Priority Discarding of Speech in Integrated Packet Networks

DAVID W. PETR, STUDENT MEMBER, IEEE, LUIZ A. DASILVA, JR., AND VICTOR S. FROST, MEMBER, IEEE

Abstract—**This paper deals with the control of short-term congestion, which we will refer to as overload, in packet networks containing a mix of data, speech, and possibly other types of signals. In such integrated packet networks (IPN's), overload control can be accomplished by taking advantage of the inherent structure of the speech signal. The proposed system model involves assigning a delivery priority to each packet (speech or otherwise) at the transmitter and discarding speech packets according to delivery priority at any point in the network in response to overload. This model attempts to minimize per-packet processing at network nodes, which will become increasingly important if IPN's are to evolve into broadband networks carrying speech, graphics, image, and video signals over fiber links. The quality of the received speech can be maintained by classifying speech segments according to their structure and coding them in a way that will ensure ease of lost packet regeneration at the receiver. We report the results of an experiment which has confirmed the general validity of this model from the standpoint of transmitter and receiver processing and subjective quality. Future work will focus on refining the transmitter and receiver signal processing techniques and developing packet discarding algorithms together with models for the performance of queueing systems involving such algorithms.**

I. Introduction

RESEARCH on telecommunications networks is beginning to focus on post-ISDN (integrated services digital network) architectures and capabilities such as an integrated services *packet* network [1] and broadband ISDN [2]. The economies and flexibility of integrated networks make them very attractive, and packet network architectures have the potential for realizing these advantages. However, the effective integration of speech and other signals such as graphics, image, and video into an integrated packet network (IPN) can rearrange network design priorities.

For example, in the first packet networks, the traffic consisted of delay-tolerant, relatively narrowband data, so the nodal processing required for each packet was only of secondary concern. However, time-critical, broadband signals like those cited above will demand broadband transmission links (such as optical fiber) and broadband packet switching. Of these two technologies, switching is likely to be the bottleneck in future IPN's. Although processing speeds will continue to increase, it will also be necessary to minimize the nodal per-packet processing requirements imposed by the network design. This is the motivation for new switching concepts like fast packet switching [3] and ATM [2].

However, the presence of these new signals can also provide new flexibility which can be exploited in the network design. Data signals must generally be received error-free in order to be useful, but the inherent structure of speech and image signals and the way in which they are perceived allows for some loss of information without significant quality impairment. This presents the possibility of *purposely* discarding limited information to achieve some other goal, such as the control of temporary congestion.

The research described in this paper is guided by both of these new principles for IPN design: minimal per-packet processing and flexibility due to signal structure. We focus initially on speech as the structured signal, but some of the techniques developed as a result of this research are likely to be applicable to graphics, image, and video signals as well.

The broad problem of effectively integrating speech into packet networks has many facets, but this paper focuses on the particular problem of overload control. In Section II, we develop a new system model for the problem of overload control in integrated packet networks. Section III describes an experiment designed to test some fundamental hypotheses of this system model. The final section presents conclusions and outlines future work.

II. A System Model for Overload Control in IPN's

A. Overload Control

1) Definition: In a packet network, overload is any "short" period of time (a few milliseconds) during which the arrival rate at a packet multiplexer exceeds the service rate by a sufficient margin to cause queue overflow. The control of overload can thus be contrasted with load management, that is, the control of the long-term average load on the network (e.g., by call denial, throughput negotiations, etc.). The contrast is in both scope and time frame. Overload control applies to individual packet multiplexers in the network, whereas load management applies to the network as a whole, or at least to subsets of network paths.

Manuscript received April 10, 1988; revised March 18, 1989. This work was supported in part by AT&T, an NSF Presidential Young Investigator Award under Grant ECS 8352173, and a University of Kansas General Research Allocation 418-3510-20-0038. This paper was presented in part at the 1989 IEEE INFOCOM Conference, April 24-27, 1989.

D. W. Petr and V. S. Frost are with the Telecommunications and Information Sciences Laboratory, University of Kansas, Lawrence, KS 66045.

L. A. DaSilva Jr. is with IBM, Rio de Janeiro, Brazil.

IEEE Log Number 8927592.

0733-8716/89/0600-0644$01.00 © 1989 IEEE

In addition, the time frames (e.g., for measurement, decision, and action) associated with overload control are shorter than those for load management by at least one to two orders of magnitude.

To control overload, it is necessary to have some indication of *impending* queue overflow. For the purposes of discussion, we assume that the queue consists of a finite number of packet buffers and then provide a mathematical definition of overload based on a simple form of prediction. We begin with the following notation:

τ:	prediction horizon (order of milliseconds or tens of milliseconds)
t_0:	current instant
$\lambda(t_0)$:	estimate of arrival rate at time t_0 (packets per second)
$\mu(t_0)$:	estimate of service rate at time t_0 (packets per second)
$q(t_0)$:	number of packets in queue (queue fill) at time t_0
Q:	maximum queue capacity (packets).

We now define overload as

$$q(t_0) + \left[\lambda(t_0) - \mu(t_0)\right] \cdot \tau > Q. \qquad (1)$$

This differs from the definition in [4] in that it incorporates queue fill and capacity in addition to arrival and service rates.

2) Discussion: All packet networks must have a means of dealing with overload, since the statistical nature of these networks ensures some finite overload probabilities. References [5] and [6] point out that the presence of multiple speech sources in a packet network can serve to worsen the overload problem by increasing the burstiness of the composite traffic. Of course, if the average load on the network is kept low enough, the probabilities of overflow in the packet multiplexer queues can be made arbitrarily small. The goal of intelligent overload control, however, is to *maximize* the average network load, subject to some service quality constraints. Effective overload control is thus especially important if high-performance, high-capacity, integrated networks are to be realized.

Because of the short overload control time frames, distributed control mechanisms, in which each overload point acts autonomously, are more attractive than centralized algorithms. Minimal per-packet processing at each switching node is also desirable if switches are to operate under the low-delay, high-capacity constraints necessary for large IPN's.

3) Overload Control Options: Given the previous discussion, it is clear that traditional window-based or stop-and-wait flow control techniques (either link-by-link or end-to-end) are undesirable as overload control mechanisms. Here we explore some other possibilities and discuss the applicability of each to data and speech packets. For data packets, we will assume no *a priori* knowledge of the structure of the information which they contain. However, the inherent structure of coded speech signals can be effectively exploited.

Note that the only variables in (1) which can be manipulated dynamically for overload control are the effective arrival rate (λ), the effective service rate (μ), and the queue fill (q). Assuming that the bytes/second service rate (transmission rate of the outgoing link) is fixed, the following four approaches are possible for overload control.

a) Discard Arriving Packets When Queue is Full: The default approach taken by many networks is to discard arriving packets only when absolutely necessary, i.e., when the queue is full. In this approach, τ (the prediction horizon) approaches zero and overload is controlled by reducing the effective arrival rate (λ). However, the only controlling parameter is queue capacity (Q), which is limited by the delay requirements for speech. This technique is clearly applicable to both speech and data packets, which would be discarded without bias.

