Experience with Speech Communication in Packet Networks

CLIFFORD J. WEINSTEIN, MEMBER, IEEE, AND JAMES W. FORGIE, MEMBER, IEEE

Abstract—The integration of digital voice with data in a common packet-switched network system offers a number of potential benefits, including reduced systems cost through sharing of switching and transmission resources, flexible internetworking among systems utilizing different transmission media, and enhanced services for users requiring access to both voice and data communications. Issues which it has been necessary to address in order to realize these benefits include reconstitution of speech from packets arriving at nonuniform intervals, maximization of packet speech multiplexing efficiency, and determination of the implementation requirements for terminals and switching in a large-scale packet voice/data system. A series of packet speech systems experiments to address these issues has been conducted under the sponsorship of the Defense Advanced Research Projects Agency (DARPA).

In the initial experiments on the ARPANET, the basic feasibility of speech communication on a store-and-forward packet network was demonstrated. Techniques were developed for reconstitution of speech from packets, and protocols were developed for call setup and for speech transport. Later speech experiments utilizing the Atlantic packet satellite network (SATNET) led to the development of techniques for efficient voice conferencing in a broadcast environment, and for internetting speech between a store-and-forward net (ARPANET) and a broadcast net (SATNET). Large-scale packet speech multiplexing experiments could not be carried out on ARPANET or SATNET where the network link capacities severely restrict the number of speech users that can be accommodated. However, experiments are currently being carried out using a wide-band satellite-based packet system designed to accommodate a sufficient number of simultaneous users to support realistic experiments in efficient statistical multiplexing. Key developments to date associated with the wide-band experiments have been 1) techniques for internetting via voice/data gateways from a variety of local access networks (packet cable, packet radio, and circuit-switched) to a long-haul broadcast satellite network and 2) compact implementations of packet voice terminals with full protocol and voice capabilities.

Basic concepts and issues associated with packet speech systems are described. Requirements and techniques for speech processing, voice protocols, packetization and reconstitution, conferencing, and multiplexing are discussed in the context of a generic packet speech system configuration. Specific experimental configurations and key packet speech results on the ARPANET, SATNET, and wide-band system are reviewed.

I. Introduction

PACKET techniques provide powerful mechanisms for the sharing of communication resources among users with time-varying demands, and have come into wide use for provision of data communications services to the military and commercial communities. The primary application of packet techniques has been for digital data communications where the bursty nature of user traffic can be exploited to achieve large efficiency advantages in utilization of communication resources. Packet networks [1]–[8] using a variety of point-to-point and broadcast transmission media have been developed for these applications, and techniques have been developed for internetwork communication [10], [11] among dissimilar nets.

Packet techniques offer significant benefits for voice as well as for data [15]–[33]. The integration of digital voice with data in a common packet-switched system offers potential cost savings through sharing of switching and transmission resources [30], as well as enhanced services for users who require access to both voice and data communications [59]–[61]. Packet internetworking techniques can be applied to provide intercommunication among voice users on different types of networks. Significant channel capacity savings for packet voice can be achieved by transmitting packets only when speakers are actually talking (i.e., during talkspurts). The silence intervals can be utilized for other voice traffic or for data traffic. Packet networks offer significant advantages for digital voice conferencing in terms of channel utilization (only one of the conferees needs to use channel capacity at any given time) and in terms of control flexibility. A packet network allows convenient accommodation of voice terminals with different bit rates and data formats. Each voice encoder will use only the channel capacity necessary to transmit its information rather than the fixed minimum bandwidth increment typically used in circuit-switched networks. The digitization of voice in packet systems provides the opportunity for security techniques to be applied as necessary to the speech traffic. Secure packet data communication techniques [13] can be applied as well for data users who require this service. Packet networks also provide a system environment for effective exploitation of variable-bit-rate voice transmission techniques, either to reduce average end-to-end bit rate or to dynamically adapt voice bit rate to network conditions.

