Adding Voice to an Office Computer Network

by D. C. SWINEHART, L. C. STEWART, and S. M. ORNSTEIN

Computer Science Laboratory
Xerox Palo Alto Research Center

Captions

Figure 1. The physical architecture of the Etherphone system. Etherphones and servers are connected to Ethernet local networks, interconnected by a packet gateway. A 1.5 Mbps Ethernet is used because of the availability of a VLSI controller.

Figure 2. Components of a typical Etherphone office installation including telephone set, microphone and speaker, Etherphone processor, and Ethernet transceiver.

Figure 3. Internal structure of the Etherphone hardware. The slave processor handles voice signal processing while the main processor handles network communications. The two processors normally execute in parallel and communicate through shared memory.

Abstract: This paper describes the architecture and initial implementation of an experimental telephone system developed by the Computer Science Laboratory at the Xerox Palo Alto Research Center (PARC CSL). A specially designed processor (*Etherphone* $_{TM}$) connects to a telephone instrument and transmits digitized voice, signalling, and supervisory information in discrete packets over an Ethernet local area network. When used by itself, an Etherphone processor provides the functions of a conventional telephone, but it comes into its own when combined with the capabilities of a nearby office workstation, a voice file service, and other shared services such as databases. Most of the work so far has gone into the basic provisions for voice switching and transmission. Today the system supports ordinary telephone calls and simple voice message services. We will expand these functions as we explore the integration of voice with our experimental office systems.

Introduction

Voice, in meetings and in conversation, has always played a major role in the conduct of business in the office. Since the introduction of the telephone, the role of voice in the office has reached major proportions. There have been many improvements in telephony since commercial service began in the 1880's, but except for direct dialing, few have resulted in functional improvements for the subscriber. The advent of computers and new communications technologies has begun to allow the developers of telephone equipment to provide substantial increases in function and convenience. In addition to improvements in voice communications, these advances include data and video transmission, with recent forays into electronic mail and other non-voice functions.

At PARC, we and our colleagues have approached the problem of office productivity from a different perspective. Our research into personal information systems has provided considerable experience with the design, production, and use of personal computers and programs that augment or supplant traditional paper-based office activities. At the heart of this work is the personal computer, such as the Xerox 8010 Professional Workstation. The personal workstation gives its user a coherent and integrated user interface supporting the creation, modification, storage, and printing of documents, memos, messages, and personal databases. High speed Ethernet communications link each workstation to those of other users and to other computers that provide a variety of services, such as file storage, electronic mail distribution, and high quality printing.

Ironically, the typical office in our laboratory includes a multifunction workstation and a stand-alone telephone instrument two isolated systems. Telephone equipment manufacturers have approached the merger of these technologies by treating office applications as extensions to their basic telephone service (Refs. 27-29). With the Etherphone project, we have the opportunity to gain

additional insights into the potential for voice by instead treating voice communications as an extension to our existing office systems.

This paper discusses our goals and our reasons for undertaking the project. An overview of the system architecture is presented, and the fundamentals of Ethernet voice transmission are described. These introductory sections provide the context for a description of the structure and functions of the major system components, including applications packages and programs that have been developed for existing workstations.

Objectives

The drawbacks of the telephone are as evident as its benefits. So many calls fail to reach the intended party that the term "telephone tag" has been coined to describe the resulting frustration. Conversely, when a call is successful, the called party often wishes it had failed; unwanted or ill-timed telephone calls are a major source of interruption and annoyance in the workplace.

As long as people remain busy and mobile, many of these problems will defy solution by technical means. Moreover, great care is required whenever one considers changes to a social institution. The conventions of telephone use are deeply established. Although people are often annoyed by the present arrangements, they also tend to be quite conservative and to resent changes in the system. Nevertheless, we feel that improvements in several areas will prove useful:

Improved Control

The twelve pushbuttons and hookswitch on a telephone, combined with a set of call progress tones from the central office, provide a reasonable user interface for placing and answering telephone calls. However, newer telephone systems have made available myriad additional features without substantially improving the user interface. Multiple-line telephones and telephones with additional pushbuttons for common functions improve matters, but most attempts to provide more telephone capabilities become too complicated to control long before their designers run out of ideas. Many features are never used because they are too hard to remember and too hard to use.

