AN 8086-BASED PACKET-VOICE SIMULATOR FOR LOCAL AREA NETWORKS

by

PHILIP MUI

A thesis
presented to the University of Ottawa
in partial fulfillment of the
requirements for the degree of
MASTER OF APPLIED SCIENCE
in
the DEPARTMENT of ELECTRICAL ENGINEERING

OTTAWA, Ontario, 1983

The University of Ottawa requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

ABSTRACT

Packet transmission of voice on a broadcast bus type local area network is studied. Two multi-access protocols are compared, CSMA/CD and HYMAP. A real-time simulator based on an 8086 CPU is designed to study the speech quality subjectively.

Packet delay distribution and packet discard probability, which are the inputs to the simulator, are obtained from a separate simulation program under various network environments. The independent variables are the number of users and the maximum allowable network delay.

A packetization-freezed protocol is proposed to eliminate the successive collisions due to possible synchronization of packet generation among stations. The variance of the network delay is bounded by discarding packets which have not been transmitted within a certain amount of time. Speech-silence is not transmitted and the packet sequence number is used for silence recovery. Smooth speech output can be obtained by introducing additional buffer-delay at the receiver.

ACKNOWLEDGEMENTS

I wish to express my sincere appreciation to my supervisor, Dr. Nicolas Georganas, for his guidance, encouragement and financial assistance.

I would also like to express my thanks to my close friends Esper Bitar, Adrian Conway, Pavlos Kariotellis, Jacob Teo and Kevin Wong; also to Jim Kozlowski for providing me with personal support and to Miguel Rios for his kindness in lending me his simulation program.

I would like to acknowledge the technical assistance given by Mohammed Master without which my work would have been considerably delayed.

Last but not least, I want to thank all the staff of the Department of Electrical Engineering, especially René Le Hénaff for his excellent coordinating in administrative work.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

A/D - analog-to-digital conversion

BD - buffer-delay

BM - buffer mode flag

BPQ - buffer-pool queue

CR - computation routine

CSMA/CD - carrier sense multiple access with collision detection

CVSD - continuously variable slope delta modulation

D/A - digital-to-analog conversion

DBA - different back-off algorithms

DELOVER - delay over flag

DPCM - differential pulse code modulation

EOT - end of transmission

FIFO - first-in-first-out

GPAS - global priority assignment scheme

HYMAP - hybrid multiple access protocol

IDVS - integrated data and voice station

IR - initialization routine

ISR - interrupt service routine

LAN	- local area network
LPAS	- local priority assignment scheme
MAND	- maximum allowable network delay
P/S	- parallel-to-serial conversion
PCM	- pulse code modulation
PDD	- packet discard decision
PHR	- packet handling routine
PKHAND	- packet handling requested flag
PVP	- packet-voice processor
PVR	- packet-voice receiver
PVS	- packet-voice station
RM	- reservation message
RND	- random network delay
RRS	- receiver require service flag
RXQ	- receive queue
SDK-86	- 8086 CPU based system design kit
SILMODE	- silence mode flag
ST	- silence threshold
TRS	- transmitter require service flag
TXQ	- transmit queue
VT	- voice threshold

Chapter I

INTRODUCTION

The circuit switched public telephone network evolved over many years with the goal of providing voice communications. In circuit switching, a dedicated path between the users exists once a call has been setup and will exist until the call is finished. The utilization of the path is relatively high because when people are talking, it is unusual for gaps in the conversation to last a few minutes. However, when computers are communicating, circuit switching has a relatively low utilization because of the high peak-to-average data traffic. In other words, human-to-human traffic needs continuous use of a low-bandwidth channel, whereas computer-to-computer traffic needs intermittent use of a high-bandwidth channel. Packet switching and computer networks emerged as an alternative to handle computer traffic.

The main goal of a computer network is to interconnect scattered computer facilities as a whole so that all programs, data and other resources are available to anyone on the network without regard to the physical location of the resource and the user. An example of this is the ARPA network which interconnects between many large computer centers located throughout the Continental United States with connections to Hawaii and Europe via satellite links.

In contrast to the long-haul network which spans a large area is the local area network, LAN, which spans a radius of a few Km. An example of this is the Ethernet [1][32], which is based on a broadcast bus (a coaxial cable) type architecture and CSMA/CD multi-access protocol. The network runs at 10 Mb/s and can support 1024 users simultaneously within a distance of 2.5 Km. The introduction of LAN is mainly due to the superior price/performance ratio of small computers over large ones. Mainframes are roughly ten times faster than the largest single chip microprocessors, but they cost a thousand time more [1]. In many offices or institutions, a collection of micro-computers may be used to outperform the large mainframes at lower cost. A wide range of applications can be supported in a LAN such as a data base retrieval system, file transfer among computers, shared use of expensive peripherals and distribution of electronic mail [2].

However, voice communication remains an important mode of communication in an office environment in the form of two-way communication, one-way broadcast or non-real-time applications such as dictation and voice message system. Due to the advances in digital logic circuitry, it is now inexpensive to perform analog-to-digital conversion of voice and to perform sophisticated digital switching. Moreover, digital transmission has more tolerance to noise than analog transmission. Also, compression as well as encryption can be more easily implemented digitally. These advances result in a trend toward integrating voice communication onto the digital data networks [3][4][5][6]. However, before studying the effects of the integration, one has to study the transmission of voice alone on a data network. The purpose of this thesis is to perform such a study.

In the past few years, some research has been done on the packet transmission of voice over two different types of networks, namely large scale and local area networks. With respect to the former, Lincoln Laboratories carried out the first experiment with packetized voice which was conducted over the ARPANET [7]. Minole [8] derived a store-and-forward link model which was used to obtain the delay distribution and packet loss rate. Embedded speech coding was proposed by [9] as an adaptive packetized voice flow control scheme. The real-time simulation carried out at the City College of New York [10] proposed some system design parameters. Regarding packet-voice over local computer networks, most of the research work has so far concentrated on broadcast bus type network architectures. Some corporations [2][11] studied the feasibility of carrying packetized voice on Ethernet. Their results were optimistic. Computer simulations were done by [12][14] based on the CSMA/CD protocol to study the packet loss rate in terms of the packet delay and the number of users.

In this thesis, a real-time simulator will be designed to study the voice transmission on a broadcast bus type local area network based on two multi-access protocols, namely, Carrier Sense Multiple-Access with Collision Detection (CSMA/CD) [16] and Hybrid Multiple-Access (HYMAP) [15]. System design parameters of a packet-voice station are given as a result of the simulation. Due to the low cost and flexibility of the microprocessor, an 8086 microprocessor system is used as the base of the simulator.

In the CSMA/CD protocol, every station first listens to the broadcast channel before packet transmission takes place (i.e. carrier sense). If the channel is sensed idle, the station will transmit its packet. If the channel is sensed busy, someone is transmitting, then the station will defer its transmission until the channel is sensed idle. During transmission, the station monitors its own transmission and checks if there is any interference from other stations (i.e. collision detect). Their transmissions are aborted if there are two or more stations transmitting at the same time. A jamming signal is sent by the collided stations to make sure that all the involved stations know of the collision. Retransmission takes place after a random period of time based on the back-off algorithm. The one used in this study is the binary-exponential back-off algorithm.

The HYMAP [15] protocol utilizes the best features of both CSMA/CD and a collision-free protocol. When there is no collision, all stations transmit their packets based on the CSMA/CD protocol. As soon as one collision occurs, all stations will switch to a collision-free protocol instead of entering into the back-off state. The collision-free protocol synchronizes all stations in such a way that each station will transmit its packet only after the previous station has already transmitted. All stations have to be in sequence and the best way to arrange them is by numbering them, say in ascending order, from one end of the channel to the other end. When a collision occurs, the leftmost or the rightmost station on the channel starts the synchronization sequence. It transmits a negative bit sequence if it has no packet to transmit. If it has a packet to transmit, a positive bit sequence will be transmitted and the packet transmission follows right after. The next station will repeat the same procedure when it receives either a negative bit sequence or senses the end of a packet transmission from the last station. All stations follow this procedure until all packets have been transmitted at which time the protocol returns to CSMA/CD.

The reasons of studying packetized voice on a broadcast bus type network are [12]:

1. Voice is an important mode of communication in an office environment. Its applications include two-way conversations, dictation and voice message systems etc.
2. The possibility of integrating data and voice on the same network.
3. The broadcast type LAN is considered because of its simplicity, in terms of topology and device interconnection, and its flexibility in satisfying the growth and variability requirements in the environment.
4. The low bandwidth of digitized speech allows a reasonable number of users (>100) to be supported in current LANs.

The organization of this thesis is as follows:

Chapter 2 outlines the digitization techniques. The characteristics of conversational speech as well as various methods on speech quality measurement will be given.

Chapter 3 describes the design issues of a packet-voice station as well as their impacts on system performance.

Chapter 4 describes the packet-voice simulator in two sections, hardware and software. The simulation experiments and results will be given at the end of this chapter.

Chapter 5 describes the packet handling problem of data and voice. A block diagram of an integrated data and voice station will also be given.

Chapter II

VOICE DIGITIZATION TECHNIQUES

Voice is an analog phenomenon. A conversion has to be done in order to carry analog voice through a digital transmission system. This requires the use of an analog-to-digital converter (ADC) at the input and a digital-to-analog converter (DAC) at the output. They are usually implemented in a device which is called a codec (encoder and decoder). Speech is a highly compressible source that can be coded for transmission at rates ranging from 64K to below 2.4K bit/sec [17].

In general, voice digitization techniques can be categorized into two classes, waveform reconstruction and analysis-synthesis. The first class encodes the analog waveform as faithfully as possible. It generates an output waveform which looks like the original input signal. This class is representative of the general problem of analog-to-digital and digital-to-analog conversions and is signal independent. Hence, it can code equally well a variety of signals such as speech, music and tones. The most commonly used method in the higher ranges of transmission rate is the Pulse Code Modulation (PCM) in which input speech is sampled in the order of 8 KHz and coded using 6-10 bits/sample yielding 48-80 Kb/s digital signal. In addition to this standard method, several low bit-rate algorithms have been developed which take advantage of the correlation properties of the speech waveform. The basic premise is that fewer bits are needed to code the derivative of the speech than to code the speech directly. Therefore, the difference between adjacent samples are coded for transmission instead of the individual sample as in PCM.

The second class is called analysis-synthesis which is used to produce a very low digital speech data rate for narrowband transmission systems or digital storage devices with limited capacity. The input speech is analyzed at fixed-duration intervals, frames, which are coded by a small number of bits for transmission. Some approaches for analyzing speech are the recognitions of speech phonemes, pitch periods and formants. A synthesizer can then reconstruct the speech by recreating the speech phonemes, pitch periods and formants based on the received coded data. Although this technique is capable of producing intelligible speech, generally the naturalness is lost.

A spectrum of speech coding transmission rates [13] currently of interest is shown in Figure 1 . The figure highlights non-speech specific waveform codecs (waveform reconstruction) that need relatively higher transmission rates and speech-specific vocoders (analysis-synthesis) for digitization at relatively lower bit rates. The figure also indicates the quality of speech reproduction that can presently be attained at a prescribed bit rate. The quality characterizations are denoted as commentary, toll, communications and synthetic [13].

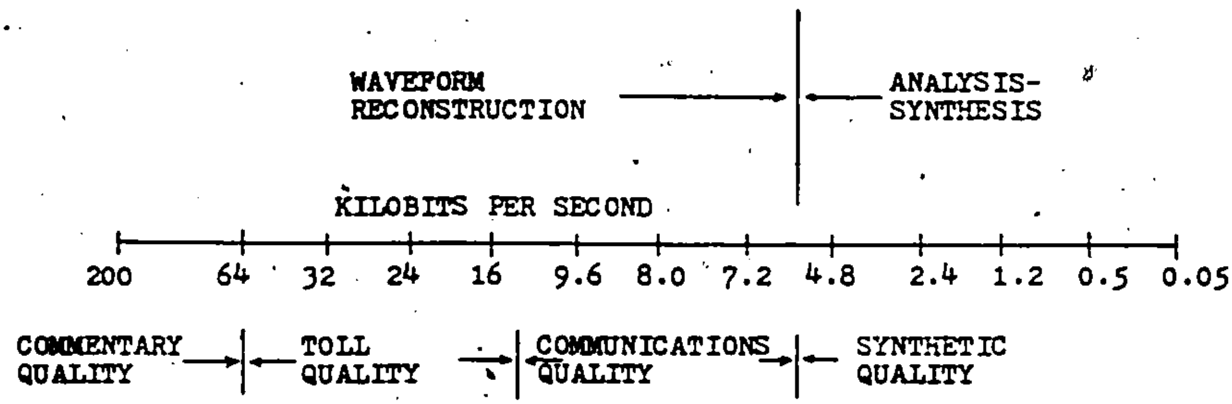


Figure 1: Spectrum of Speech Coding Transmission Rates

Vocoders in the analysis-synthesis range provide a synthetic quality where the signal usually has lost substantial naturalness. Typically the signal sounds automaton like. Talker recognition is substantially degraded and vocoder performance is talker dependent.

Waveform codecs that provide communications quality speech can be realized with rates below 16 Kb/s. The signal is highly intelligible but has noticeable quality reduction, some detectable distortion and perhaps lessened talker recognition.

Telephone toll quality digital codecs can be realized for speech signals at coding rate between 16 Kb/s and 64 Kb/s. The term toll quality is typically used to imply quality comparable to that of an analog speech signal having approximately the following properties:

- -frequency range : 200 to 3200 Hz
- -signal-to-noise ratio : > 30 dB
- -harmonic distortion : < 2 to 3 %

At the upper end of the spectrum, above 64 Kb/s, it is possible to obtain the signal-to-noise ratio and harmonic distortion characteristic of toll quality with input signal bandwidth significantly wider than normal telephone (e.g. 0 to 7 KHz or better). This grade of quality is referred to as commentary quality. It is appropriate for digitizing some varieties of radio broadcast material.

Table 1 illustrates various digitization techniques, their cost and digitization rate [17]. It must be pointed out that the indicated voice digitization rate associated with each technique in the table represents the typical bit rate required for acceptable voice quality. The cost of the vocoders is being reduced continuously by the use of custom ICs. As a general rule, as digitization rate decreases, the complexity of the encoding/decoding hardware goes up as well as the cost. The voice quality is reduced and the harmful effects of noise or distorted inputs on speech quality are magnified.

TABLE 1

Various Digitization Techniques and Their Characteristics

DIGITIZATION TECHNIQUE	TRANSMISSION RATE (Kb/s)	COST CHARACTERIZATION
LINEAR PULSE CODE MODULATION (PCM)	90-100	RELATIVELY INEXPENSIVE DEVICE
LOG PCM	48-64	
DIFFERENTIAL PCM	32-48	
CONTINUOUSLY VARIABLE SLOPE DELTA MODULATION (CVSD)	16-32	
SUB-BAND CODER (SBC)	7.2-24	CURRENTLY EXPENSIVE
LINEAR PREDICTIVE VOCODER	2.4-9.6	
CHANNEL VOCODER	2.4-4.8	
CEPSTRUM VOCODER	2.4-4.8	
FORMANT VOCODER	0.6-2.4	

2.1 CHARACTERISTICS OF CONVERSATIONAL SPEECH

During the conversation of a pair of talkers, a simple protocol of allowing one of them to speak at any given time is adopted by all people. Confusion and unnecessary delay will result if the talkers do not follow this protocol. Generally, there are ten events [18] which characterize the conversation of a pair of talkers, A and B . They are:

1. Talkspurt - a continuous segment of speech.
2. Pause - a continuous segment of silence.
3. Double Talk - a time when speech is present from both A and B.
4. Mutual Silence - a time when silence is present from both A and B.
5. Alternation Silence - the period of mutual silence between the end of one speaker's talkspurt and the beginning of the other's.
6. Pause in Isolation - a pause in which the other speaker is silent throughout the pause.
7. Solitary Talkspurt - a talkspurt which occurs entirely within the other speaker's silence.
8. Interruption - the period of talkspurt in which one speaker interrupts the other.
9. Speech after Interruption - the period of talkspurt of the interrupted speaker after interruption.
10. Speech before Interruption - the period of talkspurt of the interrupted speaker before interruption.

The four events that an average talker is aware of are the talkspurt, pause, double talk and mutual silence. Brady [18] did an extensive analysis of on-off speech patterns in 16 experimental telephone conversations. The means and standard deviations of the average of the four events are shown in Table 2 .

TABLE 2

Means and Standard Deviations of the Four Events

EVENT	MEAN(SEC)	STANDARD DEVIATION(SEC)
TALKSPURT	1.197	0.444
PAUSE	1.846	0.648
DOUBLE TALK	0.251	0.055
MUTUAL SILENCE	0.466	0.088

2.2 SPEECH QUALITY MEASUREMENT

After an analog speech signal is band-limited, digitized and passed through a digital communication network, a certain degree of quality degradation will be expected at the receiver. Obviously, for good voice communication, it is necessary to understand every word, to recognize the speaker and to be able to detect from the tone some other information that is not expressed verbally, such as the speaker's mood. There have been many studies about the components of the quality of speech communication but none of them is objective enough to automate the process and build a system for measuring the absolute objective quality of a sound reproduction system. This is because it is already rather difficult to recognize isolated words in an absolute objective way because recognition depends on the number of syllables, the choice of listeners, their conditions and experience, background noise and the particular speaker. However, there are many subjective techniques for measuring speech reproduction quality. These techniques are based on subjective evaluations by a test group of people indicating preference, intelligibility of isolated words and phrases, distinguishing rhymths and the like.

IEEE recommended three methods for subjective measurement based on individual preference [19]. They are the Isopreference Method, the Relative Preference Method and the Category-Judgement Method. These methods can only be used to evaluate speech signals which are generated by one-way communication systems. Systems, such as two-way systems in which interactive communication takes place, can not be evaluated by these methods.

2.2.1 Isopreference Method

The speech test signal is compared in a forced-choice test procedure (i.e. selection of one of a choice of two) directly with a speech reference signal that is subjected to variable degrees of degradation. The signal-to-noise ratio of the reference signal is varied between an upper and lower limit whose mean is approximately isopreference to the test signal. Two signals are isopreference when the votes show an equal preference for the speech test and speech reference signals. During the test, the listeners are presented with repeated signal pairs in the order A B A B. The listeners are asked to indicate the signal in each repeated pair that they would prefer as a source of information. The significance of an isopreference signal pair is that the test signal can be represented by the signal-to-noise ratio of the reference signal.

2.2.2 Relative Reference Method

In this method, the quality of the test signal is measured on a quality continuum defined by several reference signals which represent different types of speech distortion. The selected reference signals are placed on an arbitrary scale by considering how often each reference signal is preferred to the other reference signals. The reference signals are chosen to cover the whole range of quality from "highest quality" to "lowest quality". The test signal is located on the same rating scale by considering how often it is preferred to any reference signal.

2.2.3 Category-Judgement Method

In this method, the quality of the speech test signal is described in terms of a number of categories which have an intuitive meaning to the listeners. A number of simple categories are used by listeners to describe their impressions of the quality of the speech test signal. Example of those are Excellent, Good, Fair, Poor and Unsatisfactory.

The type of speech material for test and reference signals should be the same. Two types are recommended, narrative and short homogeneously structured sentences. Some examples of the latter type are [19]:

- It's easy to tell the depth of a well.
- The juice of lemon makes fine punch.
- A large size in stocking is hard to sell.

The selection of listeners for the purpose of speech quality testing must be guided by the purposes and application of the test. In radio and telephone communication systems, a large untrained listening group is recommended. On the other hand, if the communication systems are used by a very limited and well-experienced group of individuals, such as special space communication systems, measurements may be based on the decision of a small selected group of listeners familiar with the system.

Chapter III

DESIGN ISSUES OF THE PACKET VOICE PROCESSOR

A packet-voice station (PVS) communicates with other stations by means of a shared common channel in the form of a coaxial cable. The accessibility to the channel by a PVS is governed by a multi-access protocol whose functions were described in the first chapter. To provide network flexibility, the multi-access protocol and the Packet-Voice Processor (PVP) are usually implemented on two separate pieces of hardware as shown in Figure 2 . So, if there is a change in the system, either the multi-access protocol or the PVP , one would only have to change the corresponding piece of hardware. It is the responsibility of the multi-access protocol hardware to ensure that transmission of a packet begins when the channel is idle, abort a transmission if there is a collision and schedule for the next transmission. All these processes are fully transparent to the PVP. In this thesis, the issues involved in the design of the multi-access protocols will not be covered; only the PVP will be considered.