This approach fails to recognize that some information may indeed be more important than other information, i.e., that the information *should* be discarded with bias. In coded speech, for example, active speech is usually more important to the communication than background noise during pauses. Some data packets may be more important than others by virtue of a greater network "investment" in them (e.g., number of nodes already traversed). We call this principle *priority discarding*, in which *minimal* short-term signal degradation is deliberately introduced in order to improve overall network performance. An early example of priority discarding is found in the older T1 multiplex systems, in which the *least* significant bit of the PCM speech samples is periodically discarded to provide signaling information.

The remaining three approaches to overload control incorporate this principle of priority discarding.

b) Reduce Source Packet Generation Rate: It is also possible to reduce the effective arrival rate by reducing the packet generation rate at the source. However, for this second technique to be useful for overload control, the overload point would have to be able to *directly* control the source generation rate. This is possible in local area networks (for both speech and data packets), but not in more general multinode networks.

Priority discarding using this technique was applied to a local area network in [7]. An embedded speech coding algorithm [8] was used to divide each speech sample into "essential" bits and "enhancement" bits. During overload, the (fixed-length) packet generation rate was reduced by discarding "enhancement" bits and including more samples in each packet. This was shown to be an effective overload technique in that it allowed significantly more speech sources (higher average load) for a given delay performance, with only slightly degraded speech during overload. A similar technique could be applied to systems with a speech activity detector by suppressing packets containing background noise *only* during overload periods.

We reiterate, however, that this technique is not applicable to general multinode IPN's.

c) Reduce Packet Size: Packet size could also be reduced, in which case the effective packet/second service rate [μ in (1)] is increased to reduce the overload. This third technique is clearly not applicable for data, but could be applied to speech. For example, in [3] embedded coding was again used, but speech packets arriving at a queue were *shortened* in response to overload by discarding "enhancement" bits. Again this technique was shown to significantly improve the performance of an IPN in terms of delay versus load (number of simultaneous speech sources), with only a slight reduction in perceived quality.

Packet shortening is quite amenable to a general network environment, since each overload point can exercise control autonomously. However, it has the serious disadvantage of requiring network nodes to "know" the internal structure of the packet body and to manipulate packet contents. This violates two important IPN design principles: protocol separation (layering) and minimal per-packet processing.

d) Discard Packets According to Delivery Priority: The fourth approach is to discard packets according to delivery priority. This differs from the first (default) approach in that the prediction horizon does not approach zero (i.e., packets may be discarded before the queue is full) and priority discarding is practiced. It is probably easier to discard arriving packets and thereby control arrival rate (λ). However, more control flexibility is possible by allowing previously queued packets to be discarded, controlling queue fill ($q(t_0)$). We assume only that relative importance for all packets (speech and data) is indicated by a "delivery priority" in the *network* portion of the packet header [2].

Priority discarding was applied to speech packets in [4] and [9]. Methods of determining packet delivery priorities included: embedded coding, with the coded bits for a segment of speech segregated into *separate* packets according to their importance to the decoding process; even-odd samples (see discussion of [10] below); and multiple energy detection thresholds (silence, semi-silence, active speech). Packet multiplexers could then autonomously respond to overload (as measured by queue fill thresholds or number of active sources) by simply discarding entire packets in priority order. The performance of delay versus load (number of speech sources) improved in varying degrees according to the particular combination of techniques.

e) Information Regeneration: When information is discarded by the network, it may need to be regenerated. With data packets, the information can be exactly regenerated through standard error recovery protocols. Speech, however, has many more options since the information does not need to be *exactly* regenerated. Some methods of speech regeneration include: replacing a lost packet with silence or the previous packet [11], [12]; ignoring the lost information, when embedded coding is used [4], [9]; sample interpolation, when samples are numbered modulo N and samples with the same residue are grouped together in packets [10], [13]; and regeneration based on information contained in previously received packets, for example, pattern-matching and pitch-synchronous replacement [14].

B. A New IPN System Model with Priority Packet Discarding

1) Overview: Our goal is to construct a model which considers the entire IPN (transmitters, packet multiplexers, and receivers) as a system to be optimized for higher speeds and capacities. As such, we allow more complex processing at network edges (transmitters and receivers) in order to simplify the processing at network nodes. In our model, the transmitter forms packets which vary in their importance to the reconstitution of high-quality speech at the receiver. This level of importance is indicated as a "delivery priority" in the network portion of the packet header. Packet multiplexers discard speech packets according to this delivery priority in order to control overload. The receiver then attempts to regenerate the information contained in any discarded packets. At this level of description, the approach is similar to [4] and [9], but we take a different approach in the processing and regeneration of the speech, as discussed below. Although our model is concerned specifically with speech, the approach can be extended to other structured signals such as graphics, image, and video signals.

We will specifically exclude techniques which distribute information about a single segment of speech into several (say N) packets [4], [10], [13]. Such approaches have a fundamental N-to-1 disadvantage compared to our single packet per segment approach. They either produce N times as many packets to be processed at each network node (if segment size is the same) or N times as much packetization delay (if the number of packets produced is the same). In addition, these approaches fail to take advantage of the relative importance of different segments of speech.

2) System Description:

a) Transmitter Subsystem: In our system model, the transmitter (Fig. 1) first classifies speech segments according to models of the speech production process (e.g., voiced sounds, fricatives, and plosives). This model-based classification is used to remove redundancy during coding, to assign delivery priorities, and to regenerate discarded speech packets.

After classification, the transmitter removes redundancy from the speech using a coding algorithm based on the determined model. For example, voiced sounds (e.g., vowels) could be coded with a block-oriented pitch prediction coder. After coding, the transmitter assigns a delivery priority to each packet based on the quality of regeneration possible at the receiver. The assignment depends partly on the class of speech packets and is de-

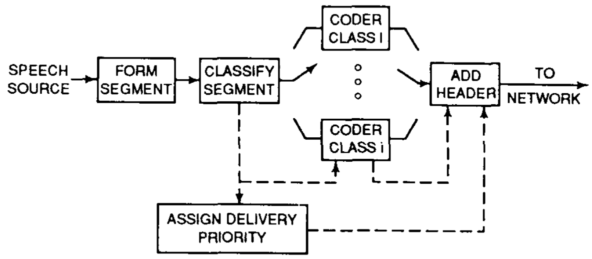


Fig. 1. Transmitter subsystem.

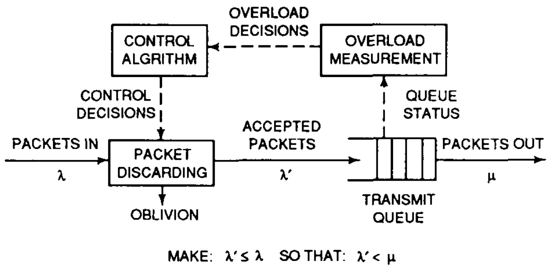


Fig. 2. Packet multiplexer subsystem (arriving packets discarded).

termined in some other fashion for data packets (perhaps by giving all data packets the same high delivery priority).

In forming packets from speech segments, the delivery priority would be included in the network portion of the packet header (for example, the ATM header [2]) and the classification and any coding parameters would be included in the end-to-end portion of the header (ATM adaptation header). In contrast to speech-detector-based transmitters used in all previously cited studies except for [7], all segments of speech (even background noise) are "launched" into the network and only discarded by the network for overload control. We will expand on this point in the discussion of the receiver subsystem.

b) Packet Multiplexer Subsystem: Packet multiplexers (Fig. 2) exist at each outgoing link of each network node as well as at each multiplexed network access point. Each packet multiplexer monitors local overload and discards packets according to packet delivery priority (read from the network portion of the packet header) and some locally determined measure of overload level. Specific measures of overload and discarding algorithms are the subject of continuing research by the authors. Fig. 2 assumes that arriving packets are discarded, but it would also be possible to discard already-queued packets.