Similar problems with the composition and editing of text and graphical documents led to the development of our workstation-based systems (Ref. 23). The contextual power of the high-resolution display screen, "mouse" pointing device, and menu-based software have increased the ability of the casual user to master large numbers of operations and quite complex situations. We believe that the management of interactive voice will submit to the same techniques, provided the workstations have adequate programmed access to the telephone-switching functions.

Informed "Agents"

A private secretary can act as an agent for an executive who cannot afford the time to answer every incoming call. The secretary can take messages, locate elusive conversants, and screen incoming calls. In the modern office, this kind of assistance is rare. Attempts to simulate some of these functions with simple recording machines have been notoriously unsuccessful, since many callers find them annoying or intimidating. While an unattended system cannot provide the finely tuned discrimination of a private secretary, it can provide at least simple filters of the flavor: ‘‘No calls for half an hour, unless it’s Bob, or unless Fred calls about the budget.’’

Conversely, for the individual who needs to be out of the office but who wishes to be available by telephone, we can employ the capabilities of our distributed systems to allow the telephone system to track his location. The same kinds of information in the hands of human attendants can supply informed assistance to those calling from outside the Etherphone system.

Voice as Data

Leaving a recorded message can often be an effective substitute for those calls that go unanswered. Telephone answering machines make this possible, but their imperative nature demands a well-constructed, composed message rather than an informal communication. Few of us are able to produce such a composition in real time. These machines alienate callers, who often hang up without leaving any useful message.

Our approach to this problem is to shift the control over producing a voice message from the callee to the caller. The system can allow the caller to record, audition, and modify a message before sending it or deciding not to. The message might contain text, recorded voice, or a combination. If experience with electronic text messages is any guide, users will come to prefer recorded messages to telephone calls in some situations. Experience with stand-alone voice message systems that are beginning to appear tends to support the value of caller-controlled voice messages (Refs. 15, 26, 28).

Voice in messages and documents will find application far beyond simple telephony. We intend to explore voice annotation of ordinary office documents, spoken reminders, musical prompts, and other combinations of voice and visual media.

In summary, we feel that our computing and communications environment can be brought to bear on voice in two ways. One way is to improve our control over otherwise ordinary voice communications; the other is to incorporate the voice medium into our integrated environment on an equal footing with text, graphics, and data. While the facilities we have built or proposed are not entirely new, we believe that the full integration of voice into our computer-based office systems will make existing functions more effective and new applications possible.

System Architecture

PARC Environment

For most office applications, a distributed network of personal workstations, interconnected by high-bandwidth local-area networks or by specialized PABX lines, is supplanting the traditional central computing installation. At PARC, both the standard Ethernet, operating at 10.0 Mbps, and an earlier research Ethernet, operating at 3.0 Mbps, are in general use (Refs. 16, 6). These networks link personal workstations to each other and to shared resources such as file servers, mail servers, and high-quality laser printers (Refs. 2, 17). Communications outside the local area are provided primarily by leased lines most operating at 9.6 or 56 Kbps, with some operating at up to 1.5 Mbps (Refs. 3, 20). Several types of personal workstations are in use, including the Xerox Alto, Dorado (1132), 1108 and 1100 processors (Refs. 24, 14). These same processors are used as the controllers for most servers. Every workstation has a high-resolution bitmap display (1000 by 800 pixels), a keyboard, and a mouse (Ref. 23).

In our laboratory, most users work within the Cedar programming environment. Cedar is an experimental integrated system that is used for software development and computer-aided design, as well as for day-to-day operation of office applications, such as document preparation, graphics, and electronic mail (Refs. 5, 23).

