voice channel implementation over TCP/IP protocols is presented.

I. INTRODUCTION

Lately, great efforts have been made in order to integrate voice and video into packet switching networks. Many of these are intended to integrate real-time voice and data in IEEE 802.3-5 local computer networks, some of which involve simulations [1-6] and others experimental implementations [7-9].

Future trend is to integrate voice, data, and video into a single workstation. Broadband networks (ex: FDDI,

B-ISDN) are expected to support these multimedia applications, however, the idea of using Ethernet networks as transmission medium should not be put aside. They represent a low error with reasonable bandwidth, cheap and widely spread transmission medium for communications within buildings and other institutions (ex: universities, research centres, industrial manufacturing sites, etc). On the other hand, some of these networks, which today handle only data, have a low occupation rate. Finally, with Ethernet's migration from 10 to 100 Mbit/s [10], it is expected that a greater number and more bandwidth demanding applications can be supported.

In the sections which follow, we review voice traffic characteristics and the Ethernet local network, describe an experimental prototype design and implementation, measure system performance when only voice traffic is present, and then measure its behaviour in an integrated voice/data environment.

II. TRANSMITTING REAL TIME VOICE ON ETHERNET

A. Voice traffic characteristics

Digital speech, with telephone quality at 64 kbit/s, is achieved by sampling voice signal at 8 kHz, 8 bits per sample. Speech consists of alternating segments of speech followed by silence, with each speaker, in a typical conversation, talking only about 40% of the time and being silent for the rest [1]. This characteristic can be exploited in order to achieve greater system performance by silence suppression. The primary requirement for voice traffic is its real-time treatment. Voice samples which are not delivered within some previously well established period are rejected, resulting in speech loss. Owing to speech redundancy and attending to human audition characteristics, losses of 1 or 2% are acceptable without incurring significant quality degradation [2, 11].

Transmission/reception of digital voice through a packet switched network is simple, but leads to inevitable delays. This delay has a fixed and a variable component. Packet generation time, time spent in reception queues, and other minor fixed delays (ex: transmission time) contribute for the fixed component. The variable component is essentially due to the delay in obtaining the transmission medium and is highly dependent on the access protocol. Various efforts have been made to establish a limit for the

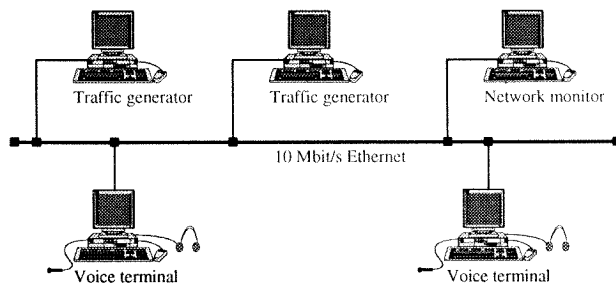


Fig. 1 - System configuration used during the experimental study.

maximum acceptable delay but, due to the subjective nature of the matter, there is no unanimous opinion globally accepted. For conversations within a LAN, delays between 100 and 200 ms are tolerable [12].

Another important issue in packet voice systems is the amount of speech present in each packet. Large packets increase effective bandwidth by reducing the probability of collisions in CSMA/CD based systems. However, it increases the delay's fixed component and the amount of speech lost per packet may become unacceptable. Packet length must result from a compromise between overall delay, system performance, and the amount of speech lost per packet.

B. Review of Ethernet's principles of operation

Ethernet uses *carrier sense multiple access with collision detection* (CSMA/CD) to allow multiple stations to share a single broadcast bus. If a transmitting station detects a collisions, it immediately aborts transmission, sends a jam sequence, and then reschedules its packet for transmission at a random time calculated according to a truncated binary exponential backoff algorithm. Detailed descriptions of these protocols may be found in [13, 14]. CSMA/CD based networks are known to give equal access opportunities to all stations. In environments where voice and data coexist, this is obviously not the best approach. In the presence of data traffic, voice stations should have priority over data stations.

III. PROTOTYPE DESIGN AND IMPLEMENTATION

A. System configuration

In order to further understand the problems involving a synchronous packet voice communication system over a non deterministic multiple access network, a prototype, that implements a 64 kbit/s full-duplex voice channel, was developed. It currently consists of two voice terminals, two traffic generators, and a network monitor as illustrated in Fig. 1.

Each subsystem is individually implemented on a PC and, in each, the network interface board used was the WD8003E [15]. This board contains an 8 Kbyte on-board PC-addressable memory buffer, and is compatible with IEEE 802.3/Ethernet standards. The voice board used was

the Sound Blaster Pro version 2.0. The Traffic Generator as well as the Network Monitor, used exhaustively during this experimental study, belong to a set of network traffic analysis tools developed at our University [16, 17].