In data communication, the primary objective is to receive all data as it was transmitted without any errors. The delay performance is secondary. For example, in

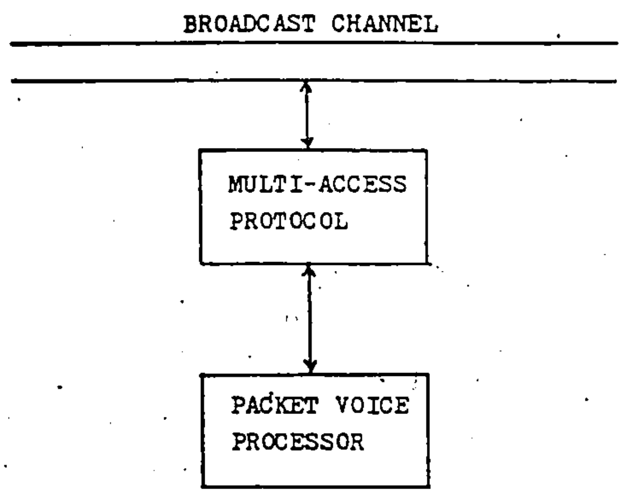


Figure 2: Block Diagram of a Packet Voice Station

terminal/computer communications, an user will at most be annoyed if the network delay is too large. The meaning of communication will not be affected as long as all information is received without error. So, in data communication, delay can be tolerated but not error. But in voice communication, person/person, one can afford to have small errors but not large delay. A small loss of speech does not affect the intelligibility of the conversation but too much delay will make the conversation difficult. In this mode of communication, there are two main important requirements [20][28]:

1. A timely continuous (smooth) voice output is necessary in order to maintain a good quality speech.

2. End-to-end delay must be held to small fractions of a second (300 ms in an echo-free system) to overcome the psychological effect of the round-trip delay.

More explanation is required to clarify these two points. If the voice output is not smooth, part of the voice output is lost intermittently, then the quality of speech will be degraded. To produce a smooth voice output, packets have to be fed to the voice decoder synchronously. The term "synchronously" refers to the delivery of the packet with the same rate at which it was generated at the transmitter. Due to the delay characteristic of the multi-access protocol, packets will suffer random delay and they will arrive at the receiver asynchronously.

The second point is better clarified by an example. Consider a network with a round-trip delay of one second. A question is raised by an user A who expects an instantaneous reply from user B. Due to the delay of the network, the reply will reach user A after one second even though user B replies instantaneously. User A will wonder why user B replies after a pause. This delay will degrade the interactive ability and will disturb the users psychologically. In a local area network environment, point 2 can be achieved easily because the maximum network delay is well below the one specified.

Before discussing the design issues of a PVP, one has to study the effects on a packet after it has passed through a broadcast channel. The net effects are packet loss and delay. The packet loss is mainly due to existence of noise on the broadcast channel and to transmission circuit errors. This type of deficiency is independent of the PVP and will not be studied further.

Packets that are transmitted by a station arrive at the destination with a variable end-to-end delay. The end-to-end delay is defined as the time between the generation of the first bit of a packet at the originating encoder and its delivery to the decoder at the receiver. It is composed of the following four elements:

1. Originating PVP Packetization Delay, P

   This delay is a function of the codec rate and packet size. It is given by:

$$P = \frac{\text{PACKET SIZE}}{\text{CODEC RATE}}$$

   Its impact on the end-to-end delay will be given later in more detail.

2. Network Delay, N

   It is composed of queueing and contention delays which are highly dependent on the multi-access protocol used. This delay increases with network load because as the number of packets in the network increases, the likelihood of collision increases as well and a packet will take a longer time to transmit successfully.

3. Packet transmission Delay, T

This delay is the time that a packet takes to transmit at the channel transmission rate. It is directly proportional to the packet size and is given by:

$$T = \frac{\text{PACKET SIZE + OVERHEAD}}{\text{CHANNEL DATA RATE}}$$

4. Propagation Delay, D

This is the amount of time that a packet takes to travel from one point to the other at the speed of light and is given by:

$$D = \frac{\text{DISTANCE TRAVELLED}}{\text{SPEED OF LIGHT}}$$

The ratio of the propagation delay to the packet transmission time is an important factor which determines the throughput and the network delay of the system. Generally speaking, the throughput decreases as this ratio increases [15].

The total end-to-end delay (TD) is the sum of all the delay elements and is given by:

$$\text{TOTAL END-TO-END DELAY, TD} = P + N + T + D$$

The delay TD should not exceed 300 ms which has been shown to be commercially acceptable in an echo-free (4-wire) system [21] [28]. When the delay reaches 800 ms, adverse psychological factors impede normal conversation [21]. For a given fixed packet size, P and T are fixed delays and the propagation delay is a constant. The sum of them, P,T and D, is the lower bound of the end-to-end delay. The remaining element, namely network delay, contributes to the fluctuation of the end-to-end delay. Under light load when there is no packet collision, the network delay is zero. In this case, all packets that are generated at the transmitter arrive at the receiver synchronously with a delay equal to the lower bound. As the network load increases, collisions occur and the network delay will no longer be zero. Due to the variation of this delay, packets will arrive at the receiver asynchronously which will degrade the quality of speech if nothing is done to improve it. This is the purpose of this chapter: to discuss the design issues of a PVP with emphasis on the receiver packet handling problem such as smooth speech delivery. Protocols will be proposed to eliminate the negative effects caused by the variance of the network delay.

A PVP consists of a transmitter and a receiver as depicted in Figure 3 . The encoder is responsible for converting the band-limited analog speech signal into digital form. When the speaker is active, silence detection is activated and packetization begins. Packets are generated periodically by the packetizer. Packets will be transmitted at the same rate as they were generated, if the network load is low with no collisions. As the load increases, contention and queueing delays increase and the packet transmission rate will no longer be uniform. Packets that are not transmitted instantaneously due to the heavy network load are put in the transmit queue. During silence periods, no packets will be generated. In order to identify the generation time of each packet, a time-stamp or sequence number is appended to the header of each packet. This information is necessary for the receiver to determine the delay experienced by each packet, to determine the possible packet loss and to recover the speech-silence during which packets were not transmitted.

The main design issues of a PVP are:

1. Select a robust digitization algorithm ,so that a temporary loss of packets does not degrade the speech quality too much.
2. Select the silence detection thresholds to minimize the packet generation rate, while maintaining all the speech intelligibility.

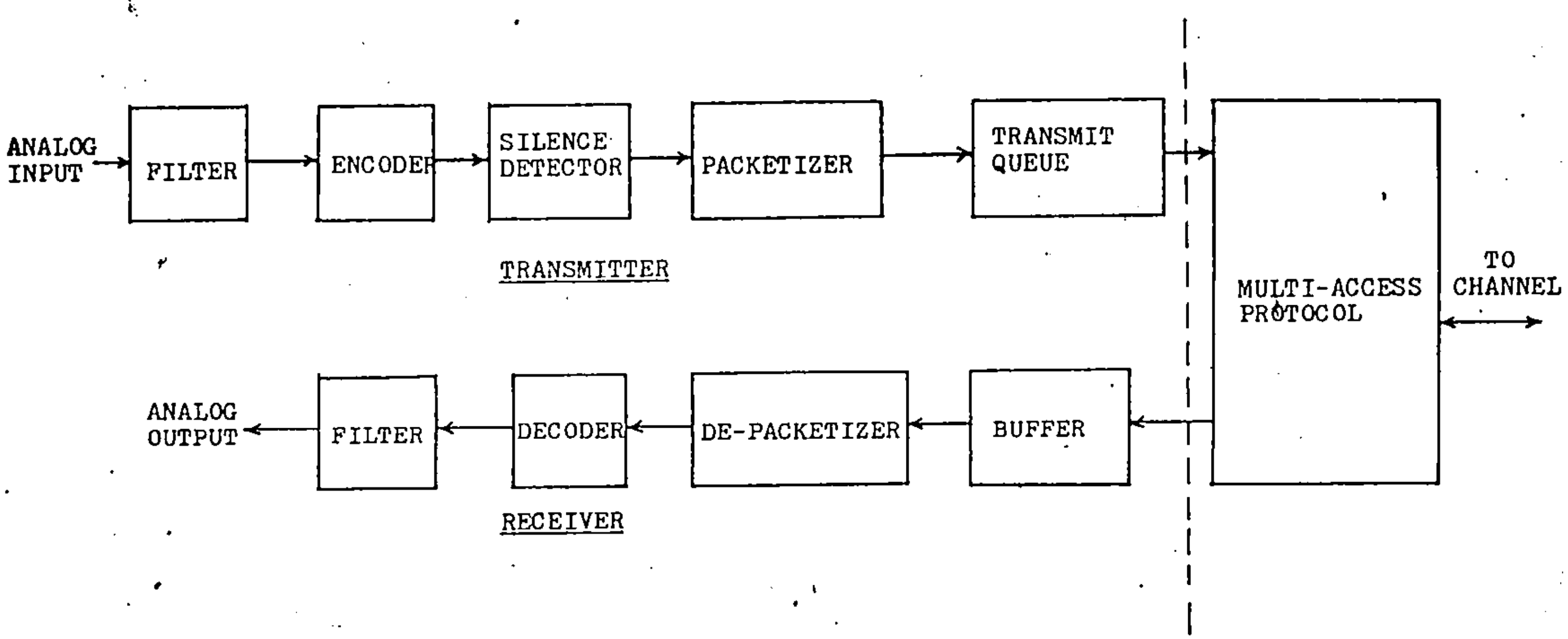


Figure 3: Block Diagram of A PVP

3. Keep the end-to-end delay, which is dominated by the packetization and the network delay, to less than 300 ms.
4. A receive-end protocol is required to overcome the variable network delay experienced by packets.
5. A protocol is required for silence recovery at the receiver.
6. A scheme of handling loss packets is required.

These issues are considered next in more detail.

3.1 ANALOG-TO-DIGITAL CONVERSION

This is the front end process of the transmitter. The analog signal has to be converted into digital form before it is handled by the rest of the system. In the previous chapter, the digitization techniques were briefly described and the transmission rates required by various techniques to achieve acceptable voice quality were given. This section summarizes the considerations to be made when choosing a particular codec for implementation. The factors that will be considered are: transmission rate, tolerance of transmission errors, signal processing requirements and implementation cost [22].

3.1.1 Transmission Rate

For low transmission rates (< 9.6 Kb/s), codecs tend to be very complex because they specifically encode voice signals. At higher transmission rates, codecs tend to encode waveforms rather than voice, and hence are less complex. Codecs using adaptive predictive techniques can achieve cummunication quality speech at 7.2 Kb/s but are more complex than those using log-PCM at 48 Kb/s.

If a dedicated channel is used for a conversation between two users, it does not matter what the transmission rate is, as long as the quality is satisfactory. However, in a local area network environment, where a channel is shared among users, lower transmission rate is a necessity in order to support many users. Consider an ideal broadcast channel, collision-free, and with M users. Let

C - Channel Bandwidth (b/s)

E - Codec Transmission Rate (b/s)

V - Packet Size (b)

O - Overhead size (b).

Packet Transmission Rate of channel = $\frac{C}{V + O}$

Packet Generation Rate of codecs = $\frac{ME}{V}$

To transmit all packets, which are generated, on the channel, it is necessary that the packet generation rate be equal to the packet transmission rate, such that:

$$\frac{ME}{V} = \frac{C}{V + O}$$

$$M = \frac{C}{E}\left(\frac{V}{V+O}\right)$$

The number of users (M) is inversely proportional to the codec transmission rate (E). Figure 4 is a plot showing their relationship. If the transmission rate is reduced by half, then the number of users can be doubled.

3.1.2 Tolerance of Transmission Errors

Of all the digitization techniques listed in the previous chapter, delta modulation is the least sensitive to random, independent channel errors. A CVSD codec provides better than 90% intelligibility with error rate as high as 10% [22]. Error probability below 0.001 is usually unnoticeable. The reason that a delta modulation codec has such a good performance on error is that a single channel error produces a decoder error magnitude of only one quantization level. Also, channel errors do not produce adjustment in the step size in a CVSD system. On the other hand, the PCM digitization technique is more sensitive to errors. When a channel error occurs on the most significant bit, a relatively large error spike is produced.

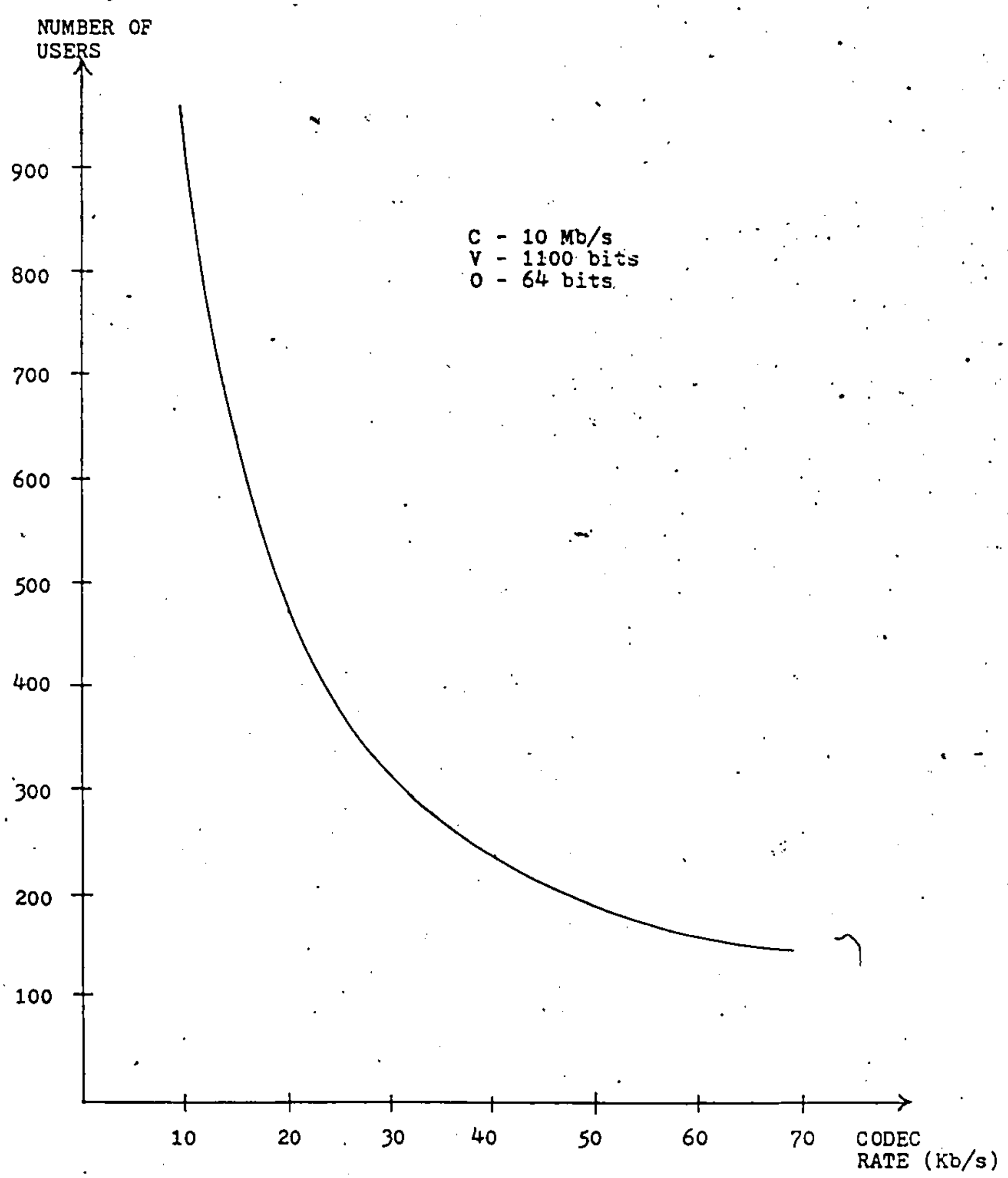


Figure 4: Number of Users Versus Transmission Rate

The channel error sensitivity of vocoders is greater than that of the PCM systems. By redundantly encoding the most critical information in the bit stream, the performance can be improved significantly.

3.1.3 Signal Processing Requirements

Many functions such as adding gain or attentuation, conferencing, sensing speech activity or converting from one algorithm to another can be done on the encoded speech by digital signal processing techniques. One of the digitization techniques that can greatly simplify the digital processing operations is the PCM. Most other encoders have been designed with no digital signal processing requirements in mind. Thus, the signal processing functions may require conversion to analog or a relatively complicated digital simulation of an analog process.

3.2 SILENCE DETECTION

In a telephone system, a channel is occupied by a pair of users throughout the conversation for regional calls. The channel utilization is very low, at most 50% on the average, due to the fact that only one user is talking at any given time and also because pauses and silence occur between words and sentences. In a situation where expensive channels are used, such as satellite channels and ocean cables, Time-Assignment Speech Interpolation (TASI) is used to increase the channel utilization. When an user is not talking, his channel is switched to other users in the system who are talking. A channel, not necessary the same one, will be returned to the user when he resumes his speech.

In a broadcast bus type system whereby a channel is shared among users, it is of great advantage if the users transmit only actual information on the channel. Brady [18][23] measured the duration of active voice periods and experimentally observed an average activity rate of about 40 percent. Based on this result, the number of users can be increased by not transmitting the silence periods.

Silence detection can be easily implemented in digital form because most of the encoding algorithms will give a stationary output when the input is silent. During the silence period, the output of PCM is all 0s and for DM, the output bit pattern is periodic. To prevent pre-mature switching, from silence mode to voice mode and vice-versa, a switching device with hysteresis threshold characteristic, shown in Figure 5, is used. Silence mode is entered when the number of voice samples is below the silence threshold (ST). It will remain in this mode until the number of voice samples is above the voice threshold (VT), at which time the silence detector switches to voice mode. More detailed description of the silence detector will be given in Section 4.2.3 .

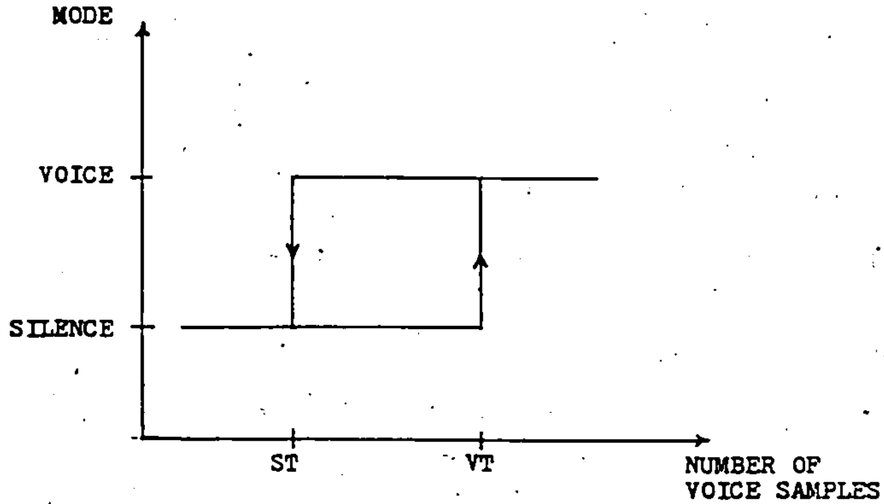


Figure 5: Hysteresis Characteristic of Silence Detector

Figure 6 depicts packet generation of a speech segment with and without silence detection. The packet generation rate is highly dependent on the thresholds settings. More packets are generated if they are small, but redundant information, such as silence, is also transmitted. This reduces the efficiency of the channel. High channel efficiency can be achieved by setting the thresholds such that only the voice signal is packetized but not the silence period. However, this requires knowledge of the statistical behaviour of the voice signal and also of the encoding algorithm used. The settings of the thresholds will be further discussed in the next chapter.

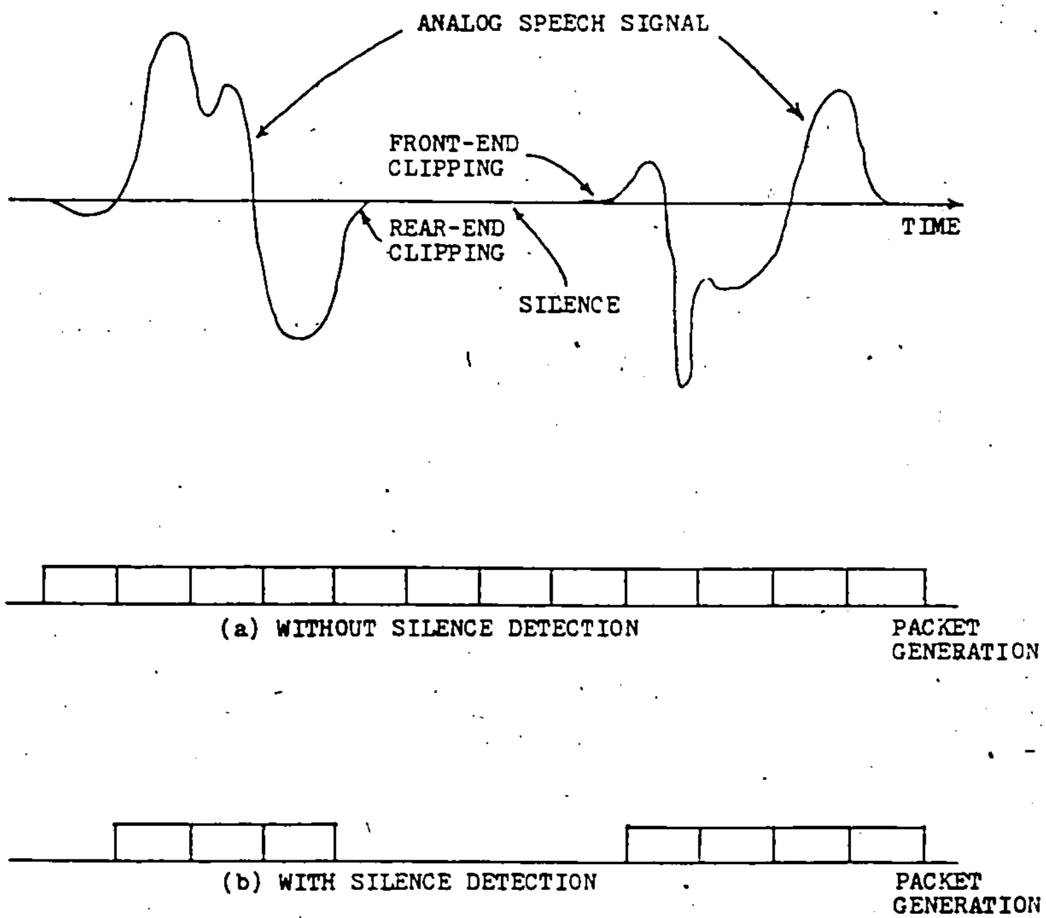


Figure 6: Packet Generation With/Without Silence Detection

Front-end clipping and rear-end clipping, shown in Figure 6, would occur due to the threshold characteristic of the switching device. The front-end and rear-end clipping signals have to be included in order to obtain a smooth speech at the receiver. Pre-offset, that is the amount of signal which is packetized before voice mode is entered, is used to eliminate the effect of front-end clipping. Rear-end clipping can be eliminated by transmitting the post-offset, the amount of signal which is packetized when the silence detector switches to silence mode [10].