Our approach to applications research is to build prototype applications based on the Cedar environment, then to use them in our daily work. This approach encourages us to carry the development of our implementations far enough that performance and reliability problems do not impair our ability to evaluate their usefulness. Once they are introduced, these applications supply both ideas and useful components for further developments. Most of our applications are structured so that users can either operate them directly, or write programs that combine their functions with those of other facilities.

Basic Architectural Decisions

The goals of improved telephone functionality and integrated voice add two new requirements to the existing capabilities in Cedar: rapid and versatile control over voice transmission and switching, and a high performance voice storage capability.

We surveyed several possible architectures for the system. Subscriber line access to a voice storage system and use of our existing Centrex service through DTMF signalling was rejected due to the low bandwidth and the inflexible control over switching. Use of a commercial PABX was attractive, but the necessary switching capabilities were not commercially available. The use of our own computers to control the switching operations of a standard PABX was rejected primarily due to the nontechnical difficulties of arranging the necessary cooperative effort.

Although reluctant to undertake the necessary development work, we concluded that the most effective way to satisfy our needs was to construct our own transmission and switching system. We elected to use the Ethernet to transmit voice as well as control information. We chose Ethernet because of its pervasiveness in our environment and in order to demonstrate the effectiveness of Ethernet for voice.

The potential of the Ethernet for packet voice transmission has been described elsewhere (Ref. 22). In the following section, we present only a brief analysis.

Ethernet Transmission of Voice

Traditional telephony relies upon circuit-switched communications. Using packet-switched communications to provide the same functions raises some interesting issues:

Delay

The first issue one must address is the end to end (microphone to receiver) delay for interactive telephone calls, since even relatively short delays can be disturbing. Specifically, end-to-end delays of less than 20 milliseconds are generally undetectable, delays of 40 to 80 milliseconds can cause trouble when significant echo is present, and delays greater than 100 milliseconds begin to disrupt normal conversation (Refs. 1, 10, 4).

In a packet voice system, the delay has two components: packetization delay and transmission delay. Packetization delay arises because the first sample of a packet cannot be transmitted until the last sample has been digitized. The delay introduced is equal to the product of the sampling interval (125 microseconds is typical) and the packet size. Transmission delays can result from software delays and network-access delays.

Ethernet transmission exhibits complex relationships among packet size, offered load, and access delay. Studies based on typical packet sizes (Refs. 16, 9, 13, 21, 25) have shown that when offered loads fall below a specific threshold there are few collisions, and network-access delays are negligible in relation to the other delay components. Above the threshold, delays climb rapidly as the network becomes saturated. Sending only large packets suitable for data transmission pushes this threshold well above 90% of full utilization. Shortening the packets to a size suitable for interactive voice reduces the threshold to the 50 - 80% range. One must engineer an Ethernet voice system to operate below the knee of this delay curve.

Our voice hardware and protocols limit packetization and transmission delay to under 40 milliseconds. We have found that voice delay due to Ethernet transmission is not a problem within the local area. However, it may become significant when our delays are added to others, notably satellite transmission times.

Capacity

One major advantage of Ethernet voice transmission is its efficient sharing of a high-bandwidth channel among users. Combined with a method that we call *silence detection*, this sharing permits concentration effects similar to those of Time Assigned Speech Interpolation (TASI) (Ref. 1). The silence-detection method involves transmitting only those packets containing significant signal energy.

To estimate the resulting capacity, consider an implementation whose voice packets contain 160 eight-bit samples (representing 20 milliseconds of voice), collected at 125 microsecond intervals. Using the 10 Mbps standard Ethernet, each packet includes 64 bytes of overhead, so each (one-way) voice connection consumes 90 Kbps. Assuming the worst case, where traffic must be limited to 50% utilization to avoid unacceptable access delay, the bandwidth available in a network dedicated to voice transmission is 5 Mbps, yielding 28 full-duplex "trunks" (55 one-way voice streams.) If we estimate a TASI advantage of 1.6 for this size trunk group (Ref. 1), a dedicated 10 Mbps Ethernet can support about 45 conversations. If at most 20% of telephones are in use at once, such a network could support in excess of 225 subscribers. This argument requires that control traffic and other non-voice traffic is negligible compared to voice traffic.