In addition, *if* error checking is performed by the nodes, *any* packet (data or speech) found to have an error is discarded. Some experimental IPN's (e.g., [3]) check for packet errors in data packets but only header errors in speech packets, assuming that it is better to deliver an errored speech packet than to discard it. Our unified approach to error control is justified since a sophisticated speech packet regeneration mechanism is built into the receiver.

This model requires very little per-packet processing for overload control at the packet multiplexer and the processing (including optional error control) is uniform for all packets. This should result in significantly faster switching nodes relative to other approaches.

c) Receiver Subsystem: The receiver (Fig. 3) decodes the samples in speech packets delivered to it (based on the classification and coding parameters contained in the end-to-end header) and determines the appropriate time to play them out. By choosing to launch all speech packets (including background noise) at the transmitter, the receiver synchronization problem requires only packet sequence numbers. This is a significant feature in light of the difficulties with the alternatives: global synchronization is administratively difficult and relative time stamps must be modified at each packet multiplexer, requiring additional per-packet processing [15]. Furthermore, potential speech detector impairments (such as clipping) are eliminated whenever the network is not overloaded. Even during periods of considerable overload, the received quality may be better if at least a few "background noise" packets are delivered and then used to regenerate noise which is similar in character to the actual noise.

If a packet is lost for any reason (discarded by the network because of overload or errors, excessively delayed in the network, etc.) the receiver must first detect the loss by inspecting sequence numbers of those packets which are received. It must further make a determination of the class of each lost packet, so that the appropriate regeneration model can be applied using previous header and sample history. A correct class determination will be critical to accurately regenerating the lost information, but this is easily done with our model as follows. In a string of packets with the same class, we can virtually ensure that the *first* packet will be received by assigning it a high delivery priority. In we assume perfect delivery of these first packets, the class of any lost packet will match the class of the last received packet. Thus, the receiver's class decisions can be virtually error-free.

3) Summary of Advantages: We now briefly summarize the advantages which we gain by taking a total system approach to this problem. The model

1) provides a powerful overload control mechanism
2) exploits the structure of speech effectively
3) allows extremely simple per-packet processing for overload control
4) requires only one packet per speech segment
5) simplifies receiver speech synchronization
6) allows reduced per-packet error processing at packet multiplexers.

III. An Experiment in Priority Packet Discarding

A. Overview

We formulated the experiment described in this section to determine the feasibility of the overall system model and to identify potential problems. More specifically, we had the following goals for the experiment:

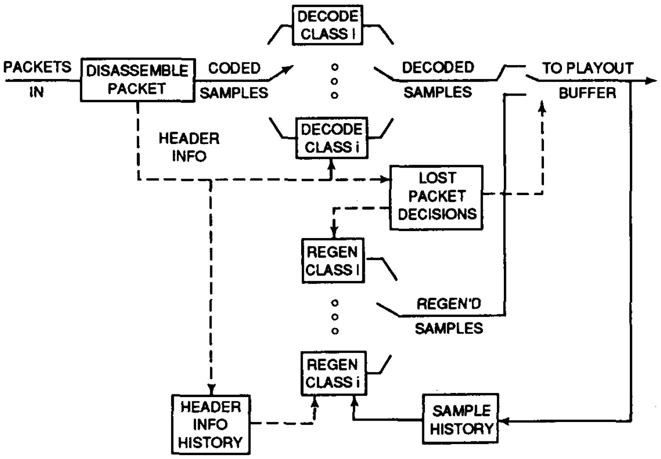


Fig. 3. Receiver subsystem.

1) To determine, for each experimental "delivery priority," the maximum packet loss rate for which the reconstructed speech is *indistinguishable* from the original speech. We emphasize that this is not only more stringent than an *acceptable* packet loss rate criterion [4], [10], [13], [16], but it has the additional advantage of being independent of application (e.g., public network versus military applications).

2) To determine whether separating the speech into production-model classes for coding and regeneration can produce different subjective results (maximum loss rates) for the different classes. Since the delivery priorities are strongly related to the classification, these differences would also yield an appropriate delivery priority order.

3) To gain a better understanding of the extent of interactions (desirable or undesirable) between the coding and lost packet regeneration techniques used for each class in the test.

4) To compare the combined quality of the coding techniques to standard 64 kbit/s PCM in the absence of lost packets.

All transmitter, node, and receiver functions were simulated in nonreal time with general-purpose computing equipment. As a controlled experiment, packets were discarded from only one "delivery priority" at a specific, fixed rate for each experimental condition. Subjective isopreference tests were then conducted to determine, for each delivery priority, the packet loss rate at which the resulting speech was perceptually different from the original digitized utterance.

B. Description of Signal Processing

In this experiment, a segment size of 64 samples was used (8 ms of speech at 8 kHz sampling), and the packet length equaled the segment size. This is on the lower end of the range given in [17] for an optimal packet length for packet speech networks. Longer packet lengths would effectively decrease the packet processing at network nodes, but are expected to be more difficult to regenerate when lost. The particular packet size is, however, of secondary importance here since we are primarily interested in demonstrating fundamental concepts.

1) Classification and Delivery Priority Assignment: Classification is important in our model, but not as critical as in some other applications. This is true partly because we may allow ourselves the luxury of including a "catch-all" class, i.e., a class in which to place segments which cannot be reliably placed anywhere else. In this experiment, we used a nonoptimized, binary tree classifier to divide the segments into four classes based on speech production models: background noise, voiced speech, fricative speech, and "other speech." Besides being used for coding and lost packet regeneration, the classification influenced the assignment of a "delivery priority" to each packet.

a) Description of Classes: The background class was intended to consist of segments containing only background noise from the speaking environment, from transmission systems, or from any other source. The voiced class contained sounds (e.g., vowels) generated by the vibration of the vocal chords resulting in quasi-periodic pulses of air which are acoustically filtered by the vocal tract. The target sounds for the fricative class were those noise-like sounds resulting from air flow turbulence caused by a constriction in the vocal tract. The "other speech" class was intended to contain sounds which cannot be characterized as background, voiced, or fricative. This could include segments containing a transition from one class to another and short transient sounds such as the plosives ([p], [t], [k], etc).

b) Classifier Feature Set: The classifier used in the experiment was based on four features extracted from each segment. We provide here a general description of each feature and its function; precise feature definitions are given in Appendix A.

Feature $F1$, a signal level feature, was the ratio of the peak value in the current segment of S samples to the value of the minimum segment peak over the last H segments.