3.3 PACKETIZATION

After the analog signal is converted into digital form which is then passed through a silence detector, the next phase is packetization. The size of a packet plays an important role in a packet voice system and is bounded by an upper and a lower bound. The upper bound is determined by the packetization delay which is a major factor in the total end-to-end delay. The larger the packet size, the longer it takes to form a packet. However, the packet assembly time plus other delay elements must be less than 300 ms as was explained earlier. A plot of packetization delay versus packet size for various codec transmission rates is shown in Figure 7 . In order to preserve speech continuity at the receiver, a packet may be made large enough to contain one phrase or sentence rather than only one word or syllable. This is because the variable delay is much more perceptible when occuring during the short gaps between words and syllables, than during the long gaps between phrases and sentences. However, the loss of a large packet will degrade the speech quality more than the one of a small packet.

On the other hand, the lower bound is determined by the throughput. As the packet size decreases, the packet generation rate will increase. Since the transmission medium is of the contention bus type, more collisions will be expected. The throughput will be degraded and the

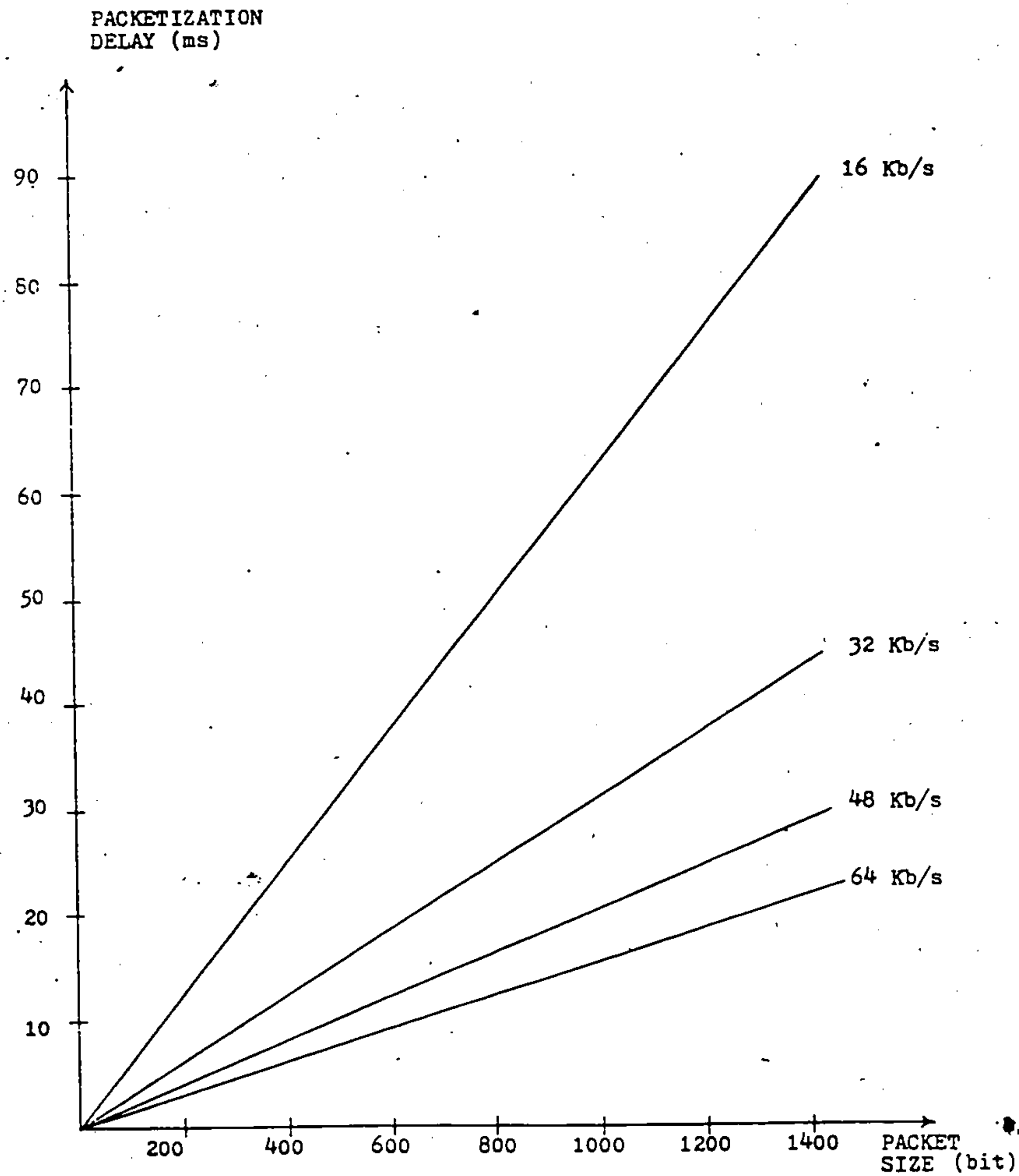


Figure 7: Packetization Delay versus Packet Size

overall network delay will be increased. Smaller packets result in excessive overhead and subsequent inefficient resource utilization. Another factor that will affect the throughput is the ratio of the propagation delay to the packet transmission time. The higher the ratio, the lower the throughput one could achieve for a given average network delay [15].

If two or more stations have the same codec rate and packet size, then their packet generation rate will be the same. In the CSMA/CD case when two or more stations are synchronous at the packet generation time, that is packets are formed at the same time, successive collisions will occur among those stations. To clarify this point, consider two stations A and B which generate packets at the same instant of time periodically as shown in Figure 8 .

As can be seen, the first talkspurt of station A as well as the first two packets of station B are generated and transmitted without any difficulty. As soon as the first packet of the second talkspurt of station A is generated at the same time that station B generates its third packet, collision occurs. They go through collision resolution and transmit successfully as indicated by the upward arrows. Since the stations generate packets at the same time and at the same rate, their successive packets will collide again if their retransmission delays are less than one packet

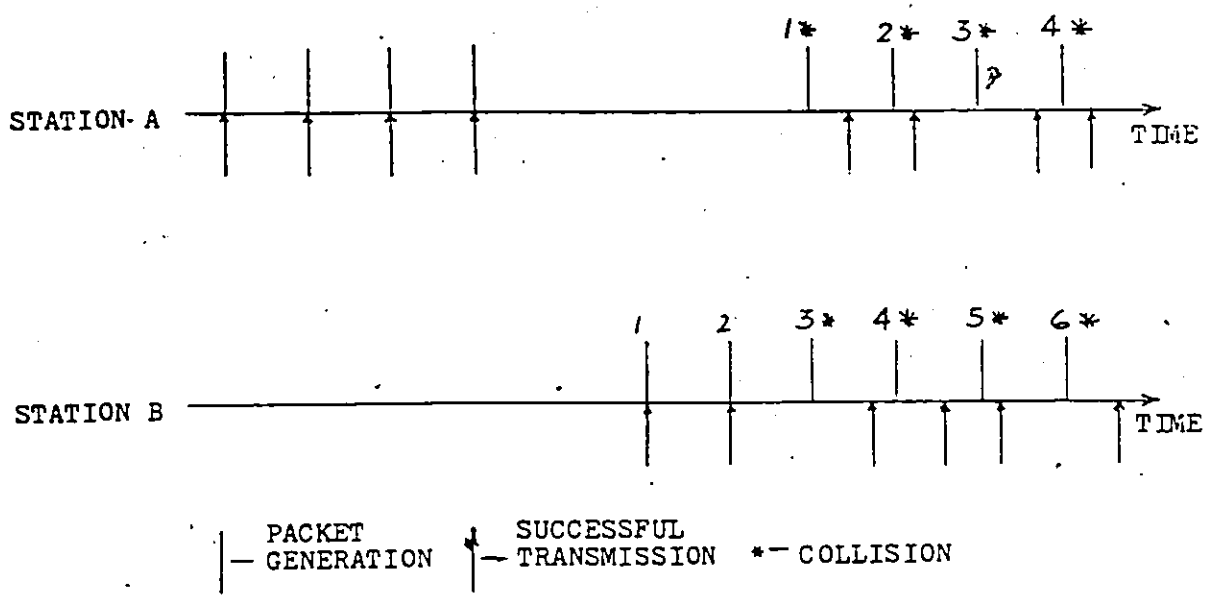


Figure 8: Successive Collisions of Station A and B

length. The next two packets that will collide are packet 2 of station A and packet 4 of station B. Collisions will carry on and on until one of the station becomes idle. The results of such a problem are an increase in network delay and a decrease in throughput. The overall system performance is degraded.

To overcome this problem, the following technique is proposed. It is based on the principle of shifting the packet generation time of a station by an amount of time that will avoid further collisions by the successive packets. If a station transmits its first packet of a talkspurt successfully, the successive slots are reserved for this station. A slot is the packet transmission time and the time between two slots is the packetization time. When another station has packets to transmit, it does not know which slots have been occupied or which slots have not. The only way to find out is to transmit its packet as usual. If there is no collision then that station has occupied the current slot as well as the successive ones. If there is a collision, the station, which attempts to look for a free slot, has to adjust its packetization time base so that there will be a free slot available when the next packet is formed. The technique of adjusting the packetization time base is described in the next paragraph.

When the two stations collide, the station which is transmitting the first packet of a new talkspurt freezes its packetization. Packetization resumes as soon as its collided packet is transmitted successfully. This procedure is shown in Figure 9 . As can be seen from the figure, packet 1 of station A and packet 3 of station B collided. Station A has to freeze its packetization, because that was its first packet, until the collided packet has been transmitted successfully. Station A moves its packetization time base to a value which ensures that there will be no further conflict with station B for the successive packets. One drawback of this technique is the temporary loss of speech during the packetization freezed period. To overcome this problem, packetization does not freeze when collision occurs. But rather it continues and as soon as the collided packet has been transmitted, the assembly of a new packet begins. The partially filled packet which is assembled when the collision occurs and stopped assembling when the collided packet has been transmitted successfully, is appended on the new packet to form an extended packet. By doing so, there will be no loss of speech.

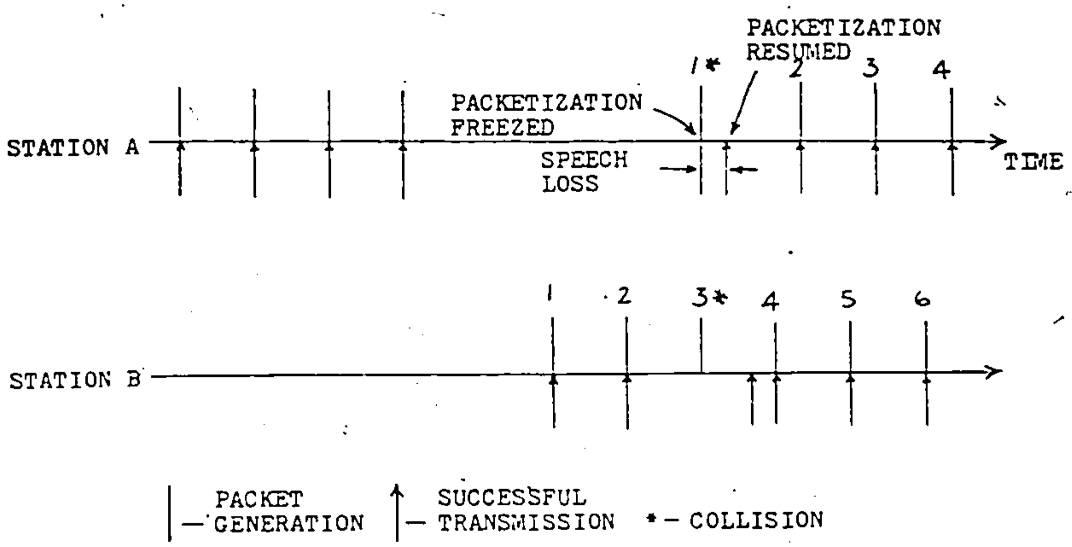


Figure. 9: Packetization-Freezed Technique

3.4 RECEIVE-END PROTOCOL

As was depicted in Figure 3, the packet voice receiver (PVR) consists of a pool of buffers, a de-packetizer and a decoder. The main objective of the PVR is to reconstruct an analog signal as close as possible to the original based on the information received. The signal degradation due to the signal conversion processes, A/D and D/A, is not of concern here. Some of the packets are lost due to the maximum allowable network delay which will be defined later. Some packets arrive at the receiver at an non-uniform rate due to the variable network delay. If the packets are delivered to the decoder the way that they are received, temporary disruption of speech might result because of the presence of gaps in between packets. It is the concern of the remaining sections to discuss various protocols for handling the packets in order to achieve a smooth speech output.

3.4.1 Waiting For Late Packets Protocol

In this protocol [8], packets are delivered to the listener the way that they are received. Consider an A talkspurt, i.e. a segment of continuous active speech with a consecutive stream of non-empty packets between a speaker pair A and B, issued by speaker A. Assume that each packet i (i=1,2,3,...) is of fixed length and takes P seconds to packetize. Then B will receive packet i at time $b(i) = D(i) + iP$ where $D(i)$ is the delay experienced by packet i. The packet transmission time and the propagation delay are ignored in this discussion.

3.4.1.1 The Variance of the Network Delay is Zero

All packets will experience a delay $\overline{N}$ and the time that B will receive packet i becomes $b(i) = \overline{N} + iP$. As can be seen in Figure 10, all packets are received at B after a fixed delay $\overline{N}$ at regular intervals and delivered to the listener synchronously. There is no gap in between packets, hence smooth speech delivery is achieved. In the figure, an upward arrow on the "packets departure from A" axis denotes the packet formation time. After a certain delay, the packet arrives at B which is indicated by the downward arrows on the "packets arrival at B" axis. An upward arrow on the "packets delivered to listener" axis denotes the time when a packet begins to be delivered to the listener.

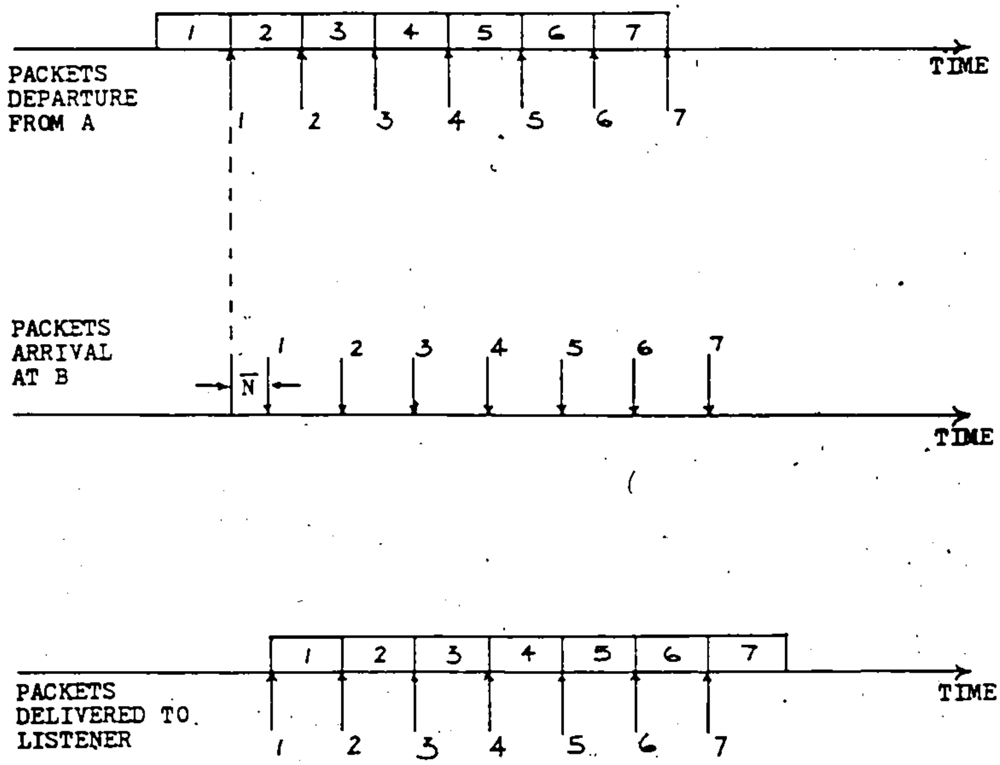


Figure 10: Synchronous Arrival of Packets

3.4.1.2 The Variance of the Network Delay is Non-zero

This is the case in a non-ideal environment. Let TF(i) be the temporal fluctuation between packets i-1 and i, as received by B. The following pseudo-code statements demonstrate the temporal fluctuation between packets.

```
        TEMP = D(1)
        DO 20, i = 2 to K
        TF(i)  = D(i) - TEMP
        IF TF(i)  < 0 THEN 10
        TEMP = D(i)
        PRINT "GAP BETWEEN PACKETS i-1 and i"
        GOTO 20
10      PRINT "NO GAPS BETWEEN PACKETS i-1 and i"
20      CONTINUE
```

where K is the number of packets.

Due to the fluctuation of the network delay, the packets arrive at B at irregular intervals. Gaps are present between packets and the length of the gap is given by:

$$G(i) = MAX\ (\ TF(i), 0\ ) = \begin{cases} TF(i) & \text{for} \quad TF(i) > 0 \\ 0 & \text{for} \quad TF(i) \leq 0 \end{cases}$$

The gap is only introduced when the term TF(i) is positive. Negative TF(i) means that packet i arrives at the receiver TF(i) units ahead of time before it has to be delivered to the listener. Table 3 is an example showing various network delays of packets with their corresponding TF(i) and G(i). The effect of G(i) on packets is illustrated in Figure 11 . As can be seen from the figure that the delay of the first packet after a gap is the maximum delay that the successive packets can have in order to obtain a smooth speech output.

TABLE 3

Example of Variable Delay on Packets

PACKET i	NETWORK DELAY D(i)	TF(i)	G(i)
1	*5		
2	3	3-5=-2	0
3	1	1-5=-4	0
4	*8	8-5=3	3
5	4	4-8=-4	0
6	*17	17-8=9	9
7	10	10-17=-7	0

NOTE: ONE PACKET IS 10 UNITS
*-MAXIMUN DELAY OF SUCCESSIVE PACKETS TO MAINTAIN A CONTINUOUS SPEECH OUTPUT

The disadvantage of this protocol is the long gap in between packets. The effect is the temporary distortion of the conversation spoken material starting from the beginning of the gap. The gaps are cumulative and if the talkspurt is too long then the cumulative delay will have a significant

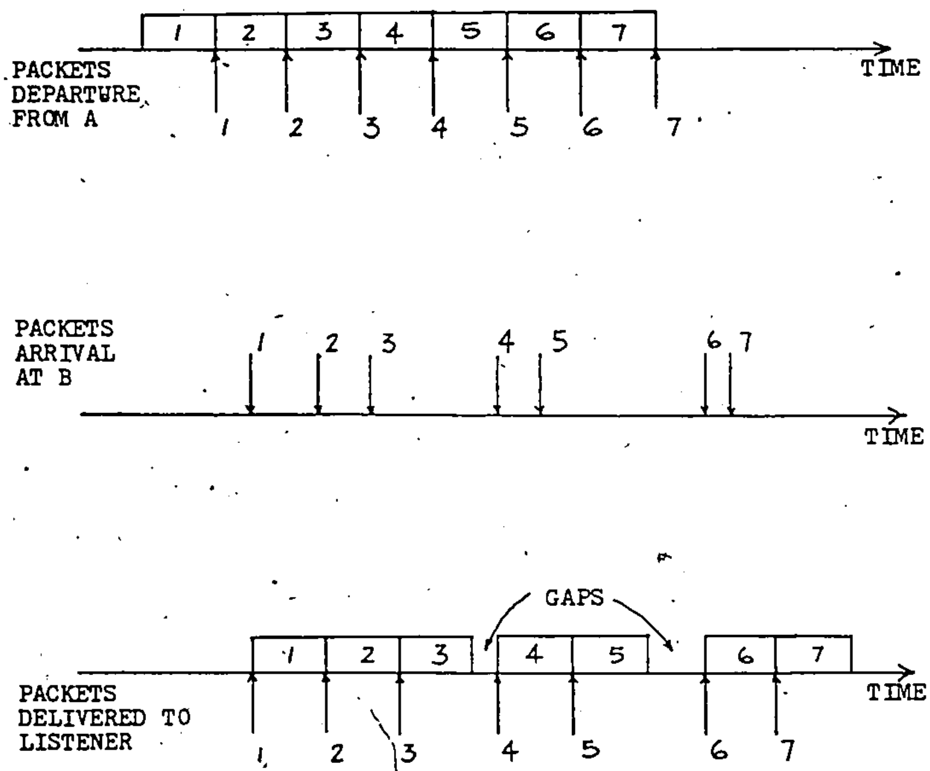


Figure 11: Asynchronous Arrival of Packets

negative effect on the speech. Consider there are K packets within a talkspurt. The time (TL) when the last packet of a talkspurt is delivered to the listener is:

i) No Gaps Between Packets (Zero Variance)

$$TL = D(1) + KP$$

ii) J Gaps Between Packets (Non-Zero Variance)

$$TL = D(1) + KP + \sum_{I=1}^{J} G(I)$$

The last term of (ii) is cumulative and will cause significant degradation on speech quality as J increases.