B. System operation

The functional basis of a full-duplex communication system are elementary. A conversation consists of two independent and identical simplex voice channels, each of which contains a packet voice transmitter and a packet voice receiver. In the remainder of this section, we describe the principles of operation of each of these subsystems. We also discuss briefly the data traffic generator and network monitor. Finally, some attention to the problem of clock drift shall be given.

1) *Packet Voice Transmitter*: Speech is digitized at a rate of 8 kHz using 8 bits/sample. Samples are gathered into an intermediate transmission buffer until a packet is filled, after which it is transferred to the network interface buffer and submitted for transmission. Although Ethernet specifies 16 as the maximum number of collisions a packet may suffer, any packet submitted for transmission, that experiences multiple collisions, is continuously rescheduled until it is either successfully transmitted or a new packet of voice samples is collected. If the latter occurs, the older packet is discarded and the new one is transmitted.

The intermediate transmission buffer is implemented with a circular FIFO (First In First Out) buffer with capacity for two voice packets. This buffer is handled by two pointers. A write pointer points to where the next packet to be acquired is to be put and the other, a read pointer, keeps track of the next packet to be sent to the network interface buffer.

At this stage, silence suppression has not yet been implemented. No error recovery mechanisms were contemplated due to real-time constraints. Suppose the receiving station detects that something went wrong (ex: a voice packet is missing); it would then be too late to formulate a request for packet retransmission. We rely on the knowledge of the good transmission characteristics of the Ethernet to achieve good results. The only additional information a voice packet contains is its sequence number. It is used, by the receiver, for statistical purposes and as a means for detecting lost packets.

2) *Packet Voice Receiver*: Packets whose destination address is that of the receiving station, and are identified as voice packets, are transferred from the network interface buffer to an internal receive buffer. Voice samples are then taken from the internal receive buffer and played back through the voice board's DAC (digital to analog converter). Voice packets are generated and submitted for transmission at regular intervals. The mechanism responsible for speech reconstruction, at the receiver, must compensate for variable delay by inserting a certain delay D before playing back each packet. This delay is limited by the maximum end-to-end delay, D_{max} ,

and by the maximum percentage of packets that can be lost at the transmitter or rejected by the receiver.

There are essentially two methods for speech reconstruction in a packet voice system [18, 19]. The first, and most simple, makes use of no timing information to determine variable packet delay as it traverses the network. The receiver introduces a fixed artificial delay D before the reproduction of the first packet of a conversation, if silence detection is not implemented, or before the first packet of each talkspurt, if silence detection is implemented, and the following packets are played back at intervals of D_g (packet generation time) after the first. If a packet is not present when it is scheduled for playback, it is discarded and a silence packet is played back instead, as shown in Fig. 2 a). The choice of D results from a trade-off between the percentage of lost packets and the maximum allowable end-to-end delay. Increasing D reduces packet loss but increases the overall delay as well as the receiver's intermediate buffer size. This method is suitable for environments in which the variance of the delay's variable component is small.

The second approach makes use of timing information, in the form of timestamps, to determine each packet's delay through the network. Montgomery [18] suggests that the first method is more appropriate for intra LANs applications where variable delay as well as other fixed delays are relatively small. For inter LANs applications, where variable delays may be rather significant, this method no longer produces satisfactory results and a method with more timing information is needed.

The reconstruction process, here presented, is based on the first method with some slight modifications. As before, the first packet of each conversation is delayed by D and the others are played back at intervals D_g after the first. D is here implemented by starting the speech reproduction mechanism after the arrival of the second, third, ..., n^{th} packet (depending on the delay D to be introduced). In opposition to the first method, if a packet arrives at the receiver after its scheduled playback time it is accepted and is immediately sent for reproduction, Fig. 2 b). We believe that for small mean variable delays with equally small variance, it is best to accept the late coming packets for playback than to reject them, this because the silence interval introduced due to the receiver's intermediate

buffer underflow is significantly smaller than D_g . The advantage of this method is that, in many cases, the listener will not become aware of speech interruption. It is Musser's opinion [6] that a long silence gap is very disconcerting to the listener and that absolute silence causes the listener to immediately detect that something is missing.