From these arguments, it seems clear that a simple single-cable Ethernet voice-transmission system can support a sizeable installation. A multiple-cable installation would be needed to carry voice traffic above these limits, or to support significant data traffic in addition to voice. Individual cables would be interconnected by packet gateways or conventional circuit switches. The very different statistics of data and voice traffic might make it advisable to partition a large system into parallel data and voice cables. Installation of a multiple-cable system would require attention to usage patterns and other traffic-engineering considerations.

Security

Security can be a problem with Ethernet or any other broadcast medium, since no physical protection against eavesdropping is possible. We impose security on our transmissions by encrypting all voice and control traffic using the Data Encryption Standard (Ref. 7). The availability of single-chip DES devices operating at 900 Kbps makes this possible. We distribute the keys using a method similar to the trusted authentication server approach advocated by Needham and Schroeder (Ref. 18).

New Functions

The broadcast nature of the Ethernet admits some new communication possibilities. Conference calls among several sites can be achieved without the need for special conference-bridge hardware: multicast techniques permit voice packets from a given site to be received and combined by each of the others (Ref. 8). In

the extreme, voice from a meeting or conference can be transmitted once and received by any number of listeners, without consuming any more bandwidth than a single conversation consumes.

Etherphone Voice Protocols

The voice protocols for the Etherphone system are based on the preceding analysis. We transmit 8000 8-bit samples per second, using the industry-standard mu-255 encoding (Ref. 11). The Etherphone processor actually supports two related protocols, one for interactive voice (telephone calls), the other to play and record stored voice.

The interactive-voice protocol was designed to meet a delay budget of 40 milliseconds end-to-end. This delay consists of a packetization delay of 20 milliseconds, hardware latency of 5 milliseconds for encryption and Ethernet transmission, software delays of up to 5 milliseconds, and what we call an *anti-jitter* delay of 10 milliseconds. To meet this budget, the protocol specifies the transmission of fifty packets per second, each containing 160 voice samples and 36 bytes of addressing and control overhead.

Anti-jitter delay is introduced by buffering at the receiving station to allow for variations in the arrival times of packets. One might say the protocol operates with about one-half packet of buffering. This delay budget does not allow time for retransmissions in the event of lost packets. This has proven acceptable, because the native packet loss rate of well-designed Ethernet components has been shown to be less than one packet in two million (Ref. 12).

Recording and playback of stored voice has slightly different characteristics. Transmission delay is not particularly important, provided that the start-up delay from request to playback is reasonably short. The stored-voice protocol calls for 100 milliseconds of buffering. The additional buffering is desirable because the shared voice file server may not be able to schedule the transmission times of packets as accurately as a dedicated voice terminal can. The additional buffering also permits retransmission, although we have not found the need to implement it.

Major System Components

Hardware choices

Perhaps the most significant aspect of the Etherphone system architectural design was the choice of voice terminal equipment. Given the existing PARC environment, voice terminals were the only new equipment required. The alternatives we explored were to add special hardware and software to each workstation, or to design a standardized Ethernet telephone peripheral the *Etherphone* TM that would do all of the necessary voice conversion and transmission. We chose the Etherphone approach for several reasons. We wanted to make voice capabilities available to

users over our full range of workstations. Furthermore, we did not want to redesign the voice hardware for each type of workstation, nor were we confident that all of them could handle the transmission and control of telephone conversations with sufficient reliability and without substantial loss of performance in their other duties.

An additional hardware-related decision was motivated by economics. We were able to limit the size and speed of the Etherphone processor, and thus its cost, by restricting its activities to those that could not be readily performed by writing software for the existing workstations or servers.