3.4.2 Discarding of Late Packets

When a receiver is waiting for a packet, other than the first packet of a talkspurt, which has not arrived at or before time T, where

$$T = D(1) + iP \qquad i = 2,3,4 \ldots$$

and D(1) is the network delay of the first packet, then the packet would be either lost or experiencing a large network delay. If the variance of the network delay is non-zero then the receiver can not be synchronized with the transmitter if the indefinite waiting protocol is used. In this section, a protocol is proposed to solve this problem.

A packet i is discarded, completely or partially, when it arrives at B at time b(i) under the following conditions:

1. Packet i is discarded completely when

$$b(i) \geq D(1) + (i+1)P$$

2. Packet i is discarded partially when

$$D(1) + iP < b(i) < D(1) + (i+1)P$$

On condition (1), packet i is considered as lost and a silence gap is used to replace the empty packet slot. When a packet is discarded partially due to condition (2), the front part of the packet which corresponds to a time duration of

$$b(i) - (\ D(1) + iP\ )$$

is discarded and is replaced by silence. The remaining part which has a duration of

$$D(1) + (i+1)P - b(i)$$

is delivered to the listener just as a normal packet. This protocol, shown in Figure 12, has the advantages of synchronizing the transmitter and receiver as well as eliminating the cumulative delay, at the expense of speech lost by discarding late packets. The length of the original talkspurt is preserved.

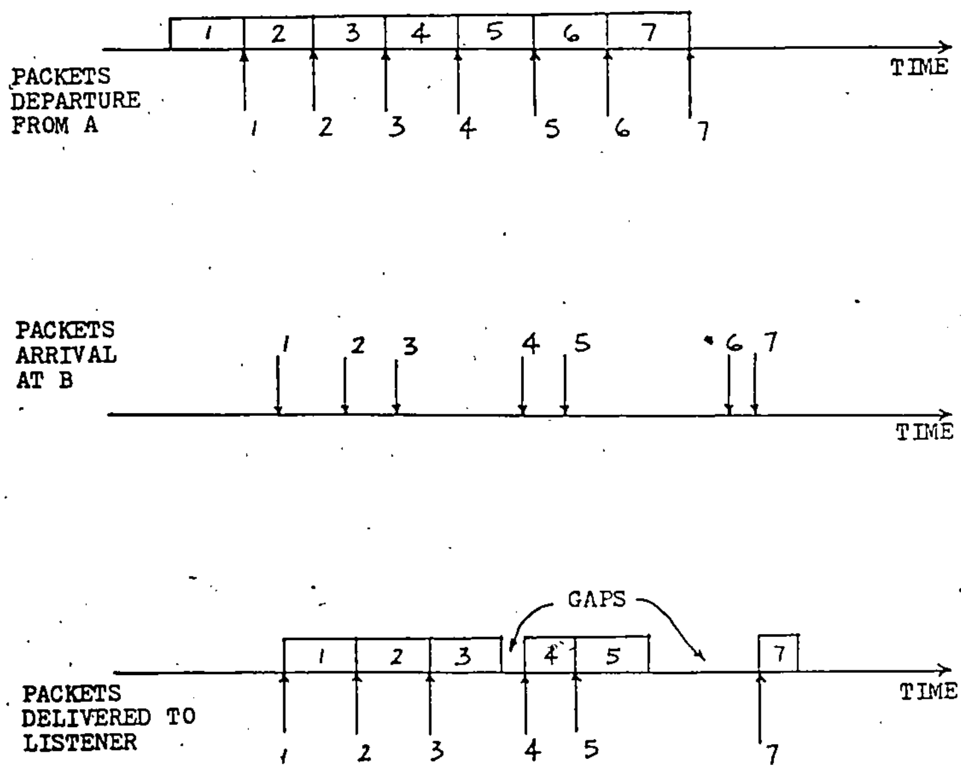


Figure 12: Discarding of Late Packets Protocol

3.4.3 Receive-end Buffering

The waiting for late packets protocol has the effect of loss of synchronization and temporary loss of speech. Synchronization can be achieved by using the discarding of late packets protocol but still temporary disruption of speech is inevitable. In this section, a buffering technique [8] is discussed and if the buffer-pool at the receiver is large enough then both synchronization and elimination of gaps can be achieved. As was described earlier, there will be no gap between packets delivered to the listener if the delay of the first packet after a gap is larger than the delay of each of the successors. So, by introducing additional buffer-delay to the first packet at the receiver before it is delivered to the listener, the temporary fluctuation of two consecutive packets becomes,

$$TF(i) = D(i) - (D(1) + BD) \qquad i=2,3.....$$

where BD is the buffer-delay and D(1) is the delay of the first packet. If BD is large enough such that D(1) + BD is always larger than the delay of packet i, D(i), then the receive buffer will never be empty during a talkspurt. The variance of the network delay is no longer affecting the smooth delivery of packets to the listener. The prices to pay for this are an increase in the end-to-end delay and buffer-pool size in return for the improved speech smoothness. A protocol of this type is depicted in Figure 13, which has a buffer-delay of two packet-lengths.

The term D(1), which is unknown to the receiver, in the temporary fluctuation equations can be eliminated by counting the buffer-delay from the end of the packetization of packet 1 instead of the time when packet 1 arrives at the receiver. The technique to perform this will be described in the next section. By doing so, the end-to-end delay of all packets can be determined and is equal to P + T + D + BD.

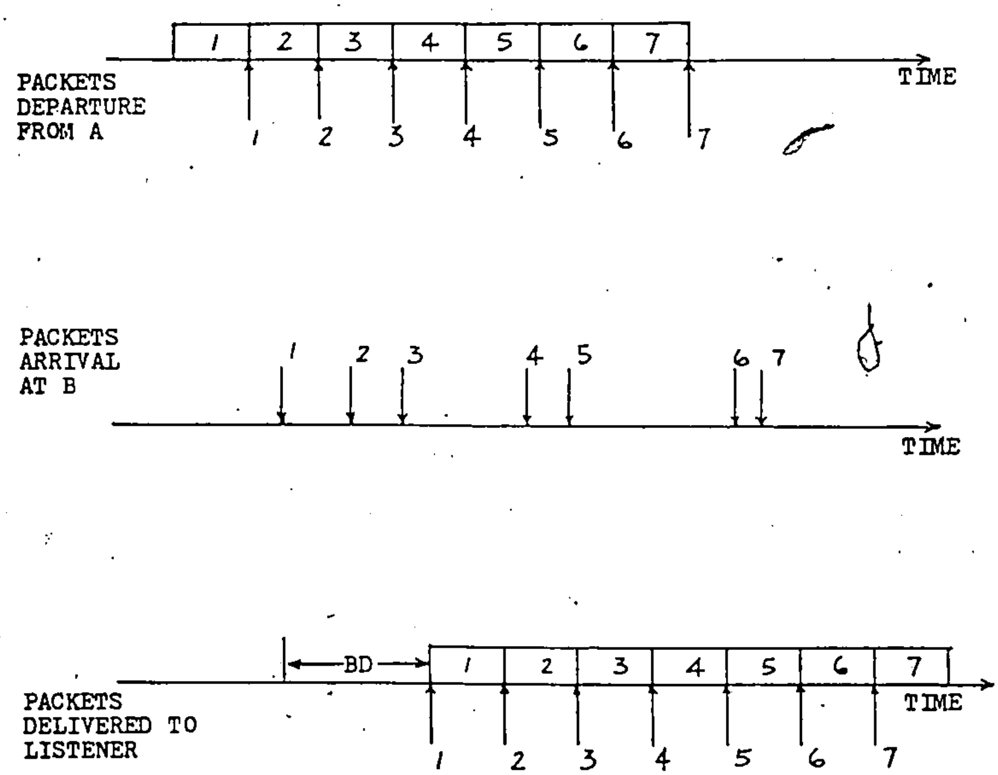


Figure 13: Receive-end Buffering Protocol

To achieve perfect synchronization, packet i must arrive at the receiver befor packet i-1 has been completely delivered to the listener. This requires a large buffer-pool whose buffer-delay must be larger than the maximum network delay. However, if the buffer-delay is excessive, two unsatisfactory conditions result:

1. too large a delay interferes with full duplex speech communication
2. too large a delay forces the use of a larger buffer-pool.

Thus, it is desirable to limit the maximun buffer-delay and discard those packets which do not arrive on time.

An extension of this protocol is to limit the network delay of all packets to a maximum. This is accomplished by discarding those packets which have not been transmitted within a maximum allowable network delay (MAND). In this case, a buffer-delay of length equal to MAND is required. This extension is implemented on the transmitter as opposed to the discarding of late packets protocol which is implemented in the receiver. It is based on the fact that it does not matter whether a late packet did actually arrive at the receiver or not, because it will be discarded anyway. This extension also has an advantage of reducing network load by discarding packets, at the transmitter, which would have been transmitted and dropped by the receiver.

3.5 SILENCE RECOVERY

As was described earlier, the silence period in a conversation is not transmitted. No information is sent concerning the length of the silence period. In this section, a technique based on two methods is proposed for recovering the silence period at the receiver : (1) based on the time-stamp and (2) based on the packet sequence number which counts the number of packets being generated. These methods differ mainly in terms of implementation. For every packet generated, a time-stamp or a packet sequence number is appended on the packet header to indicate the time or the sequence of birth. Upon reception at the receiver, the time-stamp or the packet sequence number of the packet is compared with the receiver's and it is delivered to the listener if they are the same. If there is no match, it could be due to either packet loss or silence period, zero speech values are delivered instead. By doing so, the silence period is recovered. Speech interpolation [29] can be used to improve the speech quality during the packet loss period if information concerning the silence period is sent by the transmitter.

Both of these methods require the initialization of the clocks or packet sequence counters in the transmitter and receiver at the same time. The procedure to do this is by sending a call setup packet during the call setup time. The transmit clock or packet sequence counter is started as soon as this packet has been transmitted. Upon receiving this packet, the receiver starts its clock or counter. There is only a delay of one packet transmission time plus the propagation delay. The call setup packet may be made very small, just contains enough information for initialization, to shorten the packet transmission time. If the receive-end buffering protocol is used, the receiver's clock or counter does not start until the count down of the buffer-delay plus one packet-length has expired. By doing so, the buffer-delay is referenced to the end of packetization of the first packet. If the packet sequence number is used then the silence detection must be done on a packet basis so that the exact silence period can be reconstructed.

Chapter IV

DESIGN OF A PACKET VOICE SIMULATOR

In many design problems, there are objective criteria which can be used to evaluate the performance of the system. For example, in data network protocol design, the goal is to achieve high throughput and low delay. A simulation of this system can be made and these quantities can be obtained and evaluated objectively. One would try to obtain a normalized throughput as close to unity and delay as close to zero as possible. However, in a voice communication system, the situation is totally different. The basic requirement of such a system is speech quality which consists of intelligibility, recognizability and fidelity. Different individuals have different perceptions in listening and a subjective measure is so far the best way to evaluate such a system. Based on this requirement, the output of the simulation has to be in analog form so that the subjective measurement can be carried out.

There are basically two modes in voice communication, one-way and two-way, such as voice dictation and person-to-person conversation respectively. In terms of simulation, the one-way voice communication can be done in either a real-time or a non-real-time basis. In the latter, the input speech segment is first digitized and stored in a mass storage device, such as a hard-disk before the simulation of the network environment takes place. Thus, the input signal of the simulator is in digital form. This requires a big amount of memory storage which is costly. A codec which runs at a rate of 64 Kb/s requires a storage of 120 Kbytes to store 15 seconds of speech. The output of the simulator, the speech signal, can be evaluated either on-line or off-line. The latter one is mostly adopted by storing the output speech on a tape recorder before evaluation takes place.

Real time simulation does not require so much memory storage because the input speech is not pre-stored digitally. Rather, the input speech is being digitized during the simulation phase and the simulator keeps only a few packets of speech. Thus, mass storage memory is not required. The price to pay for this is the overhead during the packetization phase in the real-time simulation program.

Simulation in two-way mode has to be done in real-time because the interactions between two users are of concern. The response is instantaneous and non-real-time simulation does not support this feature.

In simulating a local area network environment which has a round-trip delay of less than three hundredths [15] of a second, two-way mode communication does not have to be considered. In two-way mode simulation, the main objective is to study the ability of interaction between two users. Too much delay will make the conversation difficult as was described in Chapter 2. With a delay of less than 300 ms, this problem will never occur and this is why simulation of the two-way mode is not necessary in this environment. In this study, real-time simulation of voice communication in one-way mode will only be considered.

Because of the distributed control of the broadcast bus type architecture, each station does not know how many users there are in the system. In general, the traffic on the bus is proportional to the number of users. As the number of users increases, the delay of the packets as well as the packets being discarded start to increase. In such a system, one can either simulate a single station with network delay distribution and packet discard probability as the system variables or the whole system with the number of users as the system variable. The former scheme is adopted in this study because of the ease of implementation. It is assumed that packets behave independently of each other in the sense that a given packet will not influence the behaviour of future packets. The network delay distribution and packet discard probability are obtained from a separate simulation program written by Rios [15]. The inputs of this simulation program are:

- round-trip propagation delay
- jamming time

- separation time between packet transmissions
- packet transmission time
- number of stations
- average packet generation rate of the stations
- maximum allowable network delay.

The outputs of the simulation program are:

- network delay distribution
- average packet discard probability
- throughput
- average network delay

The average packet discard probability is the ratio of the total number of packets being discarded to the total number of packets being generated among all stations. A packet is discarded if the sum of the queueing and the contention delays exceeds the maximum allowable network delay. This was explained in the extension of the receive-end buffer protocol. The network delay distribution is expressed in terms of the percentage of packets and the amount of time that the packets have to wait in the transmitter before a successful transmission is achieved. These two parameters will serve as the inputs to the packet voice simulator. The simulation program requires about one hour to simulate, on a VAX 11/750 computer, 150 stations with a packet generation rate of 36 packets/sec. According to the author of the program, no effort was ~~put to~~ shorten the execution time.

The packet-voice simulator is designed according to the requirements which were described in Chapter 3. The processes of the simulation are summarized below:

1. Input analog speech signal is band-limited and digitized.
2. The digital signal is assembled into packets. Speech-silence is not transmitted. The packet sequence number is used for silence recovery.
3. The packet discard probability determines the number of voice packets discarded from the system. When a packet is not discarded, it is delayed by a random amount of time drawn from the network delay distribution.
4. At the receiver, receive-end buffering protocol is used and packets are delayed before being delivered to the listener. In no buffering mode, late packets are handled by the waiting for late packets protocol.
5. Silence is played during packet loss and speech-silence.
6. The analog speech signal is reconstructed by digital-to-analog conversion followed by low-pass filtering to remove the quantization noise.

The packet-voice simulator is based on an Intel System Design Kit SDK-86 which uses a 16-bit CPU, the 8086. By using a microprocessor based system, the hardware count is reduced and the flexibility of the system is increased. The latter is achieved by software which can be changed easily according to the system's requirements. The simulator will be described in two parts, the hardware and the software.

4.1 THE SIMULATOR HARDWARE

A block diagram of the packet-voice simulator is shown in Figure 14 . The input analog speech signal is band-limited and is digitized by the encoder. The serial digital data stream is converted to parallel by a serial-to-parallel (S/P) converter and is then fed to the SDK-86. After the data has been processed by the SDK-86, the parallel-to-serial (P/S) converter converts the output data to serial form. The decoder converts the digital data to analog form and the quantization noise is removed by low-pass filtering. The square-wave generator provides different clock frequencies to the codec. The hardware is described in two sections, the support hardware and the SDK-86.

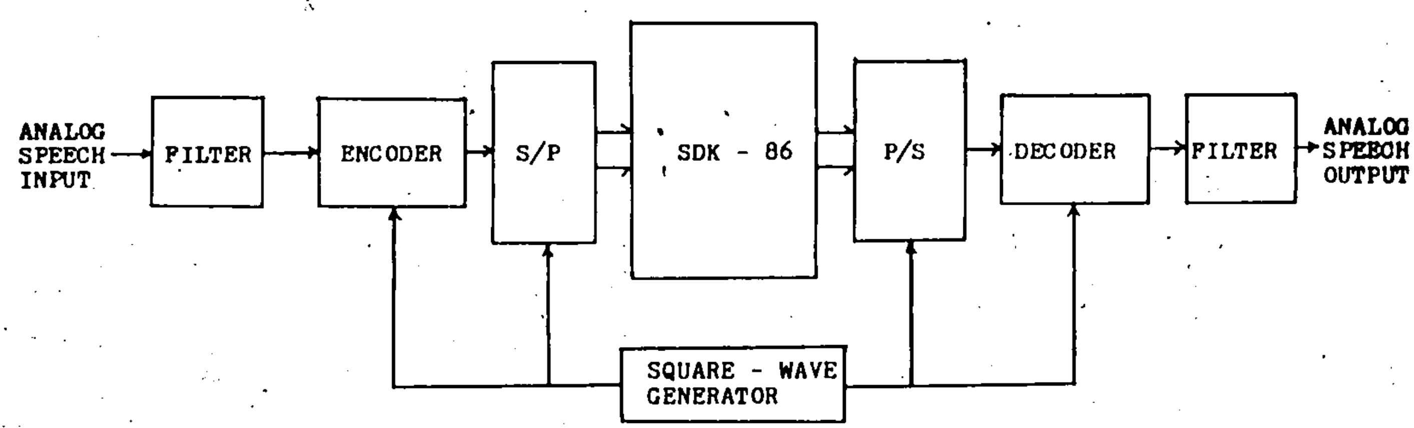


Figure 14: Block Diagram of the Simulator

4.1.1 The Support Hardware

The support hardware consists of the input and output filters, encoder, decoder, square-wave generator, S/P and P/S converters. The input filter is a second order Butterworth filter with a cut-off frequency at 3200 Hz. An elliptic filter with a cut-off frequency at 3000 Hz is used for the output. It provides excellent performance for 12 Kb/s to 40 Kb/s systems [24]. The output filter is more crucial than the input one because steep roll-off is required at the output to remove the quantization noise.

The speech encoding/decoding is done by a CVSD modulator/demodulator (MC 3417 - Motorola) [24]. It can work either as a encoder or a decoder alternately but not both at the same time. A block diagram of the CVSD encoder and decoder is shown in Figure 15 . The only difference between them is the addition of a comparator in the encoder. The digital output of the encoder reflects the difference between the input voltage and the integrator output. It also controls the direction of ramp in the integrator. The level detect algorithm monitors the passed three bits of the digital output and changes the gain of the integrator when they are all 1s or 0s. External components, capacitors and resistors, are required in the feedback path of the integrator and the level detector within the IC. The integrator is made lossy so that during any loss of packets, the decoder output will decay to zero.