It has been necessary to address a number of issues in order to develop the techniques required to realize these benefits. The development of packet protocols for call setup and speech transport, and strategies for reconstitution of speech from packets arriving at nonuniform intervals have been required. Other issues include the development of efficient packet speech multiplexing techniques,

Manuscript received April 11, 1983; revised August 5, 1983. This work was supported by the Defense Advanced Research Projects Agency.

The authors are with M.I.T. Lincoln Laboratory, Lexington, MA 02173.

0733-8716/83/1200-0963$01.00 ©1983 IEEE

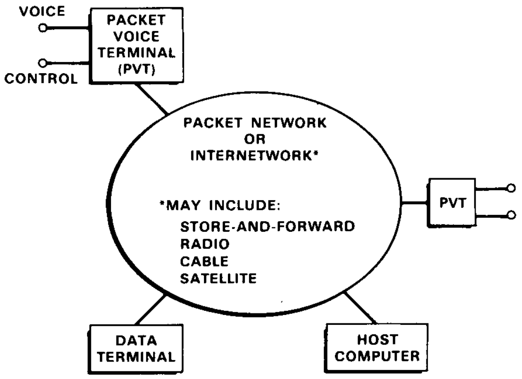

Fig. 1. Generic packet speech system configuration.

and the minimization of packet overhead and effective traffic control strategies to allow network links to be heavily loaded without saturation. System developments have been undertaken to help assess the implementation requirements for terminals and switching in a large-scale packet voice/data system, and efforts continue to drive down the size and cost of system components.

A series of packet speech experiments and system developments to address these issues has been conducted under the sponsorship of the Defense Advanced Research Projects Agency (DARPA). These efforts were initiated in 1973 by Dr. R. E. Kahn of the DARPA Information Processing Techniques Office (IPTO), who has provided leadership and numerous technical contributions through the course of the work. As will be noted in this paper and in the references, numerous individuals in several organizations have made significant contributions to the system development and experiments. The purpose of this paper is to review and evaluate the experience gained so far from these efforts in packet speech systems experiments. The perspectives and conclusions are the responsibilities of the authors and are necessarily influenced by the specific involvement of ourselves and our colleagues at Lincoln Laboratory.

This paper will begin by describing basic concepts and issues associated with packet speech systems. A generic packet speech system configuration will be described, and requirements and techniques for digital speech processing, protocol functions, packetization and reconstitution, conferencing, and multiplexing will be discussed. With this as a point of reference, the experimental system configurations and key results for packet speech on the ARPANET, SATNET, and wide-band system will be described.

II. Packet Speech Concepts and Issues

The purpose of this section is to set a general framework for the descriptions of specific experimental packet speech systems to follow in subsequent sections.

A. Generic Packet Speech System Configuration

A generic packet speech system configuration is depicted in Fig. 1. The interface between the user and the network is provided by a functional unit referred to as a packet voice terminal (PVT) [22]. The PVT may, but need not, be implemented in a single physical unit dedicated to a single voice user. Functionally, the user interfaces with the PVT much as with an ordinary telephone set, and the PVT interfaces with the packet network. In addition to being able to talk and listen, the user is provided with a full range of control and signaling capabilities including dialing and ringing. Both control signals and voice are transmitted from PVT to PVT over the network in digitized packet form. The resources of the integrated voice/data packet network are shared statistically with data traffic among host computers and data terminals as well as with other voice users. The packet network may be of the original store-and-forward type as exemplified by the ARPANET; may utilize packet radio, cable, or satellite techniques; or may be composed of an internetwork combination of these various types of packet nets, connected by gateways.

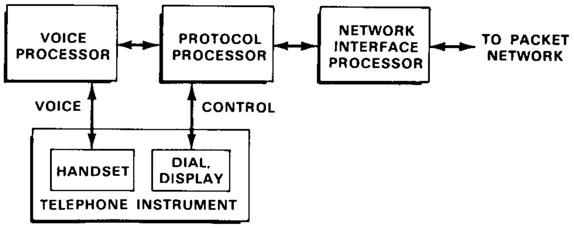

Fig. 2. Functional block diagram of packet voice terminal.