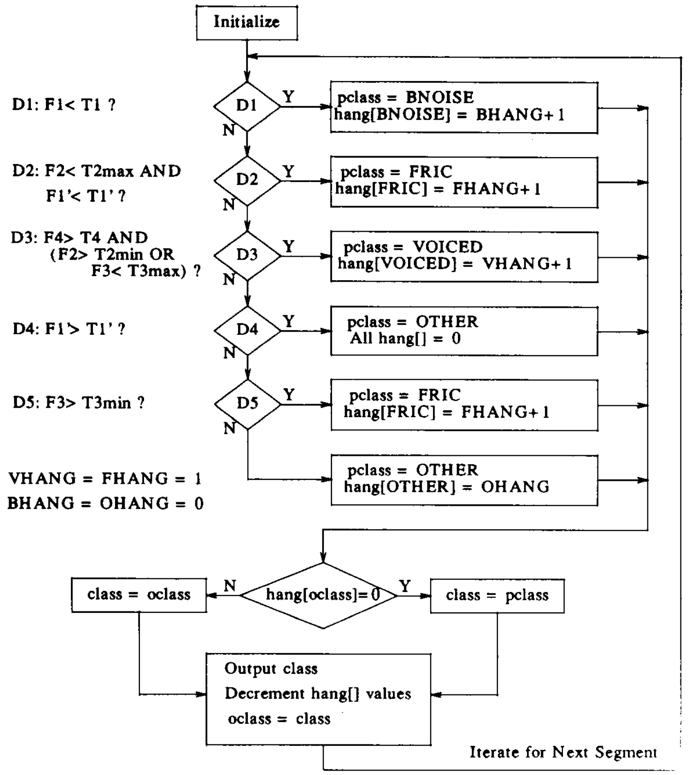


Fig. 4. Automatic classifier logic.

This was essentially a block-oriented version of the "silence compression" technique described in [18]. The rate of change of $F1$, $F1'$, was also used. Feature $F2$ was the single-sample autocorrelation coefficient. $F3$ was a new feature that we call normalized entropy, which indicates the uniformity of the probability distribution (on a log PCM basis) of the samples in the segment. It was intended to measure the "noise-likeness" of the signal. $F4$ was intended to measure the degree of periodicity of the signal. It was essentially a measure of the depth of the nulls in the AMDF function (see discussion of coding below) used to estimate the pitch period for voiced speech.

c) Classifier Logic: The basic structure of the classifier logic was a binary decision tree, with decisions made by comparing feature values to thresholds. It also incorporated a simple hangover mechanism, in which a previous classification was allowed to override the current one. Fig. 4 shows a complete flow diagram of the classifier logic. A preliminary classification (pclass) was determined from a tree of five binary decisions, $D1$-$D5$. Decision D1 isolated background noise, $D2$ isolated high-frequency dominated fricatives, $D3$ isolated voiced speech, $D4$ isolated plosive sounds (for which the hangover function was disabled), and $D5$ separated any remaining sounds into lower frequency fricatives and "other speech." After the preliminary classification, hangover values were allowed to override the preliminary classification. A single segment hangover was used for voiced speech and fricative speech (VHANG = FHANG = 1), and no hangover was used for the background and "other speech" classes (BHANG = OHANG = 0). Appendix B contains a description of classifier "tuning" and an assessment of its performance.

d) Delivery Priority Assignment: An *ordered* delivery priority assignment for the different packet types was not possible in this experiment, since one goal of the experiment was to *determine* such a priority ordering. The priority assignment was thus essentially a separation of the packets into "delivery groups" which would each be tested for tolerance to packet loss. The default means of separating into delivery groups was by the classification, resulting in four delivery groups which we designate W, X, Y, and Z.

However, we expected at the outset that the *first* packet in a string of packets, particularly voiced or fricative packets, would be critical to the regeneration of packets lost later in the string (see discussion of regeneration below). Since we also hypothesized that the "other speech" class would be the most important of the four classes for high-quality reconstructed speech, we placed these initial packets in the delivery group corresponding to the "other speech" class rather than in their default group. Table I summarizes the delivery group assignments.[1]

[1] Of course, other groupings would be possible, including placing *all* the "initial" packets in a separate group.

TABLE I
DELIVERY PRIORITY (GROUP) ASSIGNMENTS

Delivery Priority Label	Delivery Priority Contents
W	Background Noise Packets
X	Non-Initial Voiced Packets
Y	Non-Initial Fricative Packets
Z	"Other Speech" Packets Initial Voiced Packets Initial Fricative Packets

2) Model-Based Coding of Speech Packets: After speech segments (packets) were classified as described above, the transmitter applied a coding algorithm based on the appropriate speech production model to remove redundancy and provide information for lost packet regeneration.

a) Background Noise: The "production model" used for background noise was noise of unknown spectral shaping. In addition, background noise segments contain considerably less power than segments belonging to any of the other classes, and the reconstructed background noise required less fidelity. Therefore, we encoded background noise segments using 2-bit adaptive PCM (APCM) with a Jayant adaptive quantizer [19]. In order to prevent the propagation of errors after a lost packet, the end-to-end header contained the quantizer step size to be used in the encoding of the first sample in the packet.

b) Voiced Speech: The production model used for voiced speech was a low-frequency periodic signal spectrally shaped by the vocal tract. We matched our coding of voiced speech to this model by using 4-bit adaptive differential PCM (ADPCM) with a single-tap short-term predictor followed by a single-tap long-term (pitch) predictor. Using the following notation,

$s(k)$: value of speech sample at time k
$e(k)$: value of the prediction residual at time k
a: value of short-term predictor coefficient
$b(k)$: value of pitch predictor coefficient at time k
L: estimated pitch period (in samples) for current segment

the residual can be expressed in analytical form as

$$e(k) = [s(k) - a \cdot s(k-1)] - b(k) \cdot [s(k-L) - a \cdot s(k-L-1)]. \quad (2)$$

The residual was again quantized adaptively as reported in [19].

In our scheme, the short-term predictor coefficient was fixed while the pitch predictor coefficient was adapted according to a modified gradient algorithm. The adaptation algorithm can be described as

$$b(k) = b(k-1) + G \cdot \operatorname{sgn}[e_q(k-1)] \cdot \operatorname{sgn}[s_q(k-L) - a \cdot s_q(k-L-1)] \quad (3)$$

where G is an adaptation gain constant and the q subscript denotes quantization.

The pitch period was estimated from the average magnitude difference function (AMDF) for each segment, as in [20]. The speech signal was first low-pass filtered and center-clipped [21] to enhance the periodicity nulls in the AMDF, and the resulting AMDF minima were used to estimate the pitch period. Details of the voiced coding algorithm can be found in [22].

For error containment purposes, the quantizer step size and pitch predictor coefficient used to encode the first sample in the packet were included in the end-to-end header, along with the pitch estimate.

c) Fricative Speech: The production model used for fricative speech was broadband noise shaped by the vocal tract. Since the vocal tract shaping can often be modeled by a single resonance, we used 4-bit ADPCM coding with a two-tap short-term predictor. The coefficients were adapted with a modified gradient algorithm similar to (3). The same adaptive quantizer was used for fricative and voiced speech.

The header of fricative packets contained the values of the initial quantizer step size and the predictor coefficients used to encode the first sample in the packet.

d) Other Speech: Of the four classes, we knew the least about the properties of the "other speech" class, since it was by definition a "catch-all" class. Hence, we simply encoded these segments using 8-bit μ-law PCM.

3) Network Simulation: In this experiment, the network simulation was kept very simple in order to isolate and study the effect of packet loss rates on *individual* delivery priorities. Thus, the network simulation consisted merely of discarding packets from a *single* delivery priority at some fixed, nominal rate. The discarding events were independent, which does not model more bursty (and perhaps more likely) packet discarding processes. However, in the absence of actual discarding algorithms for network packet multiplexers and accompanying analysis, it is the only process which is reasonable.