A second variant of this non discarding reconstruction algorithm was implemented: a late arriving packet is scheduled for playback D_g after receiver's intermediate buffer underflow, as can be seen in Fig. 2 c). This method has the advantage of adjusting D every time the receiver's intermediate buffer becomes empty. It acts as though the reproduction mechanism is restarted with D' as the new artificial delay. The interest in restarting the reproduction mechanism is related with the fact that being D an estimate of the delay encountered by the packets before speech reproduction is initiated, it could be a worst case assumption. This reconstruction method allows regular revisions of this delay and proved to be more robust than the previous method. All results presented were obtained using the second method and speech reproduction initiated after the arrival of the second packet.

3) *Clock Drift*: In any CSMA/CD based communication system there are two unavoidable causes responsible for variable packet spacing: transmission delay and clock drift. Variable transmission delay, caused by CSMA/CD access protocol, is intimately related to the number of collisions a packet suffers before it is correctly transmitted. Only for relatively high loads does transmission delay become dominant.

Clock drift is a less obvious cause for losing speech samples and is always present, no matter what the load on the network. If the receiver's clock runs slightly slower than that of the transmitter, packets will accumulate at the receiver's buffer and it will eventually overflow. In the opposite case, if the receiver's clock runs slightly faster than that of the transmitter, the receiver's buffer will underflow and breaks will be introduced in the conversation. To bypass this problem, we introduced feedback into the receiver's reproduction frequency. It consists on varying the reproduction frequency between a high and a low threshold. As an example, suppose that speech reproduction is initiated after the arrival of the second packet, the lower threshold (NPktMin) is 1, the higher threshold (NPktMax) is 3, and the initial reproduction frequency is f_{si} . Every time a new packet is scheduled for playback, the number of packets present in the receiver's buffer is compared to the lower and higher thresholds. If it is smaller than or equal to NPktMin the new reproduction frequency is set to:

$$f_{sn} = f_{si} - Df, \quad Df \in [0, Df_{\max}]$$

On the other hand, if it is greater than NPktMax the new reproduction frequency is set to:

$$f_{sn} = f_{si} + Df, \quad Df \in [0, Df_{\max}]$$

This approach, besides solving the clock drift problem, introduces, as side effect, another interesting behaviour. If there is a sudden raise of network traffic, packet

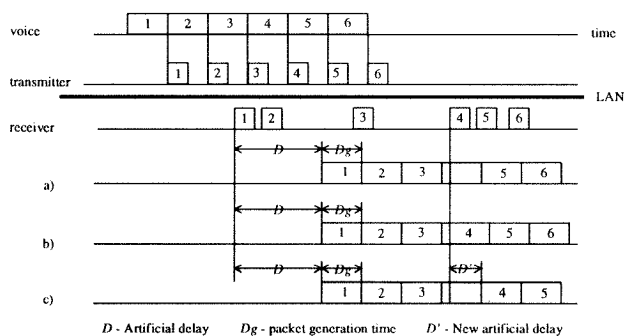


Fig. 2 - Speech reconstruction methods using no timing information.

transmission delay increases. This results in the receiver's buffer being gradually emptied. However, as soon as the number of packets in the buffer goes below NPktMin the reproduction frequency also decreases, thus diminishing the increased packet arrival delay.

The above procedure did indeed eliminate the clock drift problem but gave place to the following questions: What is the maximum value for D_f without significantly distorting the reconstructed voice signal and what are the optimum values for NPktMin and NPktMax? Informal tests were conducted with a special application, which played back a previously recorded message at different user selectable frequencies, to determine the maximum and minimum frequencies at which the differences between the message reproduced at 8 kHz and at the new frequency are negligible. Test results indicate that a fluctuation of 4% around 8 kHz is perfectly acceptable. The choice of NPktMin and NPktMax are directly related with whatever value is attributed to D and to the receiver's intermediate buffer size. Anyway, the minimum value for NPktMin is one and should be, at least, one less than the number of packet arrived before speech reproduction is started. NPktMax should be at least NPktMin+2, so that frequency adjustments aren't always taking place, and smaller than the receiver's intermediate buffer size.

IV. EXPERIMENTAL RESULTS

A large number of tests were conducted in order to determine transmission and reception algorithm's robustness as well as Ethernet's performance as a means for voice and data transmission. Each measurement had an approximate duration of 3 minutes, since this seems a reasonable value for an average telephone call duration. The various tests were executed for packet sizes of 64, 128, and 256 voice samples each, which correspond to 8, 16, and 32 ms of voice, respectively. In all cases, the system was found to provide voice quality service comparable to that of a regular telephone network. Most of the measurements took place in an isolated network where all machines attached were, somehow, related to the tests which were taking place. In this closed environment it is easier to isolate the specific parameter of interest for a particular study. The most important performance metrics used during this experimental study are: percentage of packets lost at the transmitter; average transmission delay - defined to be the time from the moment a packet enters the intermediate transmission buffer to the arrival of transmission acknowledge (successful or not); average access delay - defined to be the time from the moment a packet is submitted for transmission to the arrival of

transmission acknowledge (successful or not); average normalized network load - (n° of bits received/s)/(10 Mbit/s).