B. Generic Packet Voice Terminal Configuration

A functional block diagram of a packet voice terminal is shown in Fig. 2 which shows three major functional modules each associated with a processor. It is not necessary to use separate processors to achieve the functional modularity, but we have done so in the microprocessor PVT implementation [22] discussed later and find it convenient to use the same terminology here. The voice processor converts between analog and digital speech at digitization rates typically varying from 2 kbits/s to 64 kbits/s, and marks each *parcel* (typically 20–50 ms of speech) as containing either active speech or silence.

The protocol processor is the primary controlling module of the PVT. The protocol processor includes an interface with the user dial/display and must generate and interpret the packets necessary for establishing the call. The protocol processor provides the basic interface between the synchronous voice coding/decoding process, and the asynchronous packet network. The buffering and reconstitution algorithms to produce steady speech to the listener are implemented in the protocol processor.

The network interface processor provides the network-dependent packet transport mechanism. Ideally, all network-dependent hardware and software would be contained in this module. In practice, we have found it difficult to maintain this pure modularity because of a need to incorporate network-dependent elements into the packetization and reconstitution processes in the protocol processor.

The telephone instrument provides the simplest user interface to the PVT. The flexibility of the packet system allows the possibility of a wide range of user functions and displays, which can exceed the signaling capability of the telephone instrument. In some experiments, computer terminals have been used to augment the user interface.

An important development in the work we will describe on packet voice is the evolution of the PVT from implementation on large general-purpose computers to compact microprocessor-based systems. In our view, this development is essential in making packet voice practical and affordable. We have generally focused on a distributed approach where each separate PVT performs complete voice processing and protocol functions for one user. A more centralized approach is also possible, where a single facility would simultaneously perform the functions of a number of PVT's for multiple users.

C. Digital Speech Processing Functions

The primary voice processing function for packet speech is speech digitization. Two other important voice processing functions are also noted here—speech activity detection and echo control.

1) Speech Encoding Algorithms: Speech is a compressible source [34] that can be coded at rates ranging from 64 kbits/s to below 2.4 kbits/s. Recent packet experiments have made use of the pulse code modulation (PCM) widely used in digital telephony, but all the earlier work described in this paper used encoding techniques [36] such as CVSD (continuously variable slope delta modulation) or LPC (linear predictive coding [37]) to provide data rates low enough for use on the networks that were available for experimentation.

Packet systems offer flexibility for taking advantage of speech encoders at a variety of rates. The PVT may include a variety of (fixed) speech bits rates, which could be selectable at dialup according to network load. More complex coding schemes [42] can be applied which vary transmission rates according to the time-varying compressibility of the speech signal. Or multirate "embedded coding" algorithms [38], [39] can be used to allow rapid adaptation [33] of voice bit rates to network conditions which may vary during a call. Selection of a speech coding algorithm [35], [36] for a given application depends on many factors including network bit rate constraints, speech quality needs, noise or distortions on the input speech, and terminal cost and complexity constraints.

2) Speech Activity Detection: A key advantage of packet speech is the ability to save bandwidth by transmitting packets only during talkspurts. Therefore, accurate discrimination between speech and silence, or speech activity detection (SAD), is an essential voice processing function [43]–[45]. The SAD algorithm must minimize the average percentage activity, but also meet tight constraints on the fraction of lost speech. SAD, in a laboratory or quiet input speech environment, is relatively straightforward. But when the speaker is in a noisy environment, or when the speech originated in the switched telephone network (STN), the design of effective SAD algorithms is more difficult.

In our system model, SAD is performed in the voice processor, which marks parcels delivered to the protocol processor as silence or speech. The protocol processor would normally packetize and transmit only the speech parcels except that it may transmit additional parcels at the beginning and end of a talkspurt to improve speech quality. Such a "hangover" at the end of a talkspurt is commonly used to include weak final consonants in a talkspurt and to bridge across short gaps.. An "anticipatory" parcel at the start of a talkspurt can give a smoother startup and is easy to provide in a packet system since the required buffer space is already present for use in the packetization process.