4) Lost Packet Regeneration: For regenerating lost packets, we assumed that the receiver was able to perfectly estimate the class of each lost packet (see the previous discussion of the receiver subsystem). The receiver then regenerated the lost packet according to its class, as explained in the following.

a) Background Noise: For background noise packets, a sample history of the last 1024 received background noise samples was compiled and constantly updated. Even during overload periods, some of the background noise packets will be delivered and added to the sample history. When a packet was lost, the receiver randomly chose a segment of 64 consecutive samples from this sample history and used it as a replacement. For stationary background noise, the regenerated noise should be similar in character to the lost background noise.

b) Voiced Speech: For lost voiced speech packets, the receiver used the pitch period L in the last received voiced packet as the estimate of the pitch in the lost packet. The receiver then repeated the last L samples periodically throughout the packet, weighted by a gain factor C_j, calculated as

$$C_j = \frac{\sum_{k=(j-1)S}^{jS} s(k) \cdot s(k-L)}{\sum_{k=(j-1)S}^{jS} s^2(k-L)} \tag{4}$$

where S is the number of samples in a segment (packet), j is the number (index) of the last received voiced packet, and C_j is constrained to the interval [0.75, 1.25]. We did not attempt to apply smoothing at the boundaries of the regenerated packet [14] since no improvement could be verified in subjective tests [16] and since smoothing increases the delay considerably.

c) Fricative Speech: For fricative sounds, we would expect the prediction filter at the transmitter to produce a whitened residual. So, when a fricative packet was lost, the receiver generated a train of zero-mean, Gaussian white noise samples weighted by the power in the last received fricative packet, and then passed them through a two-tap predictor to provide appropriate spectral shaping. The predictor coefficients should be the same as used in the transmitter; we estimated their values by using the most recently adapted receiver coefficients, which were then "frozen" for regenerating the lost packet.

d) Other Speech: Since the properties of the "other" packets were not well defined, a simple packet repetition strategy was used for lost "other" packets; that is, the most recently received packet of any kind was substituted for the missing one.

C. Subjective Test Description

The performance of the coding and lost packet regeneration algorithms was evaluated by means of a subjective listening test. Since our purpose was to determine the point at which a processed utterance is distinguishable from a reference (unprocessed) utterance, we chose the isopreference testing method [23]. In such a test, the reference and processed utterances are designated A and B in each listening condition, with probability 0.5 that the reference condition is designated as A. For each listening condition, the subjects are presented with the utterances in an $ABAB$ order and then must indicate whether A or B is preferable as a source of information.

1) Procedural Details: Twenty-four subjects were used, all of whom were university students, and English was the native language of 15 of the subjects. The tests were administered in three sessions on three different days, and in each session the subjects were divided into two listening rooms. Later analysis of the results using Z tests indicated that all populations were homogeneous at a 0.01 significance level.

The source material for the test consisted of the sentence "The hogs were fed chopped corn and garbage." This particular sentence was chosen because it contains a good mix of voiced, fricative, and plosive/stop sounds, as evidenced by the results of the delivery priority assignment shown in Table II. The sentence was recorded by a male and a female speaker in a quiet room using a linear microphone handset and immediately digitized using standard μ-255 PCM at an 8 kHz sampling frequency. The average active speech power (not classified as background noise) to average background noise power for these original recordings was 45.7 dB.[2] We denote this particular signal to noise measure as S/N_b. For some of the conditions, more background noise was added to the PCM speech file resulting in a 32.1 dB S/N_b. The additional noise was recorded in a laboratory containing several microcomputers generating fan and disk noise. This type of background noise is more difficult to match at the receiver than white noise. The reference utterance for each listening condition was one of these PCM recordings.

2) Test Conditions: A *listening condition* is a particular AB pair presented to the listening subjects for which a subjective response, or vote, is obtained. The listening conditions were presented to the subjects over standard telephone handsets at a comfortable listening level. Each *processing condition* aggregates four listening conditions: two instances each for the male and the female speaker, where the second instance differs from the first only for the packet loss conditions, and then only in the particular pattern of lost packets.

Using this terminology, the test consisted of 32 processing conditions: two reference conditions, two coding conditions, five noise conditions, and 23 packet loss conditions. The two reference conditions compared the PCM coded utterance to itself for low and high background noise. These conditions served as the zero packet loss references for the packet loss conditions. The two coding conditions compared the PCM coded utterance (reference) to the packet-coded version (no lost packets) for low and high background noise. The five noise conditions were included to verify that the subjects were able to discern differences caused by a commonly-experienced impairment. These conditions compared the low-noise PCM utterance (45.7 dB S/N_b) with the same utterance after adding five different levels of computer noise, resulting in S/N_b values of 40.0, 37.6, 34.9, 32.1, and 29.2 dB. The 32.1 dB condition corresponds to the "high" background noise PCM reference condition. The 23 packet loss conditions tested different nominal packet loss rates for the four delivery priorities. In addition, priority W (background noise packets) was tested with low and high levels

[2] Note that during the active speech, the PCM quantization noise (approximate 38 dB signal-to-distortion ratio) would dominate under these recording conditions.

TABLE II
NUMBER OF PACKETS IN EACH DELIVERY PRIORITY FOR REFERENCE UTTERANCES

Speaker	Noise	Delivery Priority				Total
		W	X	Y	Z	
female	low	74	175	98	91	438
female	high	139	173	41	85	438
male	low	141	146	76	85	448
male	high	194	141	39	74	448

TABLE III
PACKET LOSS PROCESSING CONDITIONS

Delivery Priority	Background Noise Level	Nominal Packet Loss Percentages (Within Delivery Priority)
W	Low	30 50 70 90
	High	30 50 70 90
X	Low	5 10 20 30 40
Y	Low	5 10 20 30 40
Z	Low	3 5 10 20 30

of background noise. These conditions are summarized in Table III.

3) Results and Analysis: The four listening conditions per processing condition combined with the 24 subjects resulted in 96 votes per processing condition. For each processing condition, we calculated the percentage of subjects who preferred the "processed" utterance to the reference utterance. We denote this percentage by P. Thus, we expect P to be close to 50 percent when the processed utterance is indistinguishable from the reference and approach 0 percent for highly distinguishable pairs. The null hypothesis that utterances A and B are indistinguishable is rejected at a 0.01 significant level when the value of P is lower than 38 percent, which represents the threshold of distinguishability.

For the PCM versus packet-coded conditions, the value of P was 38.5 percent for the low background noise condition and 47.9 percent for the high-background noise condition. Thus, in both cases, the packet-coded version of the utterance can be considered indistinguishable from the PCM version.

Noise condition results are plotted in Fig. 5 along with the distinguishability threshold. It is seen that the subjects were able to discern small levels of added noise, with the threshold of distinguishability lying above 40 dB in terms of S/N_b.

Results for the packet loss conditions are plotted in Fig. 6(a)-(e). The plots are labeled according to the nominal class of packet placed in each delivery priority, i.e., priority W is called "background noise packets," priority X is called "voiced packets," etc. Packet loss percentages for the plotted points differ from the nominal percentages of Table III only due to pseudorandom number generator effects.