Etherphone users converse with each other using only Ethernet communications. To make calls outside the system, however, access to the public telephone network is needed. The preferred method, transplanted from the conventional PABX environment, would involve connecting a group of central-office trunk lines to a server that would complete the connections over the Ethernet. For our prototype system, we chose instead to provide each Etherphone with a jack for connection to an existing subscriber line. To complete telephone calls involving non-Etherphone users, the Etherphone connects the telephone instrument to the subscriber line. This choice allows the relatively small community of Etherphone users to retain their short extension numbers, avoids the cost of developing a trunk server, and permits direct use of the subscriber line as a backup if the hardware or software fails.

System Organization

In addition to the Etherphones, we have designed and implemented a *telephone control server* and a *voice file server*, both of which are Cedar-based applications programs. Figure 1 shows the structure of the resulting Etherphone system.

The Etherphone is responsible for voice digitization, for the analog interface to the switched telephone network, and for implementing the voice-transmission protocols. It is designed to be a voice peripheral, without local intelligence or system control responsibility. Instead, it reports user actions to the telephone control server, which returns detailed commands that control the Etherphone functions.

The *telephone control server* is the system controller. Its responsibilities are analogous to the control responsibilities of a conventional PABX: monitoring the state of the system, keeping track of the state of each Etherphone, and setting up all connections. It must also coordinate the operation of the software in users' *workstations*. This workstation software provides improved user interfaces for the telephone and other more experimental applications.

The *voice file server* uses a general-purpose computer with attached disk storage for voice. It performs standard file server functions, but is specialized for the real-time needs of telephony.

Finally, existing standard file servers and database services are used for storage of telephone directory information and for storage of users' call filters and other information.

The following sections amplify the descriptions of these major system components.

Etherphone

The Etherphone is a dual-processor computer with interfaces for the Ethernet, DES encryption, RS-232 serial devices, and voice. The *main* processor is responsible for network communications and control while the *slave* processor handles voice signal-processing functions. An analog board, a digital board, and a power supply occupy a convection-cooled cabinet measuring 12" x 13" x 5". This package is designed to sit under a user's desk, while the telephone set, microphone, and speaker occupy positions of convenience. This arrangement is shown in Figure 2. Figure 3 is a block diagram of the Etherphone's internal architecture.

The analog board is centered around an 8 by 8 analog crossbar switch that interconnects various voice sources and sinks: telephone set, telephone line, analog-to-digital converters, DTMF decoder, microphone and speaker, and line-level auxiliary inputs and outputs. The analog-to-digital conversion functions are accomplished by coder-decoder (CODEC) devices. The telephone set and telephone line interfaces are connected by relays so that a power failure or system crash will restore standard telephone service.

The digital board contains the two processors, memory systems, timing logic, and digital interfaces for the Ethernet, RS-232 devices, DES encryption, analog-to-digital conversion, and control of the analog hardware. A watchdog timer is provided for automatic recovery from an intermittent fault.

The main processor is an Intel 8088 with 8K bytes of EPROM and 56K of RAM. DES encryption is accomplished by a 900 Kbps single-chip device with direct memory access (DMA). The Ethernet controller is an internal Xerox LSI part using the protocols of the 3 Mbps experimental Ethernet, but operating at 1.5 Mbps. The RS-232 interface permits the the connection of a local display and keyboard as an option.

Programs written in the C language supply all of the main processor's functions except for a small number of low-level device drivers, which are written in an assembly language. The software includes a round-robin-scheduled multitask operating system, and a network package that implements both the voice protocols and protocols for connecting a local control program to the telephone control server. This local control program communicates user actions (such as lifting the switchhook or keying in a "5" on the telephone) to the server without interpretation. Commands from the control server instruct the Etherphone's local control program to establish Ethernet voice connections, configure the crossbar switch, generate programmed tone sequences, and manage the

telephone line interface.

The slave processor is also an 8088. It has access to 8K of private EPROM, 2K of RAM, and shared access to 48K of main memory. The slave processor executes a small, carefully coded, assembly-language program. Its functions include silence detection, the combination of incoming packets to achieve conference-call bridging, echo suppression, gain control, CODEC control, and low-level buffer management in shared memory.