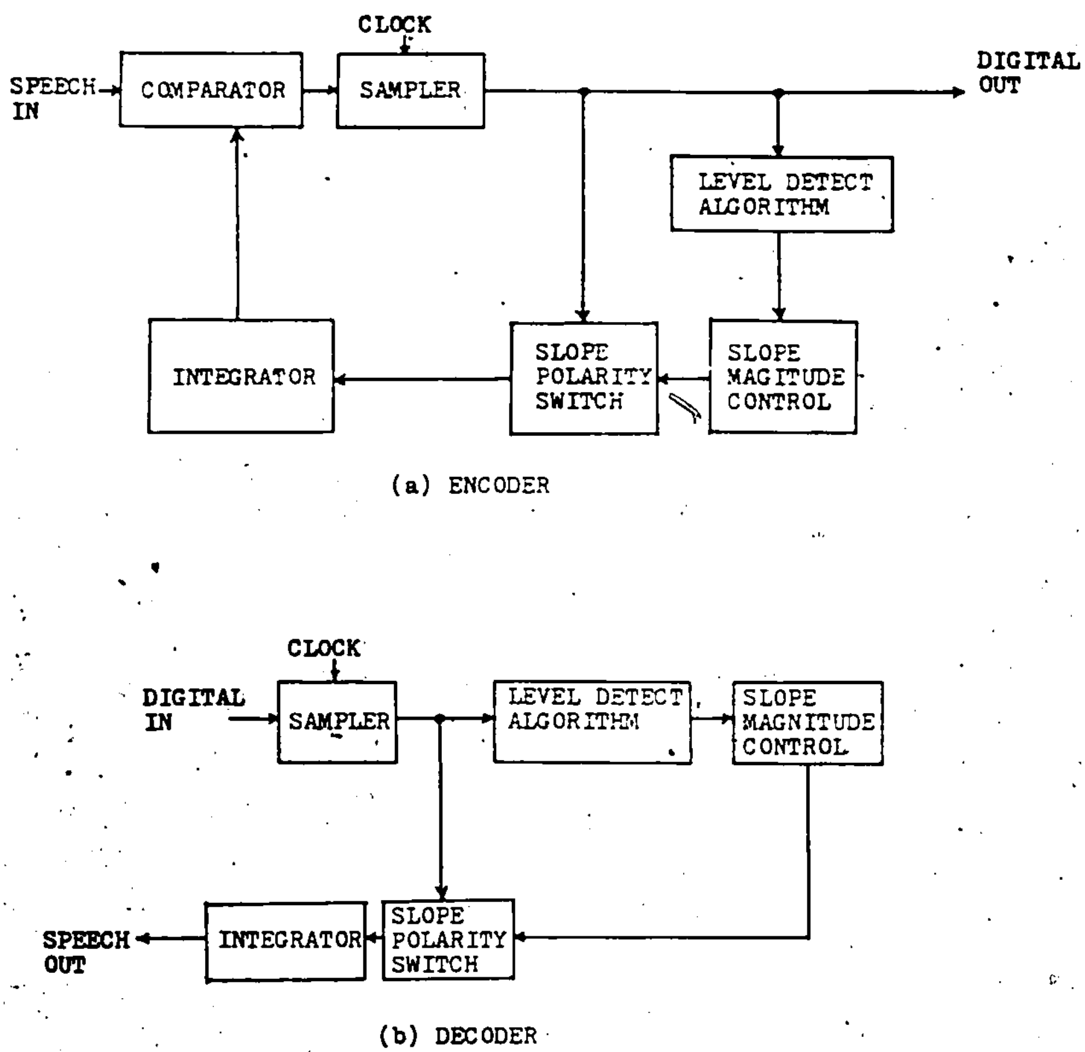


Figure 15: Block Diagram of the CVSD Encoder and Decoder

The clocks of the encoder and decoder are obtained from a square-wave generator which provides a frequency range of 16 KHZ to 64 KHZ with a 50% duty cycle. The S/P and P/S converters are 16-bit wide and are connected to the parallel input and output ports, respectively, on the SDK-86. The clockings of the digital data out of the encoder and into the S/P converter are done on separate edges of the same clock pulse to avoid timing problems. The same holds true for the P/S converter and the decoder. The data transfer from the S/P converter to the SDK-86 and from the SDK-86 to the P/S converter is done on every 16-bit interval. When 16 bits of data have been assembled on the S/P converter, the CPU is informed to read the data by means of an interrupt.

By using interrupt, possible loss of data can be avoided. To clarify this point, consider that the CPU is going to execute a routine whose execution time is longer than the time that the S/P converter requires to assemble 16-bits of data. If an interrupt is not used and the execution of the routine is just started then the parallel word will be over-written and lost before the CPU would have a chance to retrieve it. Also, the CPU intervention can be reduced which is very important in a real-time simulation. This is because most of the jobs done by the CPU are on a real-time basis and a slow down in CPU execution time will often result in simulation error. A schematic diagram of the support hardware is shown in Appendix A .

4.1.2 The SDK-86

The central unit of this simulator is the SDK-86 [25] which performs all the functions of a packet voice station, except those done by the support hardware. The SDK-86 is based on the Intel 8086 [26] 16-bit CPU running at 5 MHZ. The CPU is housed in a reliable 40-pin package and can address up to 1 megabyte of memory along with a separate 64 Kbyte I/O space. The high performance of the 8086 is realized by combining a 16-bit internal data path with a pipelined architecture that allows instructions to be prefetched during spare bus cycles. The CPU has eight 16-bit general registers and four of them can also be used for index and pointer purposes. The megabyte of 8086 memory space is divided up into logical segments of up to 64 Kbytes each and which are addressed by four segment registers. This CPU features powerful addressing modes such as the based-indexed addressing which is of convenience in handling data arrays.

The SDK-86 is a System Design Kit which consists of an 8086 CPU, 8K ROM, 2K RAM, 48 parallel lines, a RS-232 serial interface and a prototype area for system expansion. The 8K ROM is the system monitor which can be operated directly on the built-in keyboard or a terminal via the RS-232 interface. The system monitor commands include examine memory/registers, read/write hexa-decimal file, single step execution, input/output parallel data and breakpoint insertion for program debugging. An 8 Kbytes static RAM is expanded on the prototype area for both program and data storage. An interrupt acknowledgement logic circuitry which is of vector type "44" is added for interrupt hand-shaking purpose.

4.2 THE SIMULATOR SOFTWARE

In this section, the software requirements of the simulator will be given. A general flow-chart of the simulator software is shown in Figure 16 . All the functions that the simulator has to perform can be partitioned into four parts. They are initializaton, computation, interrrupt service and packet handling. For simplicity, the general flow-chart in Figure 16 does not show all the actual interactions, which will be explained later, among the four routines. The functions of the routines are given below in descending priority order:

1. Interrupt Service Routine (ISR) - input and output data, assembly and disassembly of packets, silence detection and delay count down.
2. Packet Handling Routine (PHR) - affix network delay on packets, packet discarding, get empty buffer for next packet, transmit packet, prepare packets for output and return empty packet to buffer-pool.

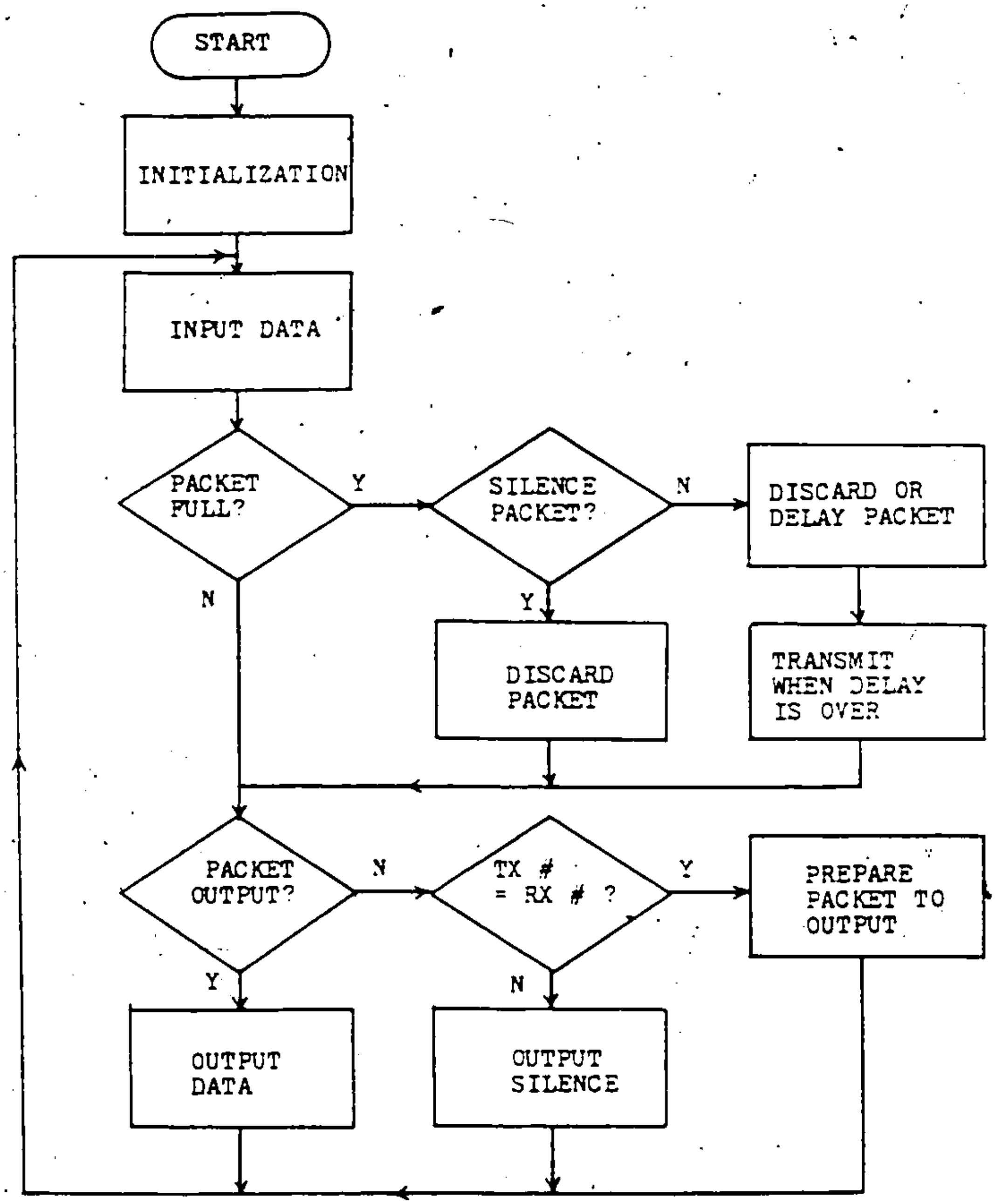


Figure 16: Flowchart of Simulator Operations

3. Computation Routine (CR) - generate random network delay and packet discard decision.
4. Initialization Routine (IR) - buffer-pool, transmit-pool, receive-pool, flags and variables initializations.

The ISR has the highest priority in execution because the input data will be over-written if it is not read before the next one is available. The second priority is assigned to the PHR because its functions have to be serviced immediately when required. The time of execution of the CR is not that crucial and it is assigned the third priority.

The interactions as well as the priority execution among the routines can be described by a state diagram as shown in Figure 17. The transitions from the CR and the PHR to the ISR are done by hardware interrupts and the transition paths are indicated by a and b respectively. Upon completing execution of the ISR, the execution resumes where they were from by following paths a' and b'. In a situation in which a transition is made from CR to ISR and PHR has to be serviced right after the end of execution of ISR, the return path a' is changed to c. This is accomplished by changing the return address to the address of the PHR. When the execution is finished in the PHR, the execution is returned to the CR by following path c'. The functions of the four routines are now described in more detail.

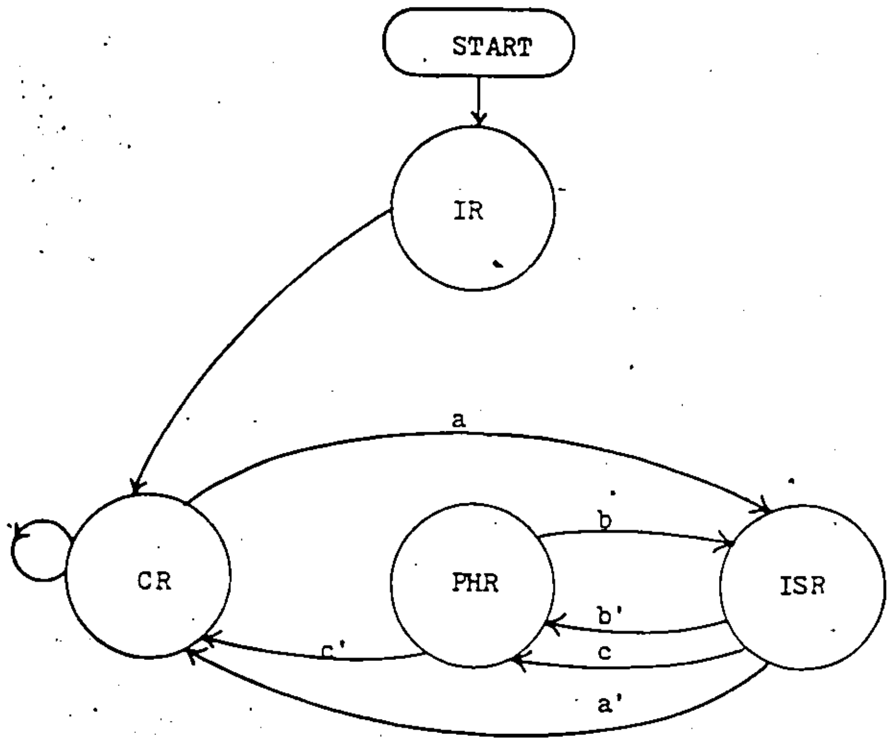


Figure 17: State Diagram Showing Interactions Among Routines

4.2.1 The Initialization Routine

The low-level tasks such as stack pointers, variables, input/output ports and interrupt handling are initialized in this routine. Two stacks will be used in the simulation program. One of them is the system stack which is mainly used for the storage of the return address and/or system flags for subroutine and interrupt operations. Another one is the user stack which is used for the storage of transmit and receive packet pointers and sizes. No operands are required for stack operation thus the execution time is shorter than if operands are used. There are six flags which are used to identify the states as well as the operations of the system. They are:

1. TRS - Transmitter Requires Service

   This flag is set when a packet is filled up and a new one is required.

2. BM - Buffer Mode

   It indicates the receiver is buffering packets and no output of packets is done.

3. RRS - Receiver Requires Service

   This flag is set when a packet has been delivered to the listener and the next one is required.

4. PKHAND - Packet Handling Requested

   It is set to indicate the packet handling routine has to be executed next.

5. DELOVER - Delay Over

   It indicates the delay count down of a packet has finished and transmission is to take place.

6. SILMODE - Silence Mode

   It indicates the status, voice or silence mode, of the silence detector.

All the packets in the system are handled in terms of pointers. They are held in three separate first-in-first-out (FIFO) queues: buffer-pool, transmit and receive. The buffer-pool queue (BPQ) holds all the empty packet pointers. The transmit queue (TXQ) contains voice packets waiting to be transmitted. All packets arriving at the receiver are queued in the receive queue (RXQ). The assembly of a packet starts by retrieving an empty packet pointer from the BPQ. It is put in the TXQ when a packet has been assembled and is sent to the RXQ after a certain network delay. All packet pointers are circulated within the simulator in a round robin fashion except in a case whereby a packet is discarded and its pointer is returned directly to the BPQ from the TXQ. The FIFO queues are implemented by software. There are two operations in using the queues, namely, fetching and releasing. The initialization of all the queues as well as the data buffer partitioning are also done in this routine.

4.2.2 The Computation Routine

The arithmetic computation that is required in the simulator is the generation of random numbers. A pseudo-random sequence is generated based on the multiplicative congruential formula [27]

$$Z(i+1) = kZ(i) \pmod{m}$$

and is computed recursively. Ref. [27] has also shown that for the multiplier $k = 65539$ and modulus $m = 2^{31}$, the sequence period is maximized and the serial correlation is minimized. The only condition is that the initial value must be an odd integer less than 9999. The modulus can be implemented easily on a 16-bit microprocessor by keeping the least significant 31 bits of the product. Hence, no division is needed. The full product result is not necessary because the most significant bits will be discarded anyway by the modulus. The random numbers range from 1 to $2^{31}-1$ and are uniformly distributed.

The random numbers are used to generate the decision of discarding a packet and the random network delay. The former is achieved by generating a random number for a given packet and if it falls between the value of 1 and the packet discard threshold then the packet is discarded. The packet discard threshold is the product of the packet discard probability and the maximum random number which is $2^{31}-1$.

The random packet delay generation is done by means of a look-up table. The domain and range of the table are the delay threshold and the network delay respectively. Let N(i) and P(i) be the network delay and the percentage of packet respectively of interval i; where i= 1,2,3,...,K and K is the number of intervals of a network delay distribution. Let MAX be the maximum random number generated. The delay threshold , DT(i), is the product of P(i) and MAX where i=1,2,3,...K . To generate a random network delay, a random number (R) is first computed. Find i such that $DT(i-1) < R \leq DT(i)$, where $DT(0)=0$. The random network delay is then $N(i)$.

The processes of generating a packet discard decision (PDD) and a random network delay (RND) are quite time consuming because they both involve a few multiplication operations. In a real-time simulation, such a slow process has to be bypassed. This is achieved by generating them before they are needed and store them in two separate FIFO queues. The PDD queue contains elements which are either one or zero. A one corresponds to "discarding" and a zero corresponds to "no discarding". The elements of the RND queue are the amount of network delay. The retrieve operation of an element from the queue takes an insignificant amount of time. Figure 18 is a flowchart of PDD and RND generations.

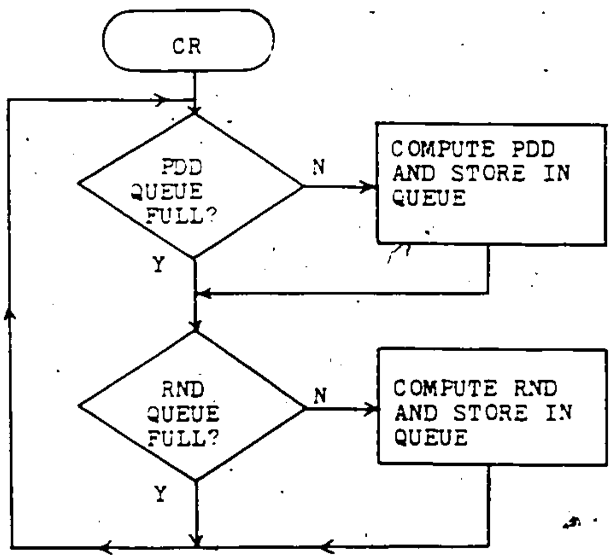


Figure 18: Flowchart of Computation Routine

4.2.3 Interrupt Service Routine

The execution of this routine starts when an interrupt signal is received by the CPU. The interrupt signal is active when 16-bit data has been assembled in the S/P converter. To prevent input data over-run, the execution time of this routine must be kept small so that the execution of this routine is finished before the next set of data is available. Basically, the functions of this routine are limited to packetization, silence detection, de-packetizaton and packet delay count down. A flowchart, which demonstrates the operation of this routine, is shown in Figure 19 . The silence detector counts the number of voice samples when a packet is being assembled. A voice sample is 8-bit long and does not have any periodic bit pattern such as 10101010 or 01010101. After a packet is formed, the number of voice samples is compared with the voice and silence thresholds. If the number exceeds the voice threshold then the transmitter switches to voice mode and in this mode all packets generated will be transmitted. Silence mode is reached when the number of voice samples is less than the silence threshold. Pre-offset and post-offset are also implemented to eliminate the front-end and rear-end clippings respectively. A flowchart of the silence detection is shown in Figure 20 .

If the receiver is not empty, the de-packetization routine will compare the transmit and the receive sequence number and determine if additional buffering is required before packet dis-assembly begins. The number of packets requiring buffering is determined by the maximum network delay allowed. A flowchart of the de-packetization phase is shown in Figure 21 .

Every voice packet has a random delay value added to it which is decremented in this routine. The DELOVER and PKHAND flags are set when the count-down of a packet reaches zero to indicate that packet transmission has to take place.