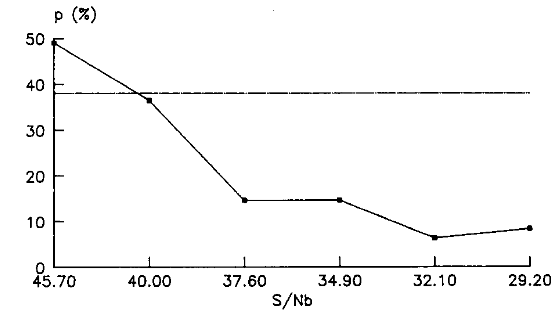


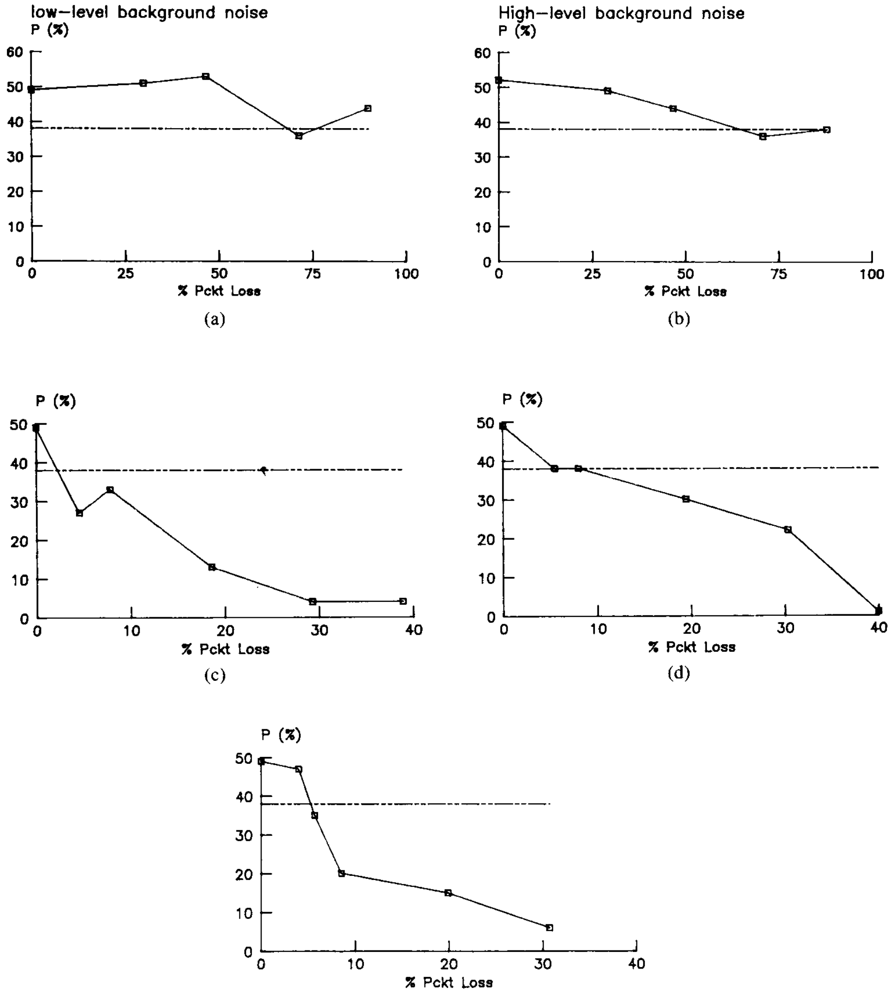
Fig. 5. Percentage of listeners (P) who chose the signal with added noise over the original signal versus signal to background noise ratio (S/N_b). The horizontal line represents the threshold of distinguishability.

Note that the value of P for the zero packet loss reference condition (in which case utterances A and B are identical) is indeed close to 50 percent. For lost priority W (background noise) packets [Fig. 6(a) and (b)], the value of P decays very slowly and the loss is indistinguishable for at least 47 percent packet loss. Even for the remaining packet loss rates, the values of P are not more than 0.02 below the distinguishability threshold. For the low-noise case, the background noise level is so low that the results are more indicative of the clipping distortion introduced by the classifier than of the packet regeneration technique. For the high-noise case, the noise level was clearly audible to the subjects (see the 32.1 dB point of Fig. 5) so the results are more indicative of the performance of the packet regeneration technique.

For priority X (voiced) packets [Fig. 6(c)], we must consider the packet loss to be distinguishable even for the lowest packet loss rate of 5 percent. Close examination of the waveforms in the vicinity of the missing packets indicates that a significant part of the problem is interaction between the particular coding and regeneration schemes used, in particular the amount of memory associated with the decoding process. Since pitch prediction was used, a regenerated packet could continue to introduce distortion in the decoding of subsequent, correctly received packets for quite some time (over 100 ms in one observed case). This is clearly to be avoided in future work.

For priority Y (fricative) packets [Fig. 6(d)], we consider the loss to be indistinguishable up to a packet loss rate of 8 percent.[3] As expected, the performance for priority Z ("other speech") packets [Fig. 6(e)] is more sensitive to packet loss, with the onset of distinguishability occurring near 4 percent packet loss.

[3] The value of P is actually below the distinguishability threshold by 0.005, but this is sufficiently close to allow us to accept the null hypothesis.



(a)

(b)

(c)

(d)

(e)

Fig. 6. Percentage of listeners (P) who chose the signal with lost and regenerated packets over the original signal versus the packet loss rate within a delivery priority. The horizontal line represents the threshold of distinguishability. (a) Background noise packets. (b) Background noise packets. (c) Voiced packets. (d) Fricative packets. (e) "Other" packets.

IV. Conclusions and Future Work

In the first part of this paper, we developed a system model for overload control in integrated packet networks incorporating the key concept of discarding speech packets according to their importance to the reconstruction of high-quality speech at the receiver. The model was motivated by the need for overload control in integrated networks, the desire to minimize the per-packet processing required at network nodes to avoid processing bottlenecks in future IPN's, and the desire to exploit the flexibility provided by the structure of speech signals to accomplish the overload control. We discovered that, compared to previous approaches, the priority packet discarding approach results in several advantages, some of which are serendipitous in the sense that they are only indirectly related to overload control but further our goal of reducing per-packet processing.

The second part of the paper described an experiment designed to test the validity of the model from the speech quality and signal processing points of view. In particular, we verified that different classes of speech indeed can be made to have different subjective tolerances to packet losses. On the basis of this experiment, we would assign background noise packets the lowest delivery priority (as expected), followed by fricative speech packets and finally "other speech" packets (we discuss voiced speech packet results below). Note that there are two mechanisms contributing to this result.

The first mechanism is the mere separation of the speech into the different classes, reflecting the inherent importance of the classes to subjective quality. For example, it is quite probable that the difference in packet loss sensitivity between the background noise and "other speech" classes would still have been significant if lost packets for both classes had been regenerated using a common method (e.g., by repeating the last packet). This effect is enhanced, however, by the second mechanism, which is to match the regeneration techniques to the characteristics of the particular class. This experiment has demonstrated the effect of the combination of the two mechanisms; other tests would be required to identify their individual contributions.

The experiment also demonstrated the important result that significant percentages of speech packets can be lost and regenerated with no *distinguishable* effect on perceived quality. We reiterate that this is a considerably more sensitive criterion than the acceptability criterion used in previous studies and emphasize that even very distinguishable conditions in our tests (corresponding to low values of P) may nonetheless provide quite acceptable quality in many applications.