The overall performance of the Etherphone is sufficient to support two simultaneous Ethernet conversations. For example, two conversations are needed in order to forward an arriving outside call to an attendant's Etherphone when the telephone set is already being used for an Ethernet call. In a different configuration, the Etherphone will support a four-party conference call.

Because the voice sampling rate of each Etherphone is set by its own crystal-controlled clock, synchronization of two Etherphones participating in a conversation presents an intriguing problem. A pair of Etherphones may have clocks differing by as much as one part in 10,000. In the steady state, this frequency error causes the quantity of buffered voice at the receiving Etherphone to slowly increase or decrease. Since silence detection is used to reduce transmission bandwidth, the correct buffer depth is reestablished during each silent interval. Communications with the voice file server avoid this problem by use of software feedback the file server is driven by the Etherphone clock.

Telephone Control Server

The telephone control server performs the functions normally associated with the common control facilities in a conventional PABX or telephone switching office. However, because it does not have to switch or transmit any of the actual voice traffic, it has no special hardware needs and requires only modest computing power to support its activities. The telephone control server supplies two classes of functions: Etherphone intelligence and call processing.

The control server contains the Etherphone intelligence software that interprets user actions at an Etherphone. Most Etherphones have only the standard DTMF keypad, but a small display and keyboard can be added to provide a more effective user interface when a workstation is not available (see Figure 1). In either case, control server software interprets and responds to the user's actions. These responses include service requests such as setting up and taking down connections, and sending control instructions back to the Etherphones.

The control server also performs the telephone-control functions usually referred to as call processing. It provides control and bookkeeping for connections in collaboration with the Etherphone user interface programs for the participating parties. The control server provides access to a number of databases, including directories establishing the relationships among Etherphone net-

work addresses, the names of individuals and services (such as the voice file server), telephone extensions, office locations, and nearby workstations. The call-processing portion of the control server has an additional function: it synchronizes the activities of each Etherphone with the workstation applications that manage that Etherphone and that use it to provide their voice connections. When user actions coming from the Etherphone and from the workstations are inconsistent, the server resolves the conflicts.

The telephone control server includes some maintenance facilities not directly connected with system operation. For example, the server contains software to detect the distress packets that Etherphones broadcast when they are first powered on or when unrecoverable failures occur. This caretaker software responds by performing automatic downloading and remote maintenance of Etherphone software. By consulting another database, the control server can support the simultaneous operation of Etherphones with different software or hardware configurations. The same database assigns each Etherphone to a particular telephone control server, so that one server and a few Etherphones can be used to develop new system versions, while another server supports normal operations for the remaining users.

For system control, we were able to take advantage of the recent development of remote procedure call protocols (Ref. 19). Remote Procedure Call (RPC) is a means for transforming the message-passing semantics of packet-switching networks into the procedure-call format familiar to programmers. RPC largely relieves applications programmers from any worries about packet formats, addressing, reliable communication, and security. This makes it possible to defer decisions about the partitioning of a distributed system until very late in the design. Our task was reduced to producing an RPC implementation for the Etherphone control processor, then specifying the procedural interfaces between different system components.

Voice File Server

The voice file server uses the stored-voice protocol to record and play back digitized voice segments. It runs on a Xerox Dorado computer that can be equipped with as many 300 Mbyte disks as usage warrants. At 8000 bytes per second, the voice-storage capacity of each disk is over 8 hours.

The disk is organized so that storage is allocated in one second units. This permits disk activity on behalf of a single record or playback operation to be limited to one contiguous disk transfer per second. Nevertheless, user software can specify the order and duration of voice-segment access at a grain of one millisecond. This facility makes it possible to experiment with voice editing. The very high network-communication loads presented by multiple voice-protocol connections present a special challenge. The current implementation is capable of handling about eight simultaneous transfers.

All voice is stored in encrypted form. The associated keys are stored by the telephone control server in a directory along with information granting appropriate access to each voice segment. Since each voice segment is stored only once, regardless of the number of users granted access, the file server directory also keeps track of the number of outstanding references to each segment. Voice storage is reclaimed automatically when no references remain.