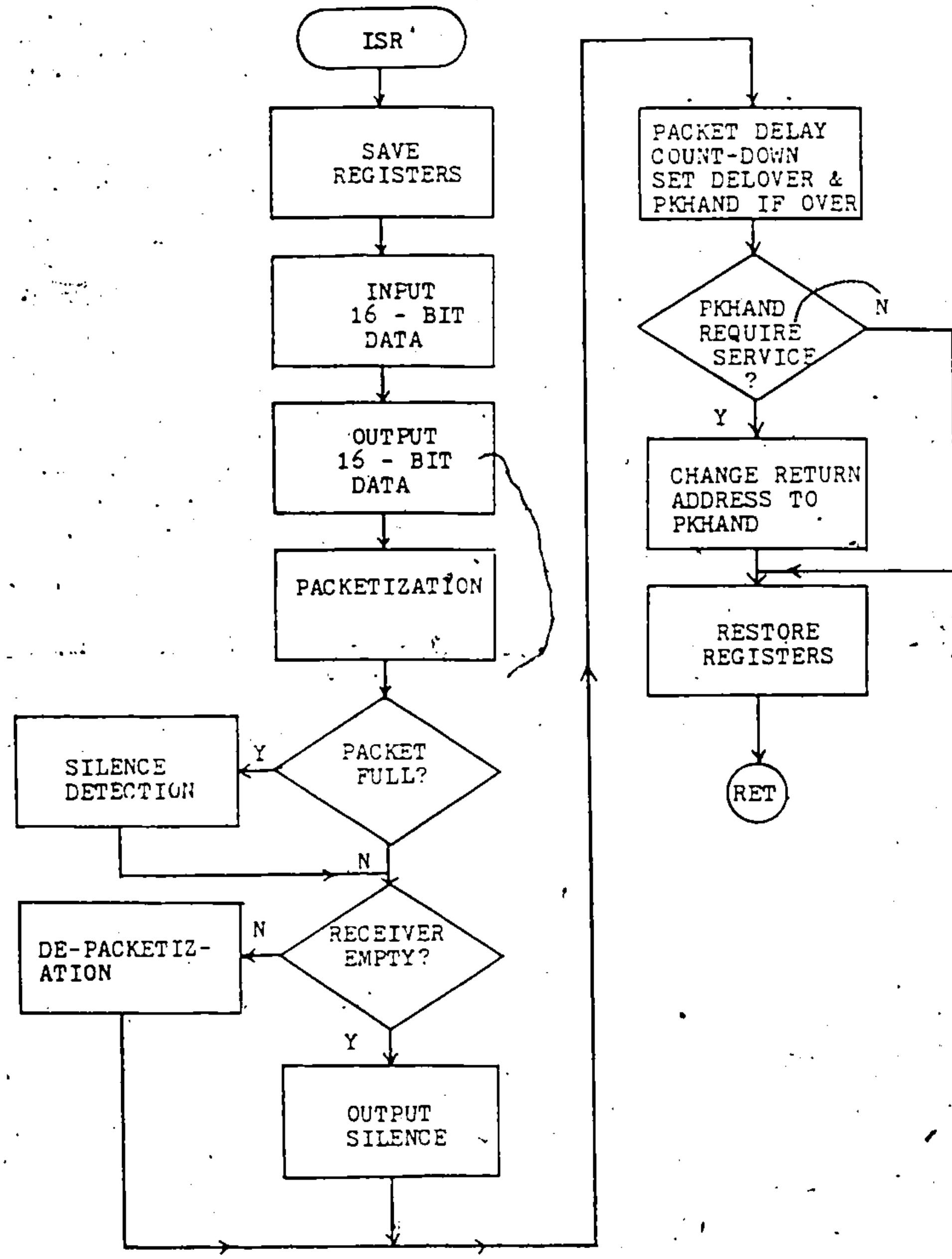


Figure 19: Flowchart of Interrupt Service Routine

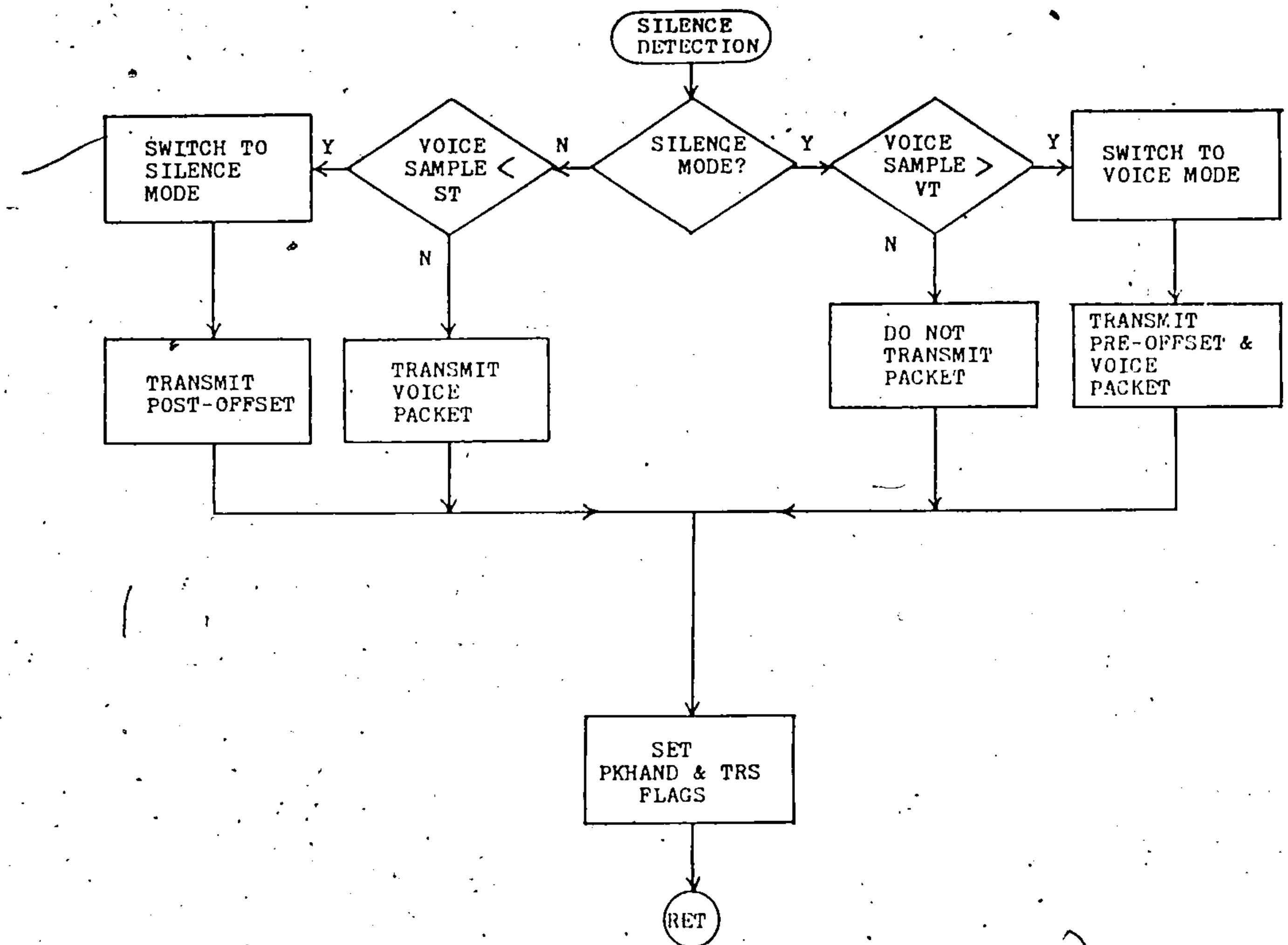


Figure 20: Flowchart of Silence Detection

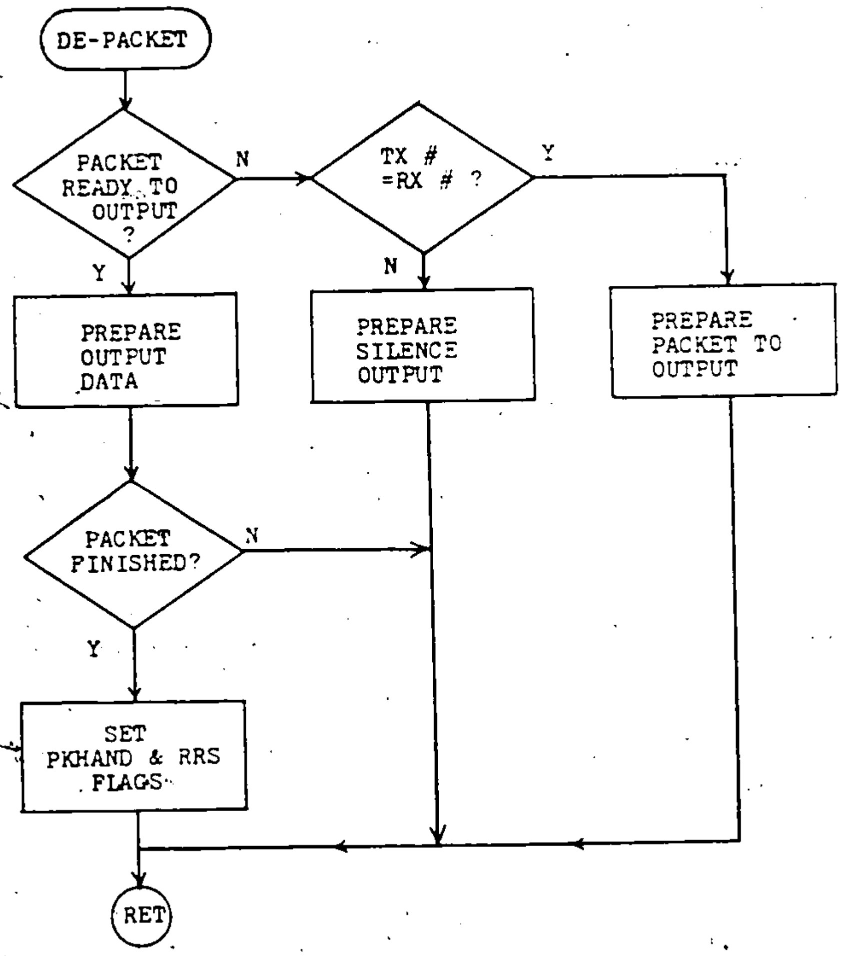


Figure 21: Flowchart of De-packetizaton Operation

4.2.4 Packet Handling Routine

This routine is invoked as a consequence of the completion of any of the following jobs:

- voice packetization,
- voice de-packetization,
- packet delay expired.

A flowchart for this routine is shown in Figure 22 . As can be seen from the flowchart, the circulation of packets in the simulator is done in this routine. The flags RRS, TRS, and DELOVER are set independently of each other. So, ~~the~~ routine loops on itself until all the jobs have been completed before exiting. The flag is reset after each job has been completed.

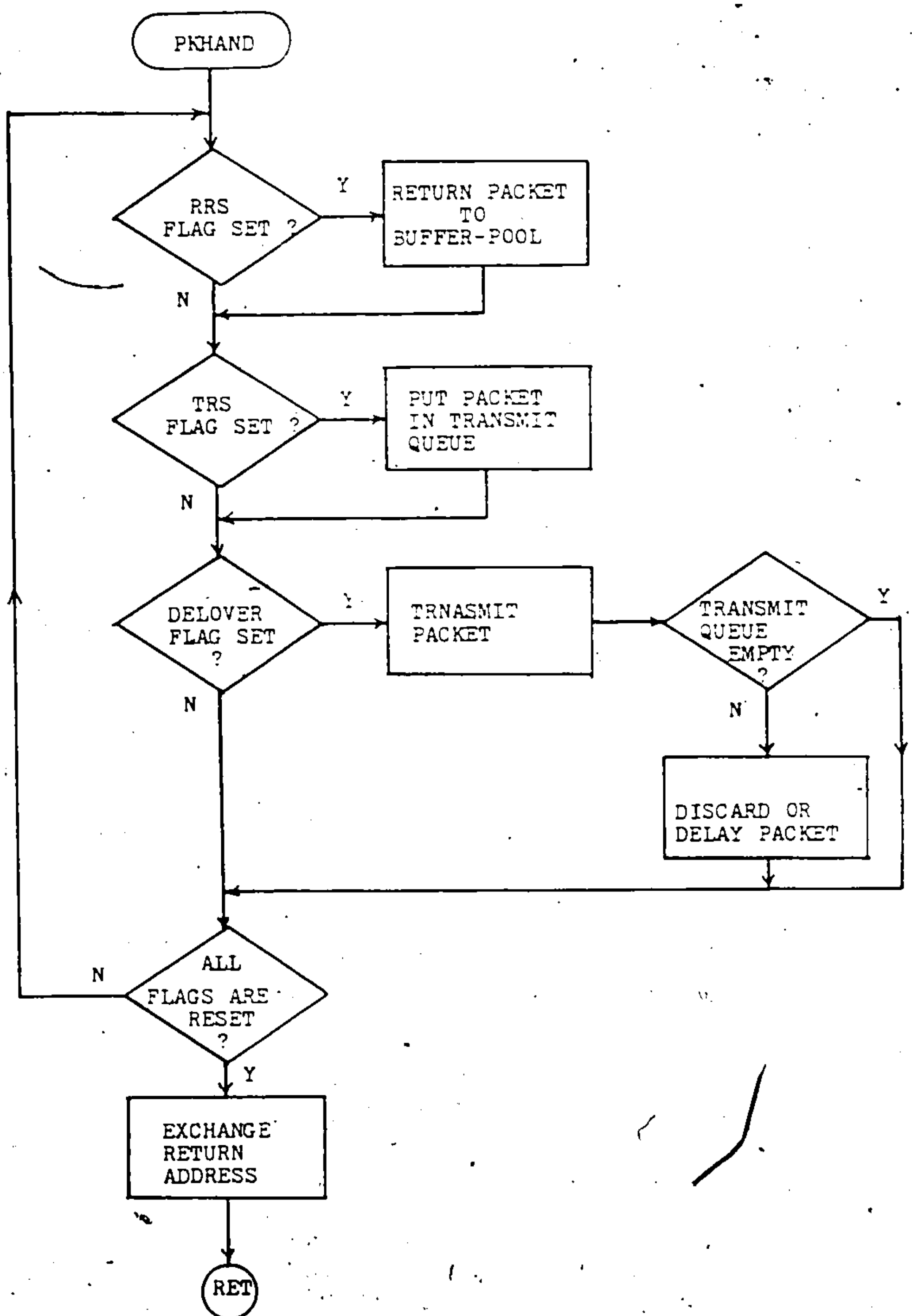


Figure 22: Flowchart of Packet Handling Routine

4.3 SIMULATION EXPERIMENTS AND RESULTS

An 8086 Assembly Language program was written as described by the flowcharts in the last section. Less overhead and thus faster execution time can be achieved in the low-level language such as Assembly as compared to the high-level languages such as Fortran. Fast execution is essential in real-time simulation as was explained in the last section. The program consists of four modules which correspond to the four routines described earlier plus a data module. The program is entered in a MDS-800 microcomputer and assembled by an 8086 cross-assembler. The program, after linking, relocation and obj-hex conversion, is down loaded into the SDK-86 for execution. The program listing is in appendix B.

The main objective in this simulation is to determine the number of users that can be supported in the system with the corresponding speech quality. Before doing so, the encoding rate, packet size and silence detection thresholds will be obtained first. An informal listening test is used for the speech quality evaluation. The speech material is pre-recorded on a tape recorder and is passed through the packet-voice simulator. The output speech is recorded on another tape recorder. Editing, such as inserting intervals in between adjacent speech signals and allowing time for the listeners to respond, is done before the speech test signals are evaluated by the listeners. The testing is performed with a good quality amplifier and headphone. The reason for using a good quality headphone instead of a telephone handset is to obtain the lowest bound of the number of users. Noise or silence gaps, which may not sound obvious on a telephone handset, are more apparent on headphones because of their good frequency response.

Three listeners are asked to rate the test speech signals from one to five: Excellent(5), Good(4), Fair(3), Poor(2) and Unsatisfactory(1). In order to establish a point of reference for listener's responses, two speech reference signals are presented (one excellent and one unsatisfactory) to the listener before the evaluation takes place. The speech test signals are presented to the listeners in random order. For every set of five speech test signals, the speech reference signals are presented to refresh the standard of references to the listener. The experiments to be performed later are based on this method for speech quality measurement. The speech sentences used for the speech test and reference signals were obtained from reference [19]. The term "number of users" used in the later experiments refers to the number of voice circuits.

The selection of the codec transmission rate is very important because it affects the number of users as well as the performance of the system. The objective is to select a transmission rate as low as possible so that it occupies less of the channel capacity. On the other hand, it should be high enough to encode the speech information with good intelligibility, recognizability and fidelity. Experiments are carried out by varying the transmission rate from 16 Kb/s up to 64 Kb/s, in steps of 8 Kb/s, and the corresponding speech quality is evaluated. The packet-voice simulator is bypassed in this experiment. The results showed that there was no significant changes in speech quality between the encoding rate ranging from 40 Kb/s to 64 Kb/s. However, the speech signal is noticeable more bass because of band-limiting. Below 40 Kb/s, the quantization noise becomes more and more noticeable as the encoding rate goes lower and lower. The speech material was barely understood at 16 Kb/s and the quantization noise was so large that the recognizability and fidelity were very much degraded. From this experiment, the encoding rate of 40 Kb/s was found to be the lowest that provides a satisfactory speech quality. Hence, it is used throughout the experiment. It is rated as "Good" in the category scale by the listeners and is considered as "toll" quality.

The packet size affects the network delay as well as the speech quality. A loss of a large packet will degrade the speech quality more than the one of a small packet. But the smaller the packet is, the higher the network delay will be.

This is an optimization problem and is very difficult to solve theoretically. Rather, an off-line simulation program written by Rios [15] was used to determine the packet size which gives the minimum average network delay for various number of users. The results of using CSMA/CD and HYMAP protocols are shown in Figure 23 and Figure 24 respectively. The packet length is expressed in time based on an encoding rate of 40 Kb/s. As can be seen from the figure that the average network delay is highly dependent on the number of users. The less the number of users is, the lower is the average network delay. In CSMD/CD, for a given number of users, there exist a packet length which gives the minimum average network delay. It is assumed that in the packet-voice system, the average number of users is 100 and the corresponding packet length, which gives the minimum average network delay, is 27.5 ms. Furthermore, the packet length obtained falls into the range of 16 to 32 ms, which Jayant [29] showed to be robust in terms of packet loss. For HYMAP, the average network delay decreases as the packet length increases. The minimum is not reached for a packet length of 77.5 ms which is much greater than the packet lengths that Jayant recommended. So, the packet length used in the later experiments will be based on the one obtained in CSMA/CD.

The settings of the silence detection thresholds, VT and ST (sect. 3.2), are done experimentally. The main objective

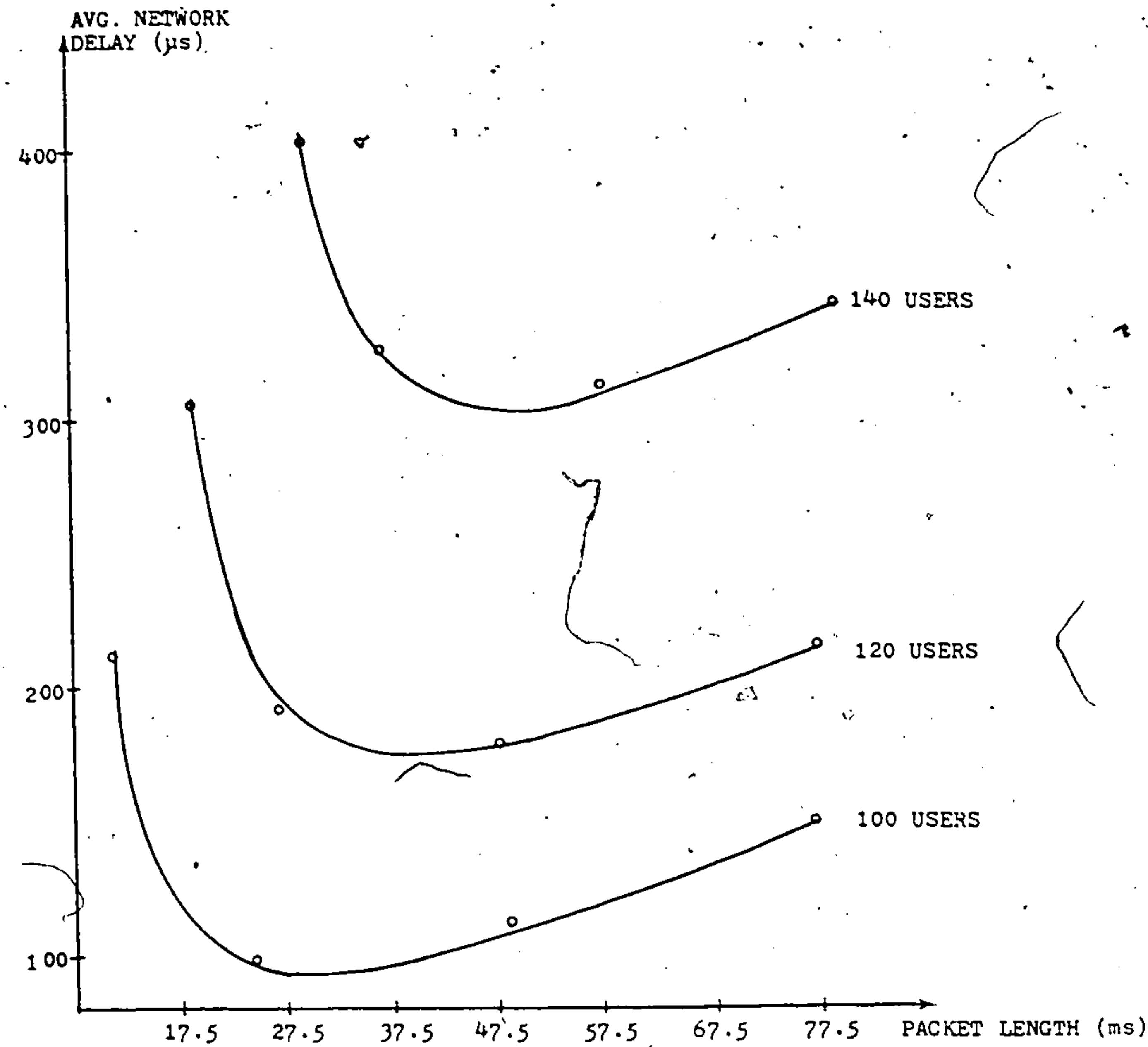


Figure 23: Packet Length Versus Average Network Delay, CSMA/VD

is to generate packets only when there is voice. So, one would try to generate as few packets as possible and at the same time the speech quality should not be affected. The testing speech material used in this experiment is obtained from a BYTE magazine article [30] and is three minutes long.