3) Echo Control: Echo control is not needed in a pure packet speech system in spite of the delays that may be present since the system is fully digital and provides isolation between the two directions of voice transmission for the entire path between sending and receiving handsets. However, echo control becomes an issue if we wish to interconnect a packet network and the common STN. Techniques for controlling echos [46], [47] include 1) echo suppression, generally aimed at passing speech in only one direction at a time; and 2) echo cancellation, which attempts to adaptively cancel echos and maintain full duplex speech. Echo cancellation is generally the preferred, but more costly, technique. Echo canceller chips which reduce the cost are becoming available. If the generic PVT were to be used to interface with the STN, it could be equipped with an echo canceller as part of its voice processor, to cope with echoes caused by the two-wire local loop in the STN. Both echo suppression [57] and cancellation [54] have been used in STN interface experiments on the wide-band network.

D. Packet Speech Protocol Functions

The development of the ARPANET as a packet communication resource was quickly, and by necessity, followed by the development of a set of protocols (i.e., rules for conducting interactions between two or more parties) to organize and facilitate use of this resource for a variety of applications. A network control protocol (NCP) was developed to allow controlled packet communication among processes running in dissimilar host computers [9]. Higher level protocols were developed to serve specific user needs. These included TELNET for terminal access to remote computers and file transfer protocol (FTP) for transmission of large files. Both TELNET and FTP obtained access to the network through NCP. This technique of *protocol layering* to partition and organize the task of providing various levels of communication services has been a fundamental aspect of the development of packet communication systems [12].

The original ARPANET protocols were designed to provide very reliable end-to-end packet delivery either at high throughput (e.g., FTP) or low delay (e.g., TELNET). Both NCP and the basic node-to-node protocols imposed end-to-end flow restrictions which included retransmissions when necessary to reliably deliver all the packets and worked against the simultaneous achievement of high throughput and low delay. But for real-time voice communication, both high throughput and low delay are needed. Some reliability may be sacrificed, as a small percentage of lost packets is tolerable. Therefore, new protocol developments were needed for packet voice.

The initial work on packet voice protocols focused around the development of a high-level protocol known as the network voice protocol (NVP). Dr. D. Cohen of the Information Sciences Institute (ISI) was the chief architect of NVP [16], [17]. Functions of NVP include

1) call initiation and termination, including negotiation of voice encoder compatibility and handling of ringing and busy conditions;

2) packetization of voice for transmission, with the time stamps and sequence numbers needed for speech reconstitution at the receiver;

3) speech playout with buffering to smooth variable packet delays.

NVP is designed to pass its packets to a lower level protocol for transport across the network to meet real-time speech requirements. In order to avoid NCP's flow restrictions, NVP bypassed NCP for packet transport. In addition, modifications were made to the basic ARPANET transport protocols to provide an "uncontrolled" packet service which reduced packet flow restrictions between IMP's (see Section IV-B). The original NVP used the basic ARPANET (host–IMP and IMP–IMP) protocols directly to deliver its packets, and was independent of and generally incompatible with other protocols (e.g., NCP) in use at the time.

Since the original NVP made use of the ARPANET directly, extension to other networks (e.g., the Atlantic SATNET) required creation of a new protocol for each new network. This motivated the development of a second generation of voice protocols with a more general internetwork-oriented approach and with network-dependent aspects limited to the lowest level. Protocol functions were separated into two levels. The "higher" functions of call setup, packetization, and reconstitution, as well as dynamic conference control features, were incorporated into a second-generation version of NVP. The lower level protocol, which has come to be named "ST," provides an efficient internet transport mechanism for both point-to-point conversations and conferences. The name ST is derived from the work "stream" which refers to the type of traffic load that voice customers offer to a packet network. ST operates at the same level in the protocol hierarchy as IP, the DoD standard internet protocol [11] for datagram traffic. ST is designed to be compatible with IP. NVP may call on IP for delivery of control packets, and on ST for delivery of voice packets.