Workstation applications

Most users of the Etherphone system will have both an Etherphone and a personal workstation. Although there can be conflicting requests for service originating from the workstation and the user's Etherphone, voice-related applications software on the workstation generally takes priority in decision making. As an example, consider an arriving call. The Etherphone is always prepared to announce the call with a ringing tone, but workstation software is first given the choice of rejecting the call, allowing the Etherphone to ring normally, or choosing some other way to announce the call.

Our goals include both producing useful workstation applications and making the Etherphone functions available to other programmers who would like to produce voice-related applications. For this reason, the workstation software includes a basic package of support facilities, then another layer of specific application programs that operate as clients of the basic package.

The support package provides applications packages with a procedural interface that is as simple as we could make it without reducing the available functions. The simplicity is obtained primarily by suppressing necessary but uninteresting details: communications with the telephone control server and the synchronization of workstation actions with those of the Etherphone itself. The support package is also responsible for preventing actions by applications programs that could reduce the reliability of the basic telephone system and other applications. The dedicated facilities of the voice transmission and switching system makes this task easier than it might otherwise be.

The existing Cedar applications layer provides a rudimentary user interface for placing and receiving telephone calls, for managing a personal telephone directory, and for dealing with voice messages. The telephone-management facilities allow call placement by name rather than by number, or by pointing at recipients' names in a directory displayed on the screen. Other displays announce the names of callers, and present logs of telephone activity. The voice-message system is patterned after our text-mail system. Facilities are provided for recording and reviewing messages, for giving a voice message a text subject field, and for directing delivery of the message to one or to several recipients. A table of contents of the user's messages is also provided. Individual messages can be played, deleted, or saved for a later time.

Since the Cedar environment is available on several different types of workstations, workstation access to the Etherphone facilities will be available at least to all Cedar users. Furthermore, the functionality of the support package is not extensive, so we do not expect it to be difficult to build similar packages that will make advanced voice capabilities available to users of other local programming environments.

Status and Plans

We are currently testing an initial system using prototype Etherphone hardware. We are planning to install additional Etherphones, offering service to a sizable segment of our laboratory, in order to test the value of the enhanced control and message capabilities and to allow Cedar programmers to develop their own applications. With the enabling facilities in place, we are just beginning to explore the potential applications of integrated voice in the office environment.

Acknowledgments

We would like to thank Jim Horning, Ken Pier, Mary Claire van Leunen, Meg Withgott, and Polle Zellweger for their valuable comments during the preparation of this paper.

References

1. American Telephone and Telegraph Company, *Notes on The Network*, 1980.

2. A. D. Birrell, R. Levin, R. M. Needham, M. D. Schroeder, "Grapevine: An Exercise in Distributed Computing," *Comm. ACM, 25, 4* (April 1982), 260-274.

3. D. R. Boggs, J. F. Shoch, E. A. Taft, R. M. Metcalfe, "Pup: An Internetwork Architecture," *IEEE Trans. Communications, Com-28, 4* (April 1980), 612-624.

4. P. T. Brady, "Effects of Transmission Delay on Conversational Behavior on Echo-Free Telephone Circuits," *Bell System Technical Journal, 50, 1* (January 1971), 115-134.

5. D. K. Brotz, "Laurel Manual," Report CSL-81-6, Xerox Palo Alto Research Center, May 1981.

6. R. C. Crane, E. A. Taft, "Practical Considerations in Ethernet Local Network Design," *Proc. 13th Hawaii International Conference on Systems Sciences*, Honolulu January 1980, 166-174.

7. Data Encryption Standard, Federal Information Processing Standard (FIPS), Publication 46, National Bureau of Standards, U. S. Department of Commerce, January 1977.

8. J. W. Forgie, "Voice Conferencing in Packet Networks," *International Conference on Communications*, Seattle, June 1980.