We first present system performance in the presence of voice traffic only. Next we analyse voice/data integration in an Ethernet network. This is followed by a detailed discussion of the effects that certain parameters introduce in global system performance.

A. Voice traffic only

In order to estimate the maximum allowable number of active voice stations, in the absence of data traffic and without incurring intolerable packet loss and excessive delay, the following assumptions were made: each traffic generator contributes with a synchronous load (simulating speech traffic) that is an integer multiple of that produced by a real voice channel; we consider that each channel's behaviour is identical to that of a real voice channel, i. e., the percentage of lost packets and the average packet delay is the same for all synchronous channels; the estimate of the number of voice channels is given by the ratio between the average network load and the load of a single voice channel. Fig. 3 shows channel degradation, as the percentage of voice packets discarded at the transmitter, when 4 hosts were contending for the transmission channel. Considering 2% as the maximum limit for packet loss [11], the value of 98 voice channels for 32 ms packets differs from the analytical estimation, presented by Friedman [8], in less than 10%.

Packet loss as a function of network load is illustrated in Fig. 4. Note that packet loss is negligible for loads up 60% and that network performance increases with packet size.

Fig. 5 shows the average transmission delay that voice packets experience. Note that delay is essentially constant for loads up to 60% and that thereafter rises exponentially. For high loads, retransmission frequency, during the vulnerable propagation period, is such that a good portion of transmission trials end up in a collision. This increases delay due to collision resolution which is comparatively larger for smaller packets. If to these delays we add packet acquisition time and the artificial delay introduced at the receiver, thus obtaining the end-to-end delay, we find that it is well within the acceptable limits (100-200 ms) [12].

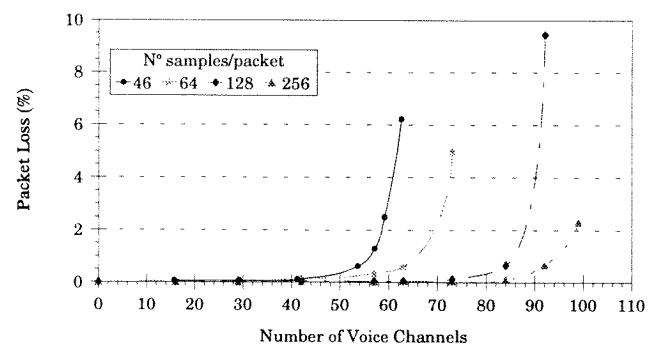


Fig. 3 - Voice channel degradation with increasing number of simultaneous calls for voice traffic only.

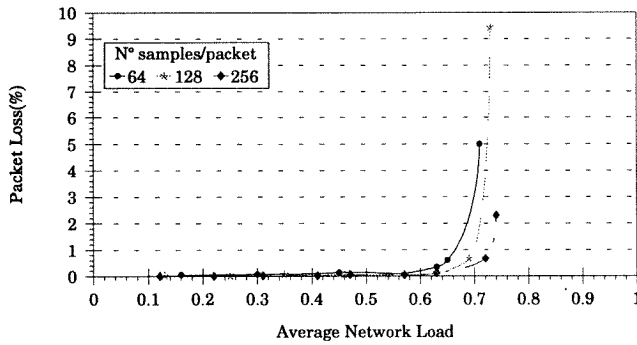


Fig. 4 - Voice channel degradation with increasing load for voice traffic only.

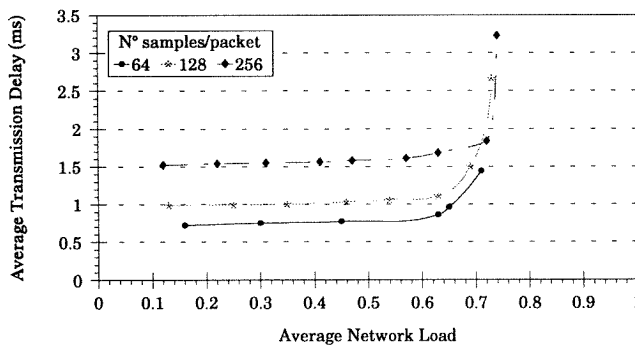


Fig. 5 - Transmission delay with increasing load (voice traffic only).