ST differs from IP in being a virtual circuit rather than a datagram protocol. Transmission of ST packets must be preceded by a connection setup process arranged by an exchange of control messages. During the connection setup, an internet route is established, and gateways along the path build tables pertaining to the connection. The preplanning involved in the connection setup and the existence of these connection-oriented tables allows ST to offer special services and efficiencies.

Fig. 3 illustrates how the current internet packet voice protocols relate to each other and to corresponding data handling protocols. Net I and net II designate individual packet networks, and might represent ARPANET, SATNET, or local area cable or radio nets. The situation depicted shows the protocol layers to be traversed in order for voice and data users on net I to communicate (through a gateway) with similar users on net II. The internet data file transfer protocol and the terminal-oriented protocol TELNET utilize a DoD standard transmission control protocol (TCP) for reliable packet delivery. TCP calls, in turn, on IP for packet transport. This is a departure from the original situation in the ARPANET, where FTP utilized NCP, which interfaced directly to the network. Similarly, NVP utilizes both IP and ST for packet transport; IP is used primarily in call setup situations, and ST is used for speech transport.

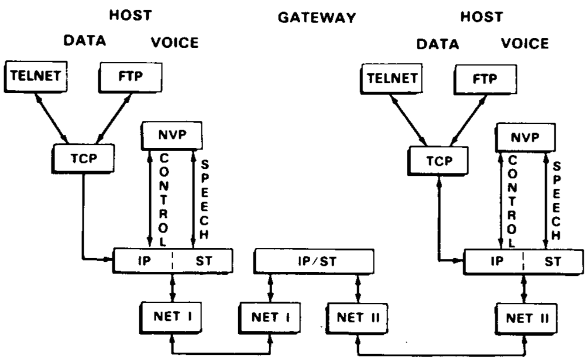

Fig. 3. Protocol hierarchy for internet packet voice and data communication.

E. Speech Packetization and Reconstitution

Packet communication necessarily involves both fixed components of delay due to transmission and propagation, and statistically varying components such as queueing delays in network nodes or in gateways. Additional varying delay components are caused by packet retransmissions to compensate for errors in delivery and by the possibility that all packets between a particular source and destination may not follow the same route. In addition to delay effects, some packets may be lost between source and destination. In this regard, a delay versus reliability tradeoff is possible where (for example) delays due to retransmissions can be reduced at a cost of an increase in percentage of lost packets.

The purpose of speech packetization and reconstitution algorithms [31] is to provide speech with 1) minimum overall end-to-end delay and 2) any anomalies caused by lost or late packets basically imperceptible to the listener. Ideally, the overall packet network would provide high enough link bandwidths and sufficient nodal processing power to keep delay and delay dispersion within tightly controlled limits. In such a case, very simple packetization and reconstitution algorithms in the PVT may suffice. However, in some situations where packet speech is required, it may not be possible to control network design. In particular, when there is a need to transmit speech over an existing packet data network, it may be necessary to use more elaborate algorithms.

1) Choice of Packet Size: Resolving the issue of packet size forces us to make some difficult compromises. In order to minimize both the packetization delay at the transmitter and the perceptual effect of lost packet anomalies at the receiver packets should be as short as possible. Experience with lost packet anomalies indicates that individual packets should ideally contain no more than about 50 ms of speech [31]; ideally, we would like packets to be even shorter to minimize packetization delay. On the other hand, in order to maintain high channel utilization, we would like to keep the number of speech bits per packet as high as possible relative to the overhead which must accompany each packet. This tradeoff is particularly difficult for narrow-band speech. For example, 50 ms of 2400 bits/s speech is represented by only 120 bits, which is less than the header size of many existing packet networks. For higher speech bit rates, relative packet overhead is less of a problem. An obvious conclusion is that future packet voice networks should be designed with minimum required header lengths.

The choice of packet size is also influenced by limitations on network throughput in packets/s. For the same user data rate, processing loads on network nodes will generally increase as packet size is decreased. This can force use of longer packets. For example, our typical range of packet sizes for real-time speech transmission across the ARPANET was 100–200 ms, corresponding to 5–10 packets/s because the network could not consistently sustain a higher rate. In some cases it may be desirable to adapt packet size to time-varying network conditions. In speech experiments conducted by SRI on packet radio nets (PRNET's) [20], [21] the radio provides channel availability information to the voice terminal which buffers speech and sends variable size packets depending on the intervals between opportunities for access to the networks.