The results for voiced speech were disappointing but nonetheless instructive. We knew that there would be some interaction between the regeneration of lost packets and the memory of the decoding process for received packets, but we were unprepared for the magnitude of the subjective effect. We are in the process of identifying appropriate coding mechanisms which drastically reduce or eliminate the memory associated with the decoding process, and we expect such decoding to significantly reduce the sensitivity of voiced speech to lost packets.

A major aspect of the proposed system model has been verified in principle. We will continue work to refine the coding and regeneration algorithms, improve the classification algorithm, and further characterize the subjective quality of the resulting system. We have also initiated research to study the effectiveness of priority discarding of speech packets as an overload control mechanism. This will involve the development of particular discarding algorithms and associated overload measures, followed by analysis of their effect on network performance. Looking further into the future, we plan to extend the basic concepts of the model to other structured signals such as image and video.

Appendix A
Classifier Feature Definitions

We begin the precise definition of feature $F1$, the signal level feature, by defining

$$x(i, j), \quad i = 1, 2, \cdots S \quad j = 0, 1, \cdots H$$

as the ith linear sample value in the jth segment of S samples preceding the current segment ($j = 0$ thus corresponds to the current segment). In the current experiment, $S = 64$ and $H = 640$ representing a history of approximately 5 s. We then define

$$P(j) = \max \left\{ \left| x(i, j) \right| : i = 1, 2, \cdots S \right\} \quad \text{(A1)}$$

so that $P(j)$ is the peak absolute sample value in the jth segment preceding the current segment. Define

$$P_{\min}(0) = \min \left\{ P(j) : j = 1, 2, \cdots H \right\} \quad \text{(A2)}$$

that is, the minimum value of $P(j)$ over the H segments preceding the current segment. $F1$ can then be expressed as

$$F1 = F1(0) = P(0)/P_{\min}(0). \quad \text{(A3)}$$

The segment index is explicitly shown since the value of $F1$ for the previous segment, $F1(1)$, is used to define an auxiliary feature which indicates how these peak ratios are changing:

$$F1' = \frac{\left| F1(0) - F1(1) \right|}{F1(1)}. \quad \text{(A4)}$$

To define $F2$, the first autocorrelation coefficient, first define

$$x(i) = \begin{cases} x(i, 0) & \text{for } i = 1, 2, \cdots S \\ x(S, 1) & \text{for } i = 0. \end{cases} \quad \text{(A5)}$$

Then the second feature is

$$F2 = \frac{\sum_{i=1}^{S} x(i) \cdot x(i-1)}{\sqrt{\left(\sum_{j=1}^{S} x^2(j) \right) \left(\sum_{k=0}^{S-1} x^2(k) \right)}}. \quad \text{(A6)}$$

To define the new normalized entropy feature $F3$, first define

$$m(i) \quad i = 1, 2, \cdots S$$

as the signed magnitude representation of the $\mu = 255$ PCM value of the ith sample in the current segment, that is, an integer in the range $[-127, 127]$. For the current segment, define the discrete probability distribution $p(k)$ as

$$p(k) = n(k)/S \quad k = -127, \cdots 127 \quad \text{(A7)}$$

where

$$n(k) = \sum_{i=1}^{S} \delta(m(i) - k) \quad \text{(A8)}$$

and

$$\delta(z) = \begin{cases} 1 & \text{for } z = 0 \\ 0 & \text{for } z \neq 0. \end{cases}$$

Now use the standard definition of entropy:

$$H = \sum_{k} - p(k) \cdot \log_2 (p(k)) \quad \text{(A9)}$$

where

$$p(k) \cdot \log_2 (p(k)) = 0 \quad \text{for } p(k) = 0.$$

As defined, H is a measure of the uniformity of the probability distribution over its *maximum possible range.* We normalize this to get a measure of uniformity over the *actual* range in the segment, as follows. Define

$$K_{\max} = \max \{k : p(k) \neq 0, k = -127, \cdots 127\} \tag{A10}$$

$$K_{\min} = \min \{k : p(k) \neq 0, k = -127, \cdots 127\} \tag{A11}$$

$$K_{\text{range}} = K_{\max} - K_{\min} + 1 \tag{A12}$$

$$H_{\max} = \log_2(K_{\text{range}}). \tag{A13}$$

That is, $H_{\max}$ is the maximum possible entropy for a discrete distribution with K_{range} possible values. Thus, we always have $H \leq H_{\max}$, and we can define normalized entropy, the third feature, as

$$F3 = H_{\text{norm}} = H/H_{\max}. \tag{A14}$$

Thus, $F3$ is a dimensionless quantity which tends to unity for any distribution which is nearly uniform over its nonzero range and tends to zero for highly nonuniform distributions.

For the definition of the periodicity feature $F4$, we assume the existence of an AMDF function for a suitably filtered version of the signal, as described in the discussion of voiced speech coding. We will denote this function

$$D(m) \qquad m = 0, 1, \cdots, M$$

where M is the size of the period search window, in samples. The fourth classifier feature is a type of max-to-min ratio involving $D(m)$, larger values of which indicate greater confidence that the signal is indeed periodic. We first establish an interval over which the minimum of the AMDF is likely to occur at the pitch period, as follows. Define M_{start} as the smallest value of m (within a reasonable range) which represents a relative maximum or minimum of the AMDF:

$$\begin{aligned} M_{\text{start}} = \min \{m : &\operatorname{sgn}[D(m) - D(m-1)] \\ &\cdot \operatorname{sgn}[D(m+1) - D(m)] = -1, \\ &m = 24, \cdots M - 1\}. \end{aligned} \tag{A15}$$

Then we define the minimum value of the AMDF as

$$D_{\min} = \min \{D(m) : m = M_{\text{start}}, \cdots, M\} \tag{A16}$$

and the smallest corresponding index value we call $m(D_{\min})$. Now we search for the first relative maximum previous to $m(D_{\min})$, which we call $D_{\max}$. Finally, the fourth classifier feature is

$$F4 = \frac{D_{\max}}{D_{\min}}. \tag{A17}$$

Appendix B
Classifier Tuning and Performance

The logic and parameter values for the classifier were refined using manual classification of two test sentences, each spoken by a male and a female, based on examination of the signal waveform and knowledge of phonemic content. The "tuning" sentences were "We were away a year ago" and "Every salt breeze comes from the sea."

During this tuning process, we attempted to bias the automatic classifier to favor the "other speech" classification, since the coding and regeneration techniques for background noise, voiced speech, and fricative speech all depend on specific characteristics of the class for proper operation, but the techniques for "other speech" are entirely general.

The assessment of classifier performance is difficult since it must be based on a manual classification, which is subjective, tedious, and fallible. We make no claims of the optimality of the automatic classifier, but offer the following discussion of its performance. A comparison of the manual and final-version automatic classification yielded a discrepancy rate of 11 percent, tempered by the following general observations.

- In view of the biasing discussed above, it is not surprising that in half of the final classification discrepancies, the automatic classifier placed a segment in the "other speech" class which had been manually placed in a difference class.
- Nearly one-fifth of the discrepancies involved manually classifying large blocks of the waveform as fricative or "other speech," when closer inspection (and automatic classification) revealed that individual segments within the block could just as easily have been classified as background noise.
- Most of the discrepancies involving a manually-classified voiced segment occurred at the beginning or end of a segment of voiced speech, where the signal properties are highly nonstationary.
- The number of discrepancies decreased significantly when noise was added to the utterance, indicating that many of the discrepancies involved subtleties of speech which were masked by the noise.