B. Integrated voice/data traffic

Several tests were performed in order to analyse the behaviour of a voice channel in the presence of real data traffic. Table I collects some performance results of tests made at our University's network. It is a good example of a real data environment with over a hundred different types of equipment running all sorts of applications. Packet loss as well as average transmission delay are small. The establishment of a conversation in such an environment, where data applications are predominant, rises no problems.

The natural question to be asked is how does the voice system respond to a larger data load? A private network with 5 PCs, where data stations generated traffic with an uniformly distributed packet length between 64 and 1500 bytes and packet frequency generation was uniformly distributed to produce a predefined average load, was used in the tests that follow. Packet loss and average transmission delay, for one voice channel, are illustrated in Figs. 6 and 7, respectively. Channel degradation, in the form of lost packets, is negligible for loads up to 55% and is highly dependent on packet size for higher loads. A comparative analysis of Figs. 4, 6 and 5, 7 shows that, in both cases, system performance is identical concerning packet loss and average transmission delay.

Table I
Voice channel performance in a real data traffic environment.

Samples per Packet	Ave. load (%)	Packet loss (%)	Ave. tx delay (ms)
64	3.48	0.00	0.74
	4.38	0.00	0.75
	4.99	0.01	0.75
	7.05	0.00	0.76
	16.57	0.04	0.83
128	3.33	0.01	0.99
	4.20	0.00	0.99
	5.53	0.00	1.01
	8.69	0.02	1.02
	18.27	0.15	1.14
256	4.66	0.00	1.53
	5.23	0.00	1.52
	5.79	0.00	1.53
	7.95	0.02	1.55
	15.08	0.00	1.66

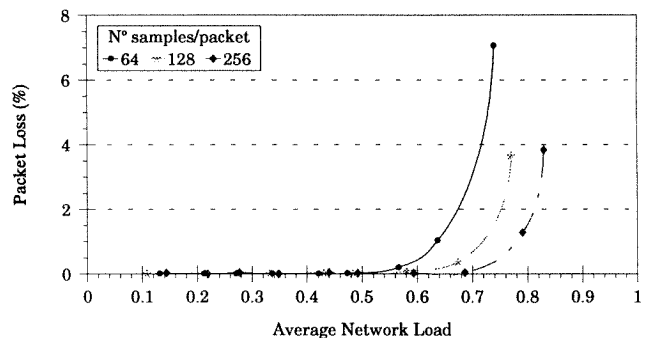


Fig. 6 - Voice channel degradation with increasing load in an integrated voice/data environment.

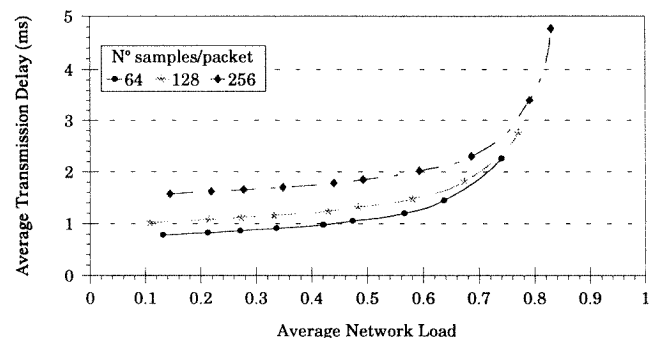


Fig. 7 - Transmission delay with increasing load in an integrated voice/data environment.

The establishment of voice conversations for loads greater than 65-70% is not recommended because at these loads delay rises exponentially and real-time constraints may no longer be kept. The extent to which voice and data may be integrated in an Ethernet is similar to formulating

the following question: How many voice users may be attached to the net, in the presence of data traffic, while preserving minimum voice quality. In order to estimate this value, part of Ethernet's bandwidth was put aside for data and the rest for voice. The estimates presented in Table II were obtained for 2% of lost packets and data loads of 0, 10, and 35%. Note that the end-to-end delays (not shown) are well within the acceptable limits.

C. Effects of other parameters

1) *Influence of active number of hosts:* The above results were all obtained with a relatively small number of machines disputing transmission channel access. From channel efficiency's point of view, the small number of hosts responsible for generating the total load is not so important, once that, for more than 5 machines, overall system performance is not as sensitive to the number of active hosts as it is to the total offered load [20]. However, when we refer to the delay, the same cannot be said. The delay a packet suffers before it is successfully transmitted is intimately related to number of collisions it undergoes, and these depend on the number of machines simultaneously accessing the transmission channel as illustrated in Figs. 8 and 9 (curves identified by markers on the left of the legend were obtained when 4 machines were disputing network access, the others were obtained with 3 machines). The number of host's influence is more evident

Table II
Maximum number of voice users for different data loads.