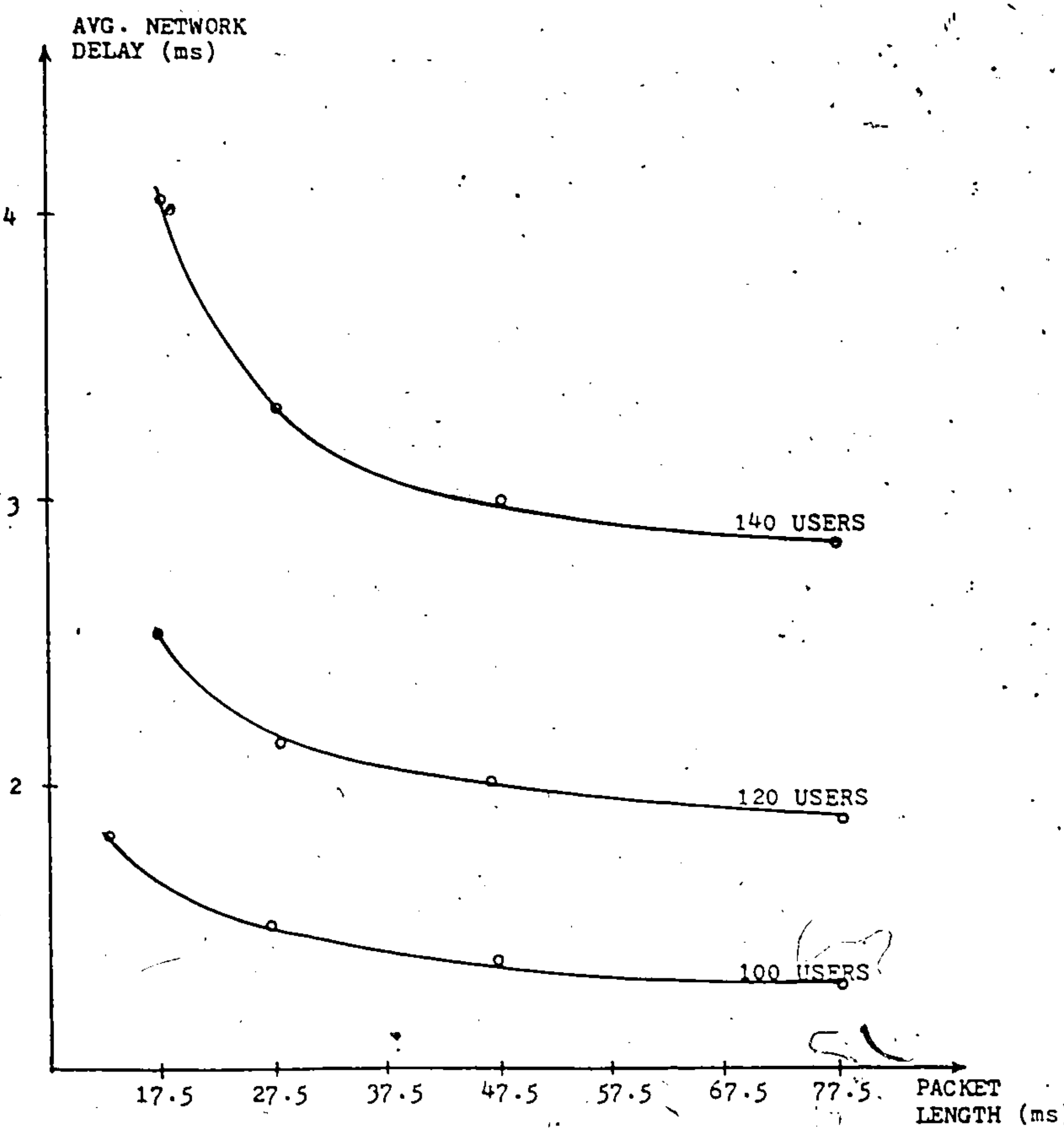


Figure 24: Packet Length Versus Average Network Delay, HYMAP

The number of packets generated (%) for various silence and voice thresholds (% of packet length) is plotted in Figure 25 . Note that the percentage of packets generated in this experiment has "relative" meaning only. The speech material used and the monologue are by no means representative of conversational speech. The speech quality is not affected when VT and ST are less than 75% and 55% respectively. When VT increases, syllables begins to be lost. Chopping noise between syllables becomes obvious when ST is above 75% and the speech quality is degraded. Based on the result of this experiment, ST and VT are set to be 55% and 75% of the packet length respectively. For all settings in which the speech quality is not affected, the pre-offset and post-offset were set to be 22% of the packet length. This is the advantage of detecting silence and voice in a packet basis because the settings of the pre-offset and post-offset are not critical.

Finally, to evaluate the speech quality for various number of users, the network delay distribution and the packet discard probability are obtained based on the following system parameters:

- The round-trip propagation delay is 15 us based on a network length of 1 Km (coaxial cable).
- The packet transmission time is 116.4 us based on a 10 Mb/s channel transmission rate and 64 bits overhead.
- The packet generation rate of each station is 36 packets/sec.
- Maximum allowable network delay (MAND) are 27.5 and 110 ms which correspond to one and four packet-lengths respectively.

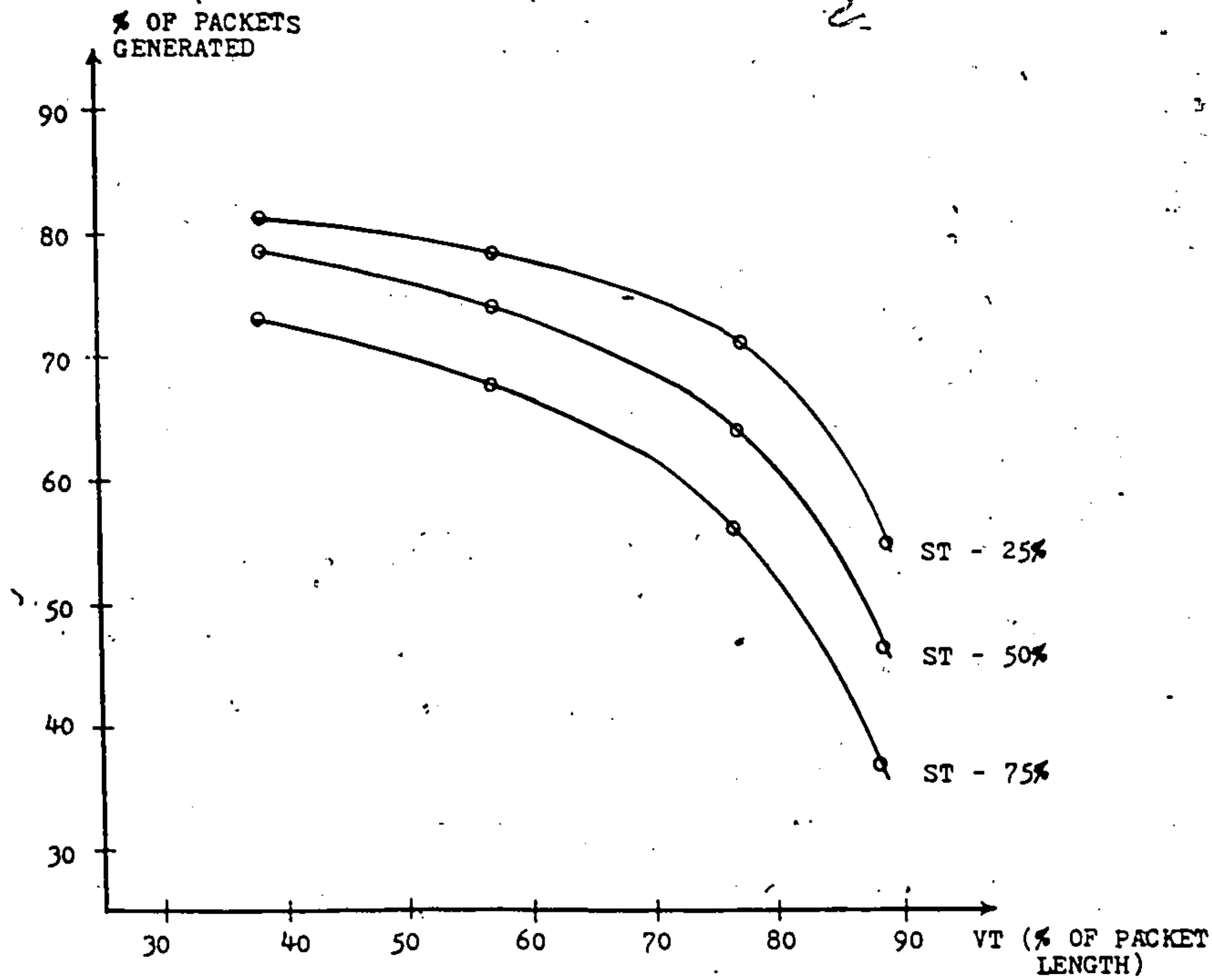


Figure 25: Packet Generation versus Silence Detection Thresholds

Figure 26 shows the packet discard probability as the number of users varies from 125 to 195. As can be seen, the HYMAP protocol shows a significant improvement, in terms of less packets discarded for a given number of users, when the MAND changes from 27.5 ms to 110 ms. When MAND is 110 ms, CSMA/CD shows improvement only when the number of users is between 153 and 168. As the number of users increases beyond 168, more packets are discarded than when MAND is 27.5 ms. This is because of the increase in MAND which allows more packets to be retransmitted and the throughput decreases as a result of this network traffic increase. HYMAP does not degrade with the increase in MAND because of the collision-free procedure which allows packets to be transmitted in an orderly manner. The typical shapes of the CSMA/CD and HYMAP network delay distributions are shown in Figure 27 for 145 users and MAND equals to 27.5 ms. Note that 85% of the packets have delay less than 1.72 ms in CSMA/CD and only 17% in HYMAP. The variance of the network delay in HYMAP is larger than the one in CSMA/CD. In a packet voice system with receive-end buffering, the variance does not have any effect on the speech quality. However, this is not true, which will shown later, for a system without receice-end buffering. All this information is input to the packet voice simulator for speech quality measurements.

Table 4 shows the number of users and the corresponding speech quality that can be achieved. It has to be pointed out that the table is by no means a representation of the results of a comprehensive subjective testing. It is suggested that the observation of this table be used as rough guidelines only. Two cases are considered, with and

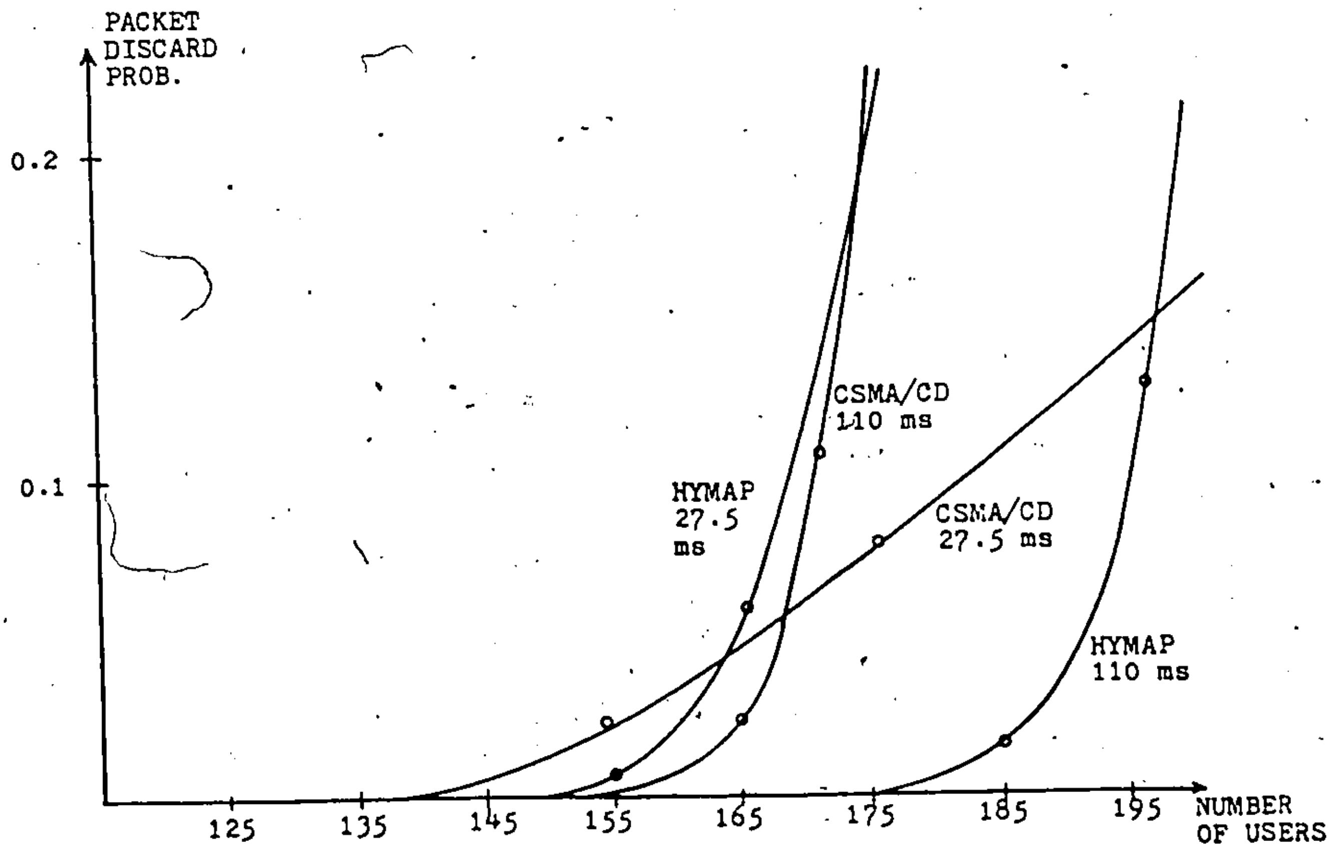


Figure 26: Packet Discard Probability for Various Numbers of Users

without the receive-end buffering protocol. If receive-end buffering is used, the buffer-delay is equal to MAND. Longer buffer-delay does not serve any purpose because the maximum network delay that a packet can have is MAND. If there is no receive-end buffering, packets are handled by the waiting for late packets protocol. The results show that the packet voice system performs much better with buffering than without it. The speech quality degradation

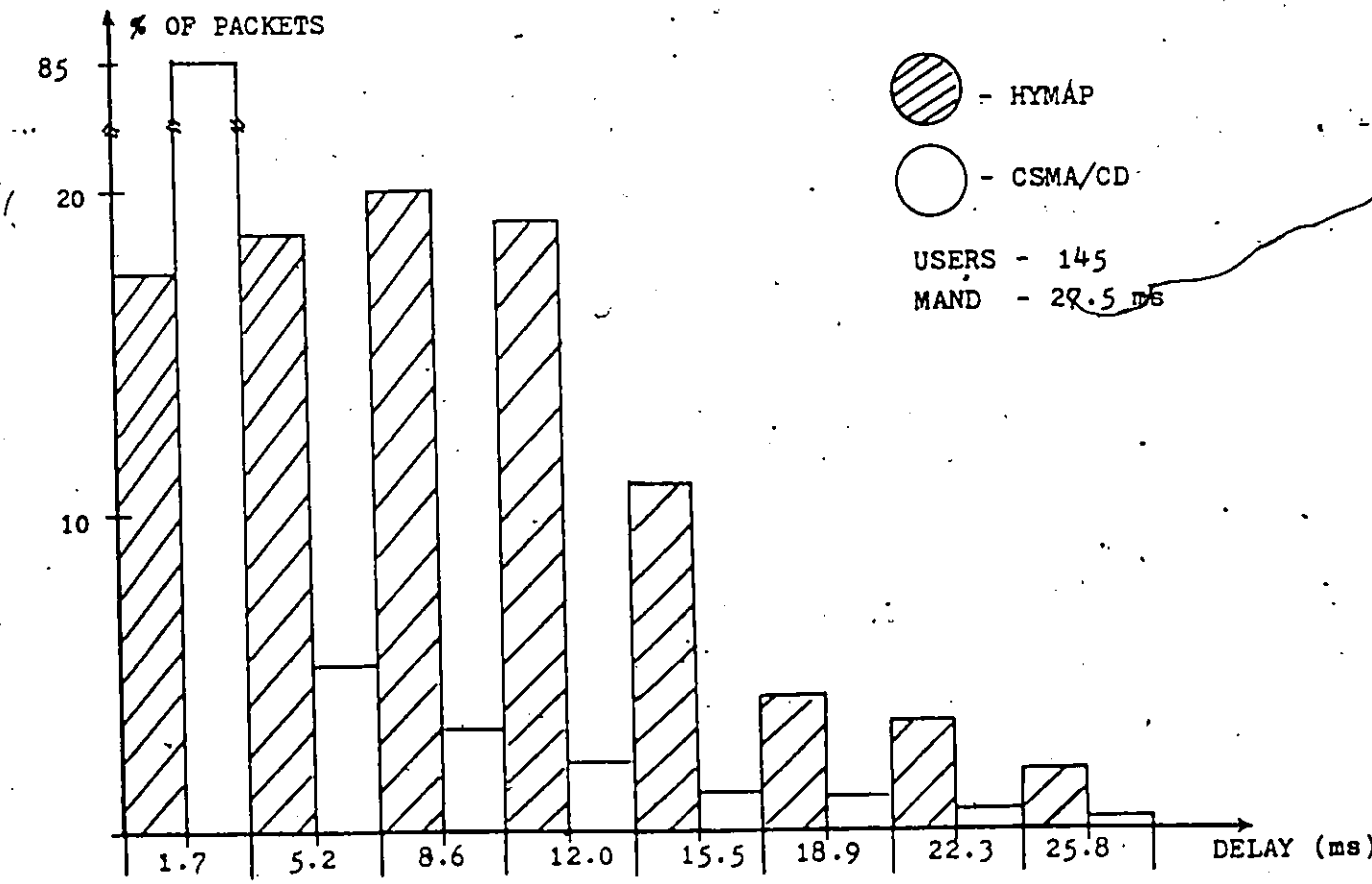


Figure 27: Network Delay Distribution

is very apparent for HYMAP when no buffering is used. This is due to the larger variance of the network delay distribution. More packets arrive late at the receiver than it is in CSMA/CD case. However, with buffering of four packets, HYMAP can support more users with "Good" speech quality.

The speech quality, rate "Fair" and "Poor", is quite sensitive to the number of users, especially when receive-end buffering is used. An increase of 7 users (in HYMAP with receive-end buffering and MAND equals to 110 ms) changes the speech quality from "Good" to "Unsatisfactory". So, to determine the maximum number of users for a given system with "Good" speech quality, the use of safety margin is necessary.

Generally speaking, with receive-end buffering, better speech quality can be achieved by increasing the MAND because more packets can be transmitted within the specified amount of time. However, this will introduce unnecessary delay. Due to the lower throughput limitation of CSMA/CD, the increase of MAND beyond a certain maximum will have a negative effect of decreasing the throughput. Certainly, MAND can not be increased without bound. Its contribution must keep the maximum end-to-end delay less than 300 ms. It should be noted that when receive-end buffering is used, the speech quality depends solely on the packet discard probability. The network delay has no effect at all.

TABLE 4

Speech Quality for Various Numbers of Users

# OF USERS	WITH BUFFERING				WITHOUT BUFFERING			
	CSMA/CD		HYMAP		CSMA/CD		HYMAP	
	27.5 ms	110 ms	27.5 ms	110 ms	27.5 ms	110 ms	27.5 ms	110 ms
125					GOOD	FAIR		POOR
135			GOOD				FAIR	
145	GOOD	GOOD			FAIR	POOR	POOR	
155				GOOD	POOR			
165	FAIR	FAIR	FAIR					UNSAT.
175		POOR	POOR				UNSAT.	
185	POOR	UNSAT.	UNSAT.		UNSAT.	UNSAT.		
195				FAIR				
				POOR				
	UNSAT.			UNSAT.				

Chapter V

AN INTEGRATED DATA AND VOICE STATION

The experimental results described in the last chapter demonstrate that packet transmission of voice is feasible on a broadcast type network. There is a trade-off between the speech quality and the number of users. In those experiments, voice occupied the whole bandwidth of the broadcast channel. If the voice traffic is limited to a certain amount, the excess channel bandwidth can be allocated for data transmission. The limitation of the voice traffic can be done by restricting the number of users and/or using an encoder with a lower coding rate.

The integration of data and voice onto the same channel introduces the problem of handling two different types of packets, data and voice. Voice communication must be done on a real-time basis and the network delay must be kept small so that less packets will be discarded by the transmitter, whereas in data communication, the delay of packet transmission is not that critical. Therefore, it is desirable from the system point of view to keep the voice packet delay short at the expense of longer data packet delay. This can be achieved by assigning a higher priority for the voice packets to transmit over the data packet [31].

In an integrated data and voice network environment, there exist three types of stations:

1. those that handle data only, e.g. a CRT terminal.
2. those that handle voice only, e.g. a digitized telephone set.
3. those that handle both data and voice.

The latter type, integrated data and voice stations (IDVS), are of concern in this chapter. IDVS has the advantage of sharing some of the low-level handware and software, such as data buffers, transmission hardware and transmission software, by the data and voice packets. The priority assignment of this type has to be done both locally and globally. The term "local" refers to a scheme within the station itself, whereas "global" refers to a scheme among all stations. In this chapter, the priority assignment scheme of an IDVS based on CSMA/CD protocol will be given. A block diagram of an IDVS will be given at the end of the chapter.

The status of an integrated data/voice station can be described by a 4-state diagram as shown in Figure 28 . They are pause (P), data (D), voice(V) and data/voice (D/V). The state diagram is simplified by excluding transitions between states D and V and also states P and D/V because these transitions seldom occur. The nodes and arrows correspond to the states and state transitions respectively. When a station is at rest, the station is at state P. When a talkspurt begins, the state V is reached, designating voice packets are being generated. A station is in state D/V if it has both data and voice packets to transmit. State D is reached if a station has only data packets to transmit. As will be shown later, global priority assignment is applied on states D, V and D/V. In addition to this, the state D/V also requires local priority assignment.