Further work is needed to refine this classification technique, but this performance is sufficient for the present experiment.

Acknowledgment

We are grateful at AT&T for the donation of workstations and voice processing hardware used in the experiment. We also thank the students who patiently participated in the subjective study and D. Reznik for his help in preparing the test material and organizing the results.

References

[1] J. S. Turner, "Design of an integrated services packet network," *IEEE J. Select. Areas Commun.*, vol. SAC-4, pp. 1373–1380, Nov. 1986.

[2] R. Handel, "Evolution of ISDN towards broadband ISDN," *IEEE Network*, vol. 3, pp. 7–13, Jan. 1989.

[3] R. W. Muise, T. J. Schoenfeld, and G. H. Zimmerman, "Experiments with wideband packet technology," in *Proc. 1986 Int. Zurich Seminar Digital Commun.*, Mar. 1986, paper D4.

[4] N. Yin, S. Li, and T. Stern, "Congestion control for packet voice by selective packet discarding," in *Proc. 1987 IEEE Global Commun. Conf.*, Dec. 1987, paper 45.3.

[5] H. Heffes and D. M. Lucantoni, "A Markov-modulated characterization of packetized voice and data traffic and related statistical per-

formance," *IEEE J. Select. Areas Commun.*, vol. SAC-4, pp. 856-869, Sept. 1986.

[6] K. Sriram and W. Whitt, "Characterizing superposition arrival processes in packet multiplexers for voice and data," *IEEE J. Select. Areas Commun.*, vol. SAC-4, pp. 833-846, Sept. 1986.

[7] V. S. Frost, E. M. Friedman, and G. J. Minden, "Multirate voice coding for load control on CSMA/CD local computer networks," *Comput. Networks ISDN Syst.*, vol. 11, pp. 99-110, 1986.

[8] D. J. Goodman, "Embedded DPCM for variable bit rate transmission," *IEEE Trans. Commun.*, vol. COM-28, pp. 1040-1046, July 1980.

[9] T. Bially, B. Gold, and S. Seneff, "A technique for adaptive flow control in integrated packet networks," *IEEE Trans. Commun.*, vol. COM-28, pp. 325-333, Mar. 1980.

[10] N. S. Jayant and S. W. Christensen, "Effects of packet losses in waveform coded speech and improvements due to an odd-even sample-interpolation procedure," *IEEE Trans. Commun.*, vol. COM-29, pp. 101-109, Feb. 1981.

[11] B. Gold, "Digital speech networks," *Proc. IEEE*, vol. 65, pp. 1636-1658, Dec. 1977.

[12] D. Cohen, "On packet speech communication," in *Proc. Fifth Int. Conf. Comput. Commun.*, Oct. 1980, pp. 271-274.

[13] N. Matsuo, M. Yuito, and Y. Tokunga, "Packet interleaving for reducing speech quality degradation in packet voice communications," in *Proc. 1987 IEEE Global Commun. Conf.*, Dec. 1987, paper 45.4.1.

[14] D. J. Goodman, "Waveform substitution techniques for recovering missing speech segments in packet voice communications," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 1440-1448, Dec. 1986.

[15] W. A. Montgomery, "Techniques for packet voice synchronization," *IEEE J. Select. Areas Commun.*, vol. SAC-1, pp. 1022-1028, Dec. 1983.

[16] O. J. Wasem *et al.*, "The effect of waveform substitution on the quality of PCM packet communications," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 342-347, Mar. 1988.

[17] D. Minoli, "Optimal packet length for packet voice communication," *IEEE Trans. Commun.*, vol. COM-27, pp. 607-611, Mar. 1979.

[18] J. G. Josenhans *et al.*, "Speech processing application standards," *AT&T Tech. J.*, vol. 65, pp. 23-33, Dec. 1986.

[19] N. S. Jayant, "Adaptive quantization with a one-word memory," *Bell Syst. Tech. J.*, vol. 52, pp. 1119-1144, Sept. 1973.

[20] M. J. Ross *et al.*, "Average magnitude difference function pitch extractor," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-22, pp. 353-361, Oct. 1974.

[21] L. R. Rabiner, "On the use of autocorrelation analysis for pitch detection," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-25, pp. 24-33, Feb. 1977.

[22] L. A. DaSilva, "A class-oriented voice communication system for packet-switched networks: Description and performance study," Masters thesis, Univ. of Kansas, Aug. 1988.

[23] "IEEE recommended practice for speech quality measurements," *IEEE Trans. Audio Electroacoust.*, vol. AU-17, pp. 225-246, Sept. 1969.

David W. Petr (S'89) received the B.S.E.E. degree from Southern Methodist University, Dallas, TX, in 1976 and the M.S.E.E. degree from Stanford University, Stanford, CA, in 1978.

From 1977 to 1986 he was employed by AT&T Bell Laboratories, working in the areas of digital transmission systems, packet communication networks and protocols, and speech coding. He was awarded three patents related to this work. During the 1983-1984 school year, he represented AT&T as a Visiting Professor at Tennessee State University, Nashville. Since 1987, he has been pursuing the Ph.D. degree in electrical and computer engineering at the University of Kansas, Lawrence, for which he received the 1987 IEEE Frank A. Cowan Scholarship for graduate studies in communications. He has also been a Research Assistant in the Telecommunications and Information Sciences Laboratory at the University of Kansas. His current research interests are focused on speech processing and overload control in integrated packet communication networks.

Luiz A. DaSilva, Jr. was born May 17, 1964 in Rio de Janeiro, Brazil. He studied at the University of Rio de Janeiro before coming to the University of Kansas, Lawrence, on a Fulbright Scholarship in 1983. He received the B.S. and M.S. degrees in electrical and computer engineering from the University of Kansas in 1986 and 1988, respectively.

During 1987 and 1988 he worked in the Telecommunications and Information Sciences Laboratory at the University of Kansas Center for Research Inc., studying new techniques for the replacement of lost speech packets. He is currently employed by IBM Brazil in Rio, and he maintains his interest in communication networks in general and packet speech networks in particular.

Victor S. Frost (S'75-M'82) was born in Kansas City, MO, on March 6, 1954. He received the B.S., M.S., and Ph.D. degrees from the University of Kansas, Lawrence, in 1977, 1978, and 1982, respectively.

From 1974 to 1977 he was a Research Technician; from 1977 to 1978 he was a Research Engineer engaged in radar simulation and modeling research; and from 1978 to 1983 he was a Project Engineer conducting research in radar image processing, all at the University of Kansas Center for Research, Inc., Lawrence. In 1981, he was a Visiting Scientist at the German Aerospace Research Establishment (DFVLR), Oberpfaffenhofen, West Germany. In 1982, he joined the faculty of the University of Kansas, where he is now an Associate Professor of Electrical and Computer Engineering and Director of the Telecommunications and Information Sciences Laboratory. His current research interest is in the area of integrated communication systems and network simulation.

Dr. Frost has received an Air Force Summer Faculty Fellowship, a Ralph R. Teetor Educational Award from the Society of Automotive Engineers, a Miller Professional Development Award, and a Presidential Young Investigator Award from the National Science Foundation. He is a member of Eta Kappa Nu and Tau Beta Pi.