N° Samples per Packet	0 % Data	10 % Data	35 % Data
	Approx. N° simult. channels	Approx. N° simult. channels	Approx. N° simult. channels
64	70	67	38
128	88	86	48
256	98	96	61

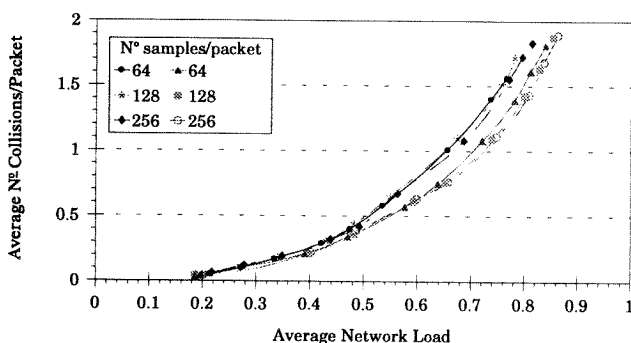


Fig. 8 - Average number of collisions with increasing load when total load is generated by 3 and 4 machines.

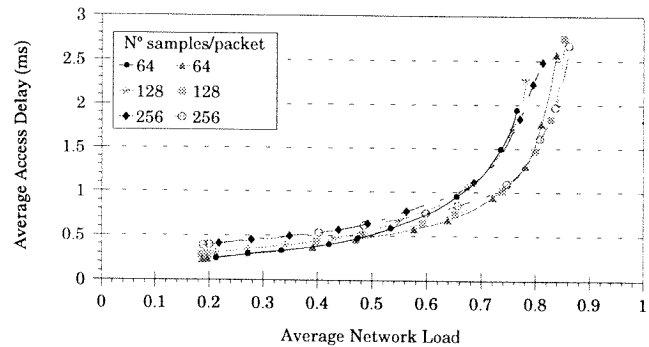


Fig. 9 - Average network access delay with increasing load when total load is generated by 3 and 4 machines.

for loads greater than 50%.

Studies of voice/data traffic on CSMA/CD networks are abundant in the literature. However, the different assumptions and performance metrics used render comparisons difficult. In order to validate the above results, concerning packet loss and delay, some tests were made using, whenever possible, the same performance metrics and implementation characteristics as those presented in the literature. The results in Figs. 3 and 10, obtained in the absence of data traffic and for 5.75 ms voice packets, are comparable to those presented by [1]. Overall system performance, of our implementation, involving voice/data integration produced apparently better results (not presented here) than those presented by [1].

Tests involving a larger number of hosts were not conducted due to the limited number of machines available. However, it is our believe that if the total offered load is distributed by a larger number of hosts, the knee of the various delay and packet loss curves will be slightly shifted to the left.

2) *Buffer Size:* It is possible to increase Ethernet's effective bandwidth by increasing packet size or arranging for larger buffers. Packet size acts directly on the fixed component of the overall delay and indirectly on its variable component. To act directly on the delay's variable component, responsible for most speech loss at high loads, several tests were run with different transmission buffer sizes. The results obtained were as expected and they show a slight increase in system performance at high loads

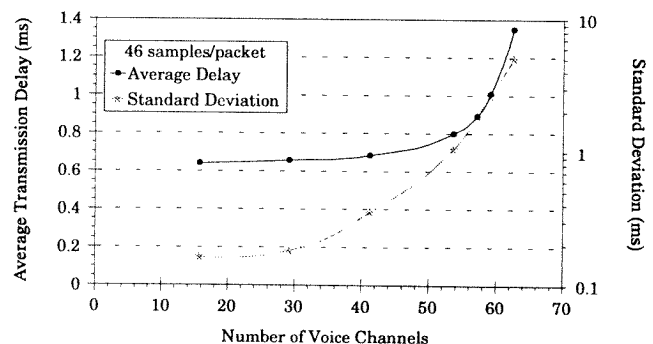


Fig. 10 - Average transmission delay and standard deviation with increasing number of voice channels (voice traffic only).

(for loads of up to 60% the extra buffer size is of little use - packet loss and delay are essentially the same as illustrated in Figs. 11 and 12). However, this improvement is less significant than that obtained when packet size was increased. When we doubled packet size, there was an improvement in packet performance of nearly 7% at 2% of lost packets. When buffer size was doubled, the improvement was less than 2% for the same packet loss (note that in both cases the maximum transmission delay is the same). This suggests that, of the total allowable delay through the system, more delay should be put apart for packet size than for transmission buffer. Similar results are reported in other studies [1].