9. T. A. Gonsalves, "Packet-Voice Communication on an Ethernet Local Computer Network: an Experimental Study," *Proc. SIGCOMM 1983 Symposium on Communications Architectures and Protocols,* Austin TX, March 1983, 178-185.

10. J. R. Cavanaugh, R. W. Hatch, J. L. Sullivan, "Models for the Subjective Effects of Loss, Noise, and Talker Echo on Telephone Connections," *Bell System Technical Journal, 55, 9* (November 1976), 1319-1371.

11. H. H. Henning, J. W. Pan, "D2 Channel Bank: System Aspects," *Bell System Technical Journal, Volume 51,* (October 1972), 1641-1657.

12. J. F. Shoch, J. A. Hupp, "Measured Performance of an Ethernet Local Network," *Comm. ACM, 23, 12* (December 1980), 711-721.

13. D. H. Johnson, G. C. O'Leary, "A Local Access Network for Packetized Digital Voice Communication," *IEEE Trans. Communications, Com-29, 5* (May 1981), 679-688.

14. B. W. Lampson, K. A. Pier, "A Processor for a High-Performance Personal Computer", *Proc. 7th Symposium on Computer Architecture, SigArch/IEEE,* La Baule, May 1980, 146-160.

15. N. F. Maxemchuk, "An Experimental Speech Storage and Editing Facility," *Bell System Technical Journal, 59* (October 1980), 1383-1395.

16. R. M. Metcalfe, D. R. Boggs, "Ethernet, Distributed Packet Switching for Local Computer Networks," *Comm. ACM, 19, 7* (July 1976), 395-404.

17. J. G. Mitchell, J. Dion, "A Comparison of Two Network Based File Servers," *Comm. ACM, 25, 4* (April 1982), 233-245.

18. R. M. Needham, M. D. Schroeder, "Using Encryption for Authentication in Large Networks of Computers," *Comm. ACM, 21, 12* (December 1978), 993-998.

19. B. J. Nelson, *Remote Procedure Call,* Report CSL-81-9, Xerox Palo Alto Research Center, May 1981.

20. Internet Transport Protocols, Xerox System Integration Standard XSIS 028112. December 1981.

21. G. J. Nutt, D. L. Bayer, "Performance of CSMA/CD Networks Under Combined Voice and Data Loads," *IEEE Trans. Communications, Com-30, 1* (January 1982), 6-11.

22. J. F. Shoch, "Carrying Voice Traffic Through an Ethernet Local Network A General Overview," presented at the *IFIT WG 6.4 Int. Workshop Local-Area Computer Networks,* Zurich, Switzerland, August 1980.

23. D. C. Smith, E. Harslem, C. Irby, R. Kimball, "The Star User Interface, an Overview," *Proc. National Computer Conference* (June

1982), 515-528.

24. C. P. Thacker, E. M. McCreight, B. W. Lampson, R. F. Sproull, D. R. Boggs, "ALTO: A Personal Computer," in *Computer Structure: Readings and Examples*, D. Sieworek, C. G. Bell, and A. Newell, Eds. McGraw-Hill, New York, 1981.

25. F. A. Tobagi, N. Gonzalez-Cowley, "On CSMA-CD Networks and Voice Communication," *Proc. Infocom* (1982), 122-1278.

26. W. D. Guns, "Computer Based Voice Mail: Status and Outlook," *SRI International Business Intelligence Program, File No. 83-757*, January 1983.

27. P. A. Strudwick, R. E. Adkins, "Integrating Voice and Data at the Human Interface: the Key to Effective Communications in the Office of the 80's," *Proc. National Telecommunications Conference* (November 1981), New Orleans, page D1.2.

28. H. G. Jud, R. E. Winter, "A Modern Integrated PABX with Centralized Message Recording and Remote Distribution," *Proc. National Telecommunications Conference* (November 1981), New Orleans, page F3.3.

29. J. Edwards, D. Lieberman, "The ROLM CBX as an Integrated Voice/Data Communications Switching Unit," *Proc. National Telecommunications Conference* (November 1981), New Orleans, page F3.1.

The authors are located at the Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA 94304.