2) Time Stamps and Sequence Numbers: To assist in the reconstitution process, it is desirable to include a time stamp and a sequence number with each transmitted packet. The time stamp allows the receiver to reconstitute speech with accurate silence gap durations in spite of varying delays between talkspurts. Incorrect gap durations can cause significant perceptual degradation in the output speech, especially for short gaps between syllables, or between words in a phrase. The time stamp also allows reordering of out-of-order packets at the receiver. The time stamp is derived by counting every speech or silence parcel generated by the voice processor. A few bits (we use 12) will suffice to cover a range of relative timing about twice the packet transit time dispersion range of the network.

The sequence number allows the receiver to detect lost packets whereas with a time stamp alone it would not be possible to distinguish silence gaps from packet loss. The detection of lost packets can be used by the receiving PVT to inform the listener (by playing out a distinct audible signal) that some speech has been lost. This can be particularly important if packets contain enough speech to include linguistically significant utterances (such as the word "not"). Detection of lost packets can also be used to allow the terminals to adapt bit rate and/or packet rate to network conditions.

If the network provides service with very short delays and very little delay dispersion, then satisfactory speech can be produced without either time stamps or sequence numbers. However, our experience, both with packet speech experiments and simulations, indicates that both time stamps and sequence numbers should be included.

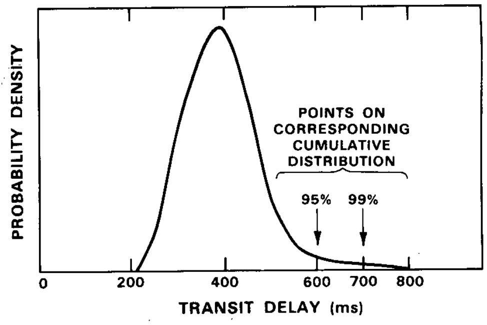

Fig. 4. Illustrative probability density function of transit delays in a packet network.

3) Reconstitution of Speech from Received Packets: The reconstitution algorithm has two major tasks, 1) it must buffer incoming packets and decide exactly when to play them out, and 2) it must decide what to play out when it has finished playing out a packet and the next packet is not available.

Fig. 4 shows an illustrative probability density function for transit delay in a packet network. The delay ranges shown are typical of some of our measurements on 10 hop paths through the ARPANET, but the points to be made are more general. In the case illustrated, 99 percent of the packets experience delays between 200 and 700 ms. Hence, a reconstitution delay (inserted at the receiver) of 500 ms would be sufficient to cover this spread. A 400 ms reconstitution delay would assure playout of 95 percent of the packets. Since some packets may be lost in the net, there is no value of reconstitution delay that can guarantee playout of all packets. Even if all packets did arrive, it would be undesirable to unduly increase delay to account for a few very late arrivals. The network's delay characteristics are generally not known in detail *a priori* and may vary with time. The degree of complexity to be built in to the reconstitution algorithm should be chosen based on the knowledge we do have of the network delays. A fixed reconstitution delay would suffice if network delays and delay dispersion are short. If delays are expected to be large or dispersions vary greatly with the network load, it would be desirable to use an adaptive algorithm (see [31] for an example of such an algorithm) to adjust the reconstitution delay to effect a compromise between packet loss and overall delay.

The other major reconstitution algorithm task is to decide what to play out when it has finished playing out a packet and the next packet is not available. This can result from a late or lost packet or it may simply indicate a pause in the talker's speech. Typically, the reconstitution algorithm has no way to distinguish these cases and should take the same action in either case. A number of fill-in strategies have been tried, including 1) filling with silence, 2) filling by repeating the last segment of speech data, and 3) filling with repeated frames of speech data which are made voice-

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase®
Smarter legal research.