V. VOICE CHANNEL IMPLEMENTATION OVER TCP/IP

Two major reasons lead to the implementation of a voice channel using the TCP/IP protocols. First, the communication system becomes independent of the network's hardware thus, allowing the establishment of a conversation across two machines no matter the diversity of the underlying hardware technologies. Secondly, it allows the user to simultaneously possess, on the same machine, the vast range of Internet applications (electronic mail, file transfer, remote login, ...) as well as the voice communication system.

A full-duplex voice channel, based on the previously presented transmission and reception algorithms,

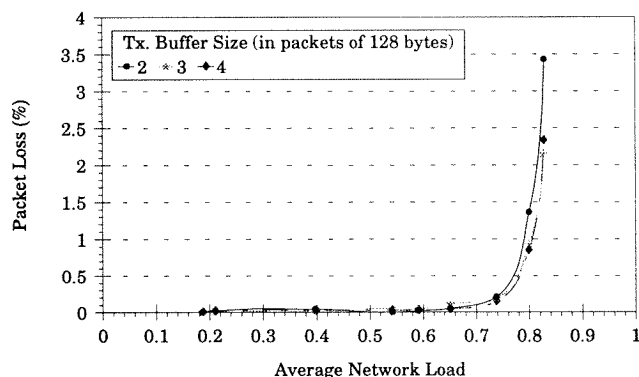


Fig. 11 - Voice channel degradation with increasing load in an integrated voice/data environment for different transmission buffer sizes.

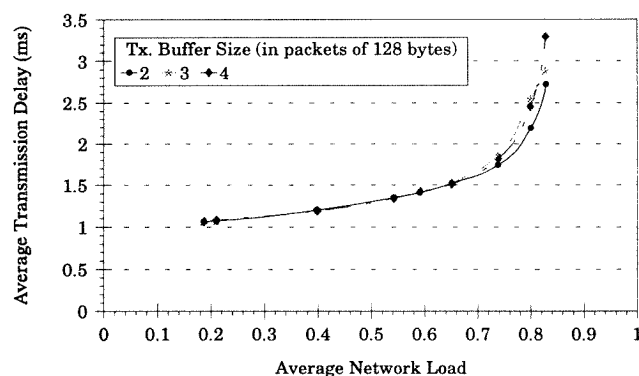


Fig. 12 - Transmission delay with increasing load in an integrated voice/data environment for different transmission buffer sizes.

compatible with the User Datagram Protocol (UDP) was implemented. UDP provides an unreliable connectionless delivery service using the Internet Protocol (IP) to transport messages between machines. It also adds the ability to distinguish among multiple destinations within a given computer.

Similar test, to those presented earlier, were carried out with this voice communication system implementation. The most important result to retain is that such an implementation is possible and practicable. Global system performance does not differ much from that of the voice channel's implementation over the logic link layer, and the conclusions at which we then arrived are here also valid.

VI. SUMMARY AND CONCLUSIONS

An experimental study of a full-duplex packet communication system was presented. The most important conclusion is that voice/data integration is possible in an Ethernet based network, with no access protocol specifications changes needed, as long as it is not overloaded. A voice channel, when submitted to loads of 65 to 70%, suffers negligible degradation in terms of packet loss and overall delay and the quality of the reconstructed signal is acceptable. For such loads, the variable component of the delay is small as well as its variance producing an end-to-end delay well within the acceptable limits of 100-200 ms.

Results show that network performance increases with packet size and that, in a real traffic environment, a voice communication system is possible with less than 2% of lost packets. Packets of 32 ms proved to be the optimum packet size. This value represents a good compromise between the percentage of packet loss, which is inversely proportional to packet length, and the probability of losing a complete phoneme, which rises rapidly for packets larger than 20 ms. The estimate of the maximum number of voice channels, in the absence of data traffic, points to a little less than 100 channels for 32 ms packets and 2% of lost packets.

The reconstruction algorithms, here implemented, proved to be an attractive alternative to the usual method for networks where the mean delay and variance are small.

Introducing feedback into the reproduction frequency, as a means to eliminate information loss due to clock drift, proved to be advantageous. No voice quality degradation results from the variable reproduction frequency.

Finally we find that the Sound Blaster, used as a voice board, is not at all adequate for such a communication system. A custom-designed voice board is necessary to achieve best results.

ACKNOWLEDGEMENT

We wish to thank the Portuguese R&D agency JNICT who supported the MSc work, under grant BM/1770/91-IA, on which this article is based upon.

REFERENCES

- [1] J. D. DeTreville, "A Simulation Based Comparison of Voice Transmission on CSMA/CD Networks and on Token Buses", AT&T Bell Laboratories Technical Journal, vol. 63 No.1, Jan. 1984, pp. 33-55.
- [2] D. H. Johnson, G. C. O'Leary, "A Local Access Network for Packetized digital voice communication", IEEE Transactions on Communications, vol. COM-29, May 1981, pp. 679-688.
- [3] John F. Shoch, "Carrying Voice Traffic Through an Ethernet Local Network, a General Overview", Proc. IFOP WG 6. F Int. Workshop Local Area Comput. Networks, Zurich, Aug. 1980.
- [4] Gary J. Nutt, Douglas L. Bayer, "Performance of CSMA/CD Networks Under Combined Voice and Data Loads", IEEE Transactions on Communications, vol. COM-30, Jan. 1982, pp. 6-11.
- [5] Timothy A. Gonsalves, Fouad A. Tobagi, "Comparative Performance of Voice/Data Area Networks", IEEE Journal on Selected Areas in Communications, vol. 7, No. 5, Jun. 1989, pp. 657-669.
- [6] J. M. Musser, T. T. Liu, G. J. Boggs, "A Local Area Network as a Telephone subscriber loop", IEEE Journal on Selected Areas in Communications, vol. SAC-1, Dec. 1983, pp. 1046-1054.
- [7] Eluzor Friedman, Chaim Ziegler, "Real-Time Voice Communications Over a Token-Passing Ring Local Area Network", in Proc. SIGCOM '86 Symp., Aug. 1986, pp. 52-57.
- [8] Eluzor Friedman, Chaim Ziegler, "Packet Voice Communications Over PC-Based Local Area Networks", IEEE Journal on Selected Areas in Communications, vol. 7, No. 2, Feb. 1989, pp. 211-218.
- [9] J. DeTreville, W. D. Sincoskie, "A Distributed experimental Communication System", IEEE Journal on Selected Areas in Communications, vol. SAC-1, Dec. 1983, pp. 1070-1075.
- [10] Stephen Michaels, "Ten Times Faster", Communications International, vol. 20, No. 4, Apr. 1993, pp. 9-12.
- [11] Nugeghally S. Jayant, Susan W. Christensen, "Effects of Packet Losses in Waveform Coded Speech and Improvements Due to an Odd-Even Sample-Interpolation Procedure", IEEE Transactions on Communications, vol. COM-29, No. 2, Feb. 1981, pp. 101-109.
- [12] Daniel Minoli, "Optimal Packet Length for Packet voice Communication", IEEE Transactions on Communications, vol. COM-27, No. 3, Mar. 1979, pp. 607-611.
- [13] ANSI/IEEE Std 802.3-1985 - Carrier Sense Multiple Access with Collision Detection Access Method and Physical Layer Specifications, IEEE, New York, May 1986.
- [14] Robert M. Metcalfe, David R. Boggs, "Ethernet: Distributed Packet Switching for Local Computer Networks", Communications of the ACM, vol. 19, No. 7, Jul. 1976, pp. 395-404.
- [15] WESTERN DIGITAL, Ethernet PC Adapter Boards WD8003E, Hardware installation guide.
- [16] Joaquim S. Pinto, José L. Oliveira, Joaquim A. Martins, "PC Workstation for Ethernet Monitoring and Management", International Telecommunications Symposium, Set. 1990, pp. 568-570.
- [17] Joaquim A. Martins, Joaquim S. Pinto, José L. Oliveira, "Measuring, Characterizing and Generating LAN Traffic", Conference Proceedings, Singapore ICCS '90, vol. 2, Nov. 1990, pp. 1116-1119.
- [18] Warren A. Montgomery, "Techniques for Packet Voice Synchronization", IEEE Journal on Selected Areas in Communications, vol. SAC-1, No. 6, Dec. 1983, pp. 1022-1028.
- [19] Thomas M. Chen, Jean Walrand, David G. Messerschmitt, "Dynamic Priority Protocols for Packet Voice", IEEE Journal on Selected Areas in Communications, vol. 7, No. 5, Jun. 1989, pp. 632-643.
- [20] John F. Shoch, Jon A. Hupp, "Measured Performance of an Ethernet Local Network", Communications of the ACM, vol. 23, No. 12, Dec. 1980, pp. 711-721.