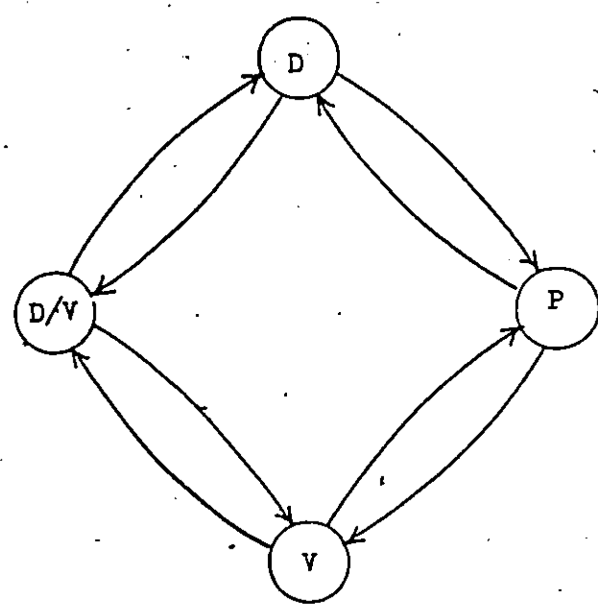


Figure 28: State Diagram of a IDVS

5.1 LOCAL PRIORITY ASSIGNMENT SCHEME (LPAS)

This scheme is applied only in state D/V, that is when there are two types of packets in the station. It allows a station to transmit its voice packets before its data packets. However, it does not guarantee the transmission of voice packets before other station's data packets. Such sequencing must be done by the global priority assignment scheme. In LPAS, two classes of priority are required. The high and low priorities are assigned to voice and data packets respectively. The high priority packets are always put in front of the low priority packets in the transmission queue so that voice packets are always transmitted before data packets. This can be achieved easily within a station because it knows what type of packets is being generated at any time.

5.2 GLOBAL PRIORITY ASSIGNMENT SCHEME (GPAS)

This scheme is applied to all stations when they have packets to transmit. The priority of packets is the same as the one in LPAS. It allows voice packets in all stations to transmit before all data packets. Since each station does not know what type of packets, data or voice, other stations are going to transmit, the best way to exchange this information is via the multi-access protocol. By doing so, the following requirements are necessary [31]:

1. Voice packets have higher priority to transmit than data packets.
2. Distributed control has to be conserved.
3. Packets of the same class should have equal right of accessing the channel.

4. The transmission of voice packets should be insensitive to the load put on the channel by data packets.
5. The scheme is robust in terms of operations.
6. The overhead to implement this scheme should be kept minimal.

Two methods are used for the ~~GPAS~~ [28][31], the Reservation Method and the Different Back-off Algorithms.

5.2.1 Reservation Method

The priority assignment of the two classes, data and voice, is based on the CSMA/CD protocol. The transmission of packets takes place one slot time after the end of a packet transmission. The information on the slot determines whose turn to transmit, high or low priority class. If a reservation message was present in the slot, the high priority class will transmit at the end of the slot, or else the low priority class. The reservation method is explained below:

1. Each station monitors activity on the channel at all times just as in ordinary CSMA scheme.
2. If the high priority class has packets to transmit, a reservation message (RM) of length L is sent at the end of a packet transmission (EOT) whether successful or not. (L is the shortest packet length detectable by all stations)

3. The reservation message will arrive at all stations within a slot time of 2D + L, where D is the end-to-end propagation delay time. At the end of the reservation slot, the high priority class proceeds packet transmission using CSMA/CD protocol.
4. The low priority class proceeds transmission, using CSMA/CD protocol, if there is no reservation message within the 2D + L slot.
5. For those packets which are formed after the reservation slot, they are transmitted based on the CSMA/CD protocol regardness of their priority class.

The reservation method is illustrated in Figure 29 .

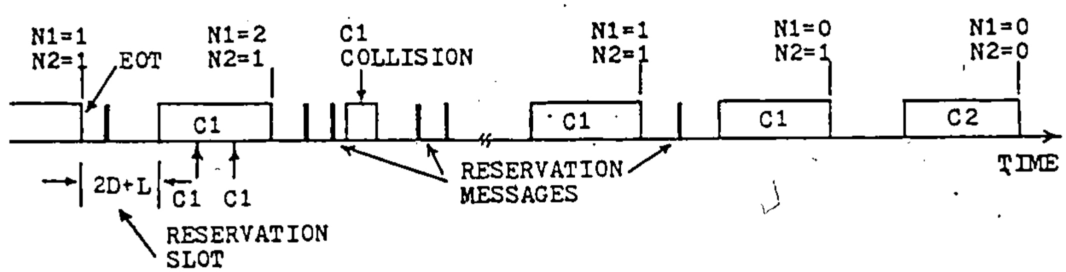


Figure 29: Illustration of Reservation Method

Let N1 and N2 denote the number of active users before **EOT** in high priority class (C1) and low priority class (C2) respectively. A vertical upward arrow represents the arrival of a new packet to the system. At the first EOT, N1=1 and N2=1. A reservation message is sent by the high priority class right after EOT. Note that the reservation message will arrive at all stations within the reservation slot, thus synchronization of all stations is not necessary. At the end of the reservation slot, the high priority class transmits its packet successfully. During the packet transmission, two new C1 packets arrive and reservation messages are sent, at the end of the packet transmission, to reserve the next frame. Unfortunately, collision of the two C1 packets occurs and they have to abort transmission and schedule for retransmissions. Again, the next frame is reserved by the high priority class by sending the reservation messages. All the stations repeat this protocol until all packets have been transmitted.

As the number of packets of the same priority class increases, more collisions will be likely to happen especially for data packets. To circumvent this problem, the P - persistent CSMA/CD protocol can be used instead of the ordinary CSMA/CD protocol. In the P - persistent CSMA/CD protocol [16], a ready station senses the channel and operates as follow:

1. If the channel is sensed idle then it transmits the packet.

2. If the channel is sensed busy, it waits until the channel becomes idle and then transmits with probability P or delays transmission by D with a probability 1-P. If the channel is still busy after the delay, the same process is repeated.

3. Collisions are handled the same way as in CSMA/CD.

As a matter of fact, the CSMA/CD protocol is a version of P - persistent CSMA/CD protocol with P equals to 1.

In terms of this reservation scheme, the overhead is minimal, only a reservation message is required. The scheme is robust because no exact information, just a reservation message to indicate the presence of the high priority class. However, the voice packets are not fully insensitive to the load increase due to data packets.

5.2.2 Different Back-off Algorithms (DBA)

In terms of implementation, the reservation method proposed in the last section requires many modifications to the CSMA/CD protocol. The DBA requires only the addition of a back-off algorithm. It is still based on the CSMA/CD protocol but two back-off algorithms are used instead of one. One is for the voice packets and the other one is for the data packets. The back-off algorithm for the voice packets is made to have a shorter average delay than the one for the data packets. For example, consider the binary-exponential back-off algorithm used in Ethernet, the retransmission delay is taken from the interval indicated below:

$$\left[0\ ,\ 2^{n}-1\right] D$$

where n is the number of collisions and D is the channel propagation delay. In DBA method, the above interval is used for the voice packet retransmission delay and the following modified version can be used for the data packets:

$$\left[0\ ,\ 2^{n+k}-1\right] D$$

where k is a positive number which is used to increase the upper bound of the delay interval. By doing so, the voice packets will have a higher probability of channel access than data packets. One drawback of this method is that the full partition of the high and low priority packets can not be achieved. A small number of the data packets may have a higher priority than voice packets.

5.3 BLOCK DIAGRAM OF AN INTEGRATED DATA AND VOICE STATION

In this section, a block diagram of an integrated data and voice station is given. It is based on the CSMA/CD protocol with the reservation method. A display terminal, (CRT), and a speech terminal (telephone set) are the sources of this station. Packets are transmitted through a common access point to the broadcast channel. A block diagram of an IDVS is shown in Figure 30 .

The driver consists of electronics to power the transmit signal down the broadcast channel. Upon reception, it is capable of eliminating the noise imposed on the received signal. Furthermore, encoding such as Manchester is done on the transmit signal so that the clock signal can be recovered at the receiver. Also, it could provide ground isolation between the channel and the station. The multi-access protocol with reservation scheme is composed of Collision-Detection, Back-off Algorithm, Carrier Detector and Voice/Data Delay blocks. One of the well known back-off algorithms is the binary-exponential used in Ethernet. Examples of others are linear incremental and random table [33]. The Carrier Detector detects the reservation message and the end of a packet transmission. If a reservation message is detected, the voice packet will have the access right to the channel at the end of the reservation slot. Data packets will have the access right if no reservation message is detected.

All packets are passed through a data-link control protocol before being transmitted. This is done by the Data Link Controller. The main function of the data-link control protocol is to allow communication among stations in a

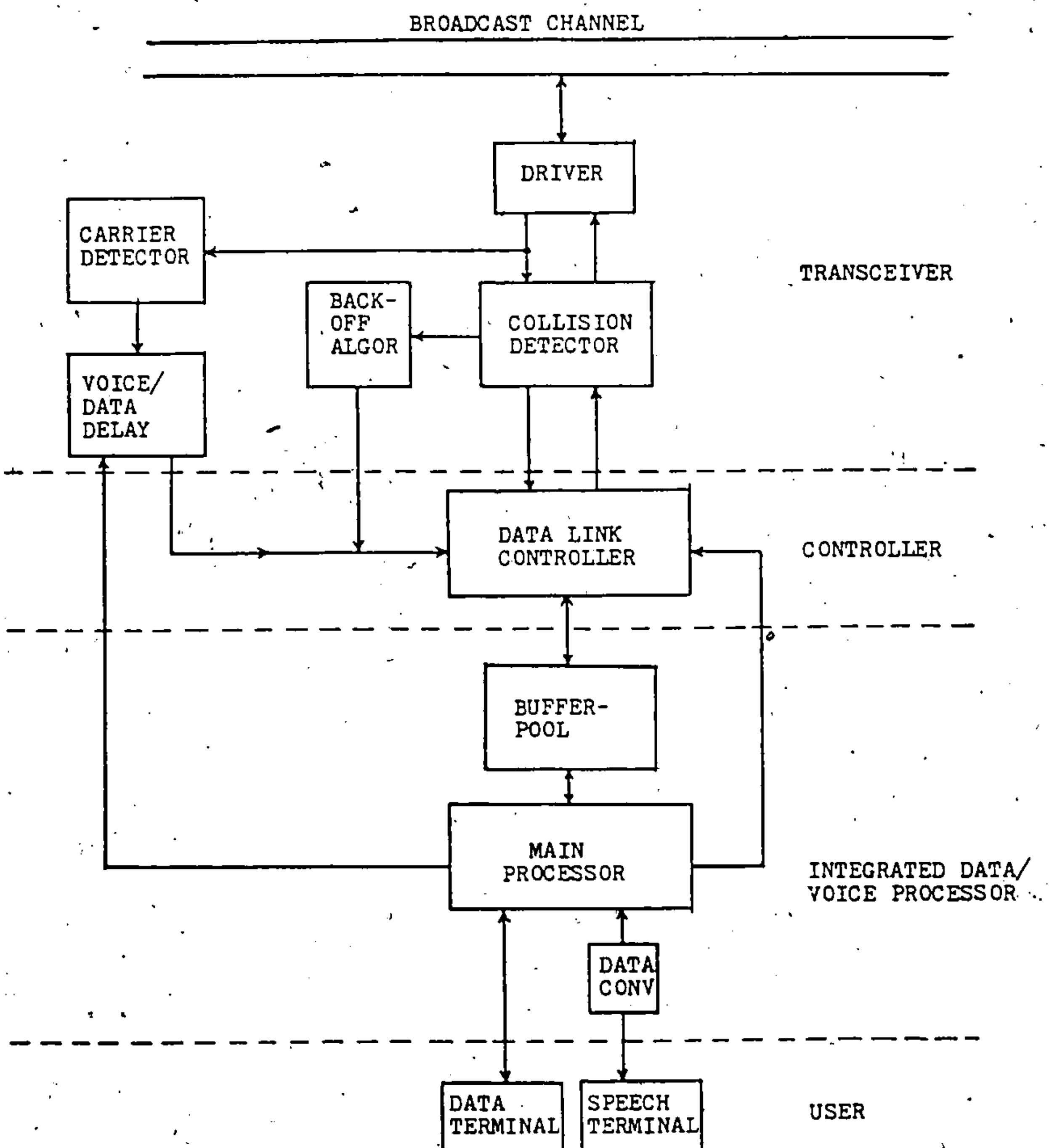


Figure 30: Block Diagram of an IDVS

reliable way. In general, the frame format of the protocol consists of the opening and closing frame indicators, destination and source addresses, control information and error checking sequence [1].

The last main block is the Integrated Data/Voice Processor. It consists of the Buffer-pool, Main Processor and Data Converter blocks. The functions such as voice/data packet assembly and disassembly, voice and silence detection, receive-end protocol and silence recovery are performed by the main processor. A common buffer-pool is used for the storage of both the voice and data information. The data converter of the speech terminal does the digital-to-analog conversion and vice versa.

All the functional blocks which were shown in Figure 30 are common to both voice and data. However, their data-link control protocols are different. As mentioned earlier, bit errors are intolerable in data communications but small errors are allowed in voice communication. Therefore, the voice data does not have to be protected against errors. The frame formats of the data-link control protocols for data [33] and a modified version for voice are shown in Figure 31.

In the data frame format, the CRC (Cyclic Redundancy Checkwords) checks the error starting from the beginning of the destination address to the end of the data field. If

OPENING FLAG	DESTINATION ADDRESS	SOURCE ADDRESS	CONTROL FIELD	DATA FIELD	CRC	CLOSING FLAG

(a) DATA

OPENING FLAG	DESTINATION ADDRESS	SOURCE ADDRESS	CONTROL FIELD	CRC	DATA FIELD	CLOSING FLAG

(b) VOICE

Figure 31: Frame Format of Data Link Control Protocol

there is an error in the frame, the sender will retransmit it again after a time out or by receiving a negative acknowledgement. However, in the voice frame format, there is no need to protect the voice data from bit error. The error protection applies only on the destination address, source address and control fields. This is to guarantee that the frame is received by the correct station and the source address and control fields are valid for further communication. Baseband transmission is used and it is believed that the error rate is very small. Therefore, if the whole frame has to be retransmitted in case of error, the delay caused by the retransmission will degrade the speech quality more than the error does. So, in voice communication, retransmission of voice packets with error is not recommended.

Chapter VI

CONCLUSIONS

A real-time packet-voice simulator based on an 8086 CPU was designed to study the speech quality on a broadcast bus type local area network. It was assumed that the telephone set is a four-wire system so that the network is free of echo. It was not necessary to simulate many stations in a real-time basis. Instead, one station was simulated and the network delay distribution as well as the packet discard probability, which characterize the network, serve as the inputs to the simulator. The two inputs are obtained from a separate simulation program.

The simulation process begins by band-limiting the analog input signal at 3200 Hz and then digitizing it at 40 Kb/s using a CVSD encoder. Speech-silence, which is detected by a silence detector, is not transmitted. The silence detection thresholds, VT and ST, are set to 75% and 55% of the packet length respectively. The packet sequence number is used to recover the silence period at the receiver. The voice digital data is assembled into packets of length 27.5 ms which go through a process of delaying and discarding before being transmitted to the receiver. The packetization-freezed protocol is used to resolve the collisions caused by the synchronization of packet generation between two or more stations. Packets that are not transmitted within the maximum allowable network delay, 27.5 ms or 110 ms, will be discarded. Upon receiving the packets, the receive-end buffering protocol is used so that a smooth speech delivery can be obtained. At the output, the CVSD decoder converts the digital signal to analog which is then low-pass filtered at 3000 Hz to remove quantization noise before being delivered to the listener.

An informal listening test was carried out to evaluate the speech quality subjectively. The results showed the number of users that can be supported for a given speech quality. For example, "Good" speech quality allows 165 users for CSMA/CD and 190 users for HYMAP if receive-end buffering of 110 ms is used (for a codec rate of 40 Kb/s and a channel transmission rate of 10 Mb/s).

The results also showed that a significant improvement of speech quality, for a given number of users, can be achieved by using HYMAP protocol with receive-end buffering and MAND equals to 110 ms. The number of users can be increased substantially by taking advantage of the maximum end-to-end delay allowed in an echo-free local area network environment. However, due to the lower throughput limitation of CSMA/CD protocol, the increase in MAND will not benefit the system significantly.

The results obtained in this thesis can be used in the design of a packet-voice system. However, more research must be done on the protocol layers above the link level [1]. For example, two features are the conference call in the application layer and call setup procedure in the session layer. Flow control can be achieved by blocking the call setup when the network load has reached a certain level.

Lastly, the block diagram of an integrated data and voice station was given. Simulations should be done to validate the priority assignment schemes and to partition the amount of data and voice traffic that are best suited in this environment.

The contributions of this thesis are:

1. The design of a real-time packet-voice simulator.
2. Packet-voice system-design parameters were obtained, such as voice and silence thresholds which were set to be 75% and 55%, respectively, of the packet-length. The size of a packet is 27.5 ms.
3. To limit the network delay, a protocol of discarding packets which have not been transmitted within the maximum allowable network delay (MAND) is proposed.
4. A packetization-freezed protocol is proposed to resolve successive collisions due to possible synchronization of packet generation among stations.

REFERENCES

1. Tanenbaum A., " Computer Networks ", Prentice-Hall, 1981.
2. Shoch J.F.," Carrying Voice Traffic Throught an Ethernet Local Network -- A General Overview", Xerox Corporation, Aug., 1980.
3. Bially T., McLaughlin A.J.," Voice Communication in Integrated Digital Voice and Data Networks ", IEEE Trans. Comm., Sept. 1980, pp.1478-1489.
4. Coviello G.J., " Comparative Discussion of Circuit-Vs-Packet Switched Voice ", IEEE Trans. Comm., Aug. 1979, pp.1153-1160.
5. Gitman I., " Economic Analysis of Integrated Voice and Data Networks ', Proc. IEEE, Nov. 1978, pp.1549-1570.
6. Gold B., " Digital Speech Network ", Proc. IEEE, Dec. 1977, pp.1636-1658.
7. Forgie J.W., " Speech Transmission in Packet-Switched Store-and-Forward Networks ", IEEE National Computer Conference, 1975, pp.137-142.
8. Minoli D., " Issues in Packet Voice Communication ", Proc. IEEE, Aug. 1979, pp.729-740.

9. Bially T., Gold b., Seneff S., " A Technique for Adaptive Voice Flow Control in Integrated Packet Networks ", IEEE Trans. Comm., Mar. 1980, pp.325-333.

10. Chakravarthy C.V., Schilling C.L., " Design of a Packet Voice Transmission System ", Nat. Telecommunications Conf. 1979, pp.13.1.1-13.1.5.

11. Melvin D.K., " Voice on Ethernet ", Intel Corporation, Nov 1981.

12. Tobagi F.A., Gonzalez-Cawley N., " On CSMA-CD Local Networks and Voice Communication ", IEEE INFOCOM 1982, pp.122-127.

13. Flanagan J.L., " Speech Coding ", IEEE Trans. Comm., Apr. 1979, pp.710-737.

14. Johnson D.H., O'leary G.L., " A Local Access Network for Packetized Digital Voice Communication ", IEEE Trans. Comm., May 1981, pp.679-688.

15. Rios M., " HYMAP, A New Protocol For Local Area Networks ", Master of Applied Science Thesis, University of Ottawa, 1983.

16. Tobagi F.A., " Multi-Access Protocols in Packet Communications Systems ", IEEE Trans. Comm., Apr. 1980, pp.468.

17. Karp H.R., " Practical Applications of Data Communication ", Elect. Mag., 1980, pp.388-403.

18. Brady P.T., " A Statistical Analysis of ON-OFF Pattern in 16 Conversations ", BSTJ, Jan. 1968, pp.73-91.

19. Standard Publication, " IEEE Recommended Practice for Speech Quality Measurement", IEEE Trans. Aud. and Elect., Sept. 1969, pp.225-246.

20. Coviello G.J., Lake O.L., " System Design Implications of Paçketized Voice ", International Computer Conference, 1977, pp.38.3-49 to 53.

21. Klemmer E.T., " Subjective Evaluation of Transmission Delay in Telephone Conversations ", BSTJ, 1967, pp.1141-1147.

22. Bellamy J., " Digital Telephony ", John Wiley & Sons, 1982.

23. Brady P.T., " A Technique for Investigating ON-OFF Patterns of Speech ", BSTJ, Jan. 1965, pp.1-12.

24. Linear Interface Integrated Circuits, Motorola, 1979.

25. MCS-86 System Design Kit User's Guide, Intel Corporation, 1980.

26. The 8086 Family User's Manual, Intel Corporation, 1980.

27. Gross D., Harris C.M., " Fundamental of Queueing Theory ", Wiley, 1974.

28. Gruber J., " Performance Considerations for Integrated Voice and Data Networks ", Computer Communications, Vol.4 No.5, Jun. 1980, pp.106-126.

29. Jayant N.S., Christensen S.W., " Effects of Packet Losses in Waveform Coded Speech and Improvements Due to an Odd-Even Sample-Interpolation Procedure", IEEE Trans. Comm., Feb. 1981, pp.101-109.

30. Ingalls D.H.," The Smalltalk Graphics Kernel ", Byte, Aug. 1981, pp.168-194.

31. Tobage F.A., " Carrier Sense Multiple Access with Message-Based Priority Functions ", IEEE Trans. Comm., Jan. 1982, pp.185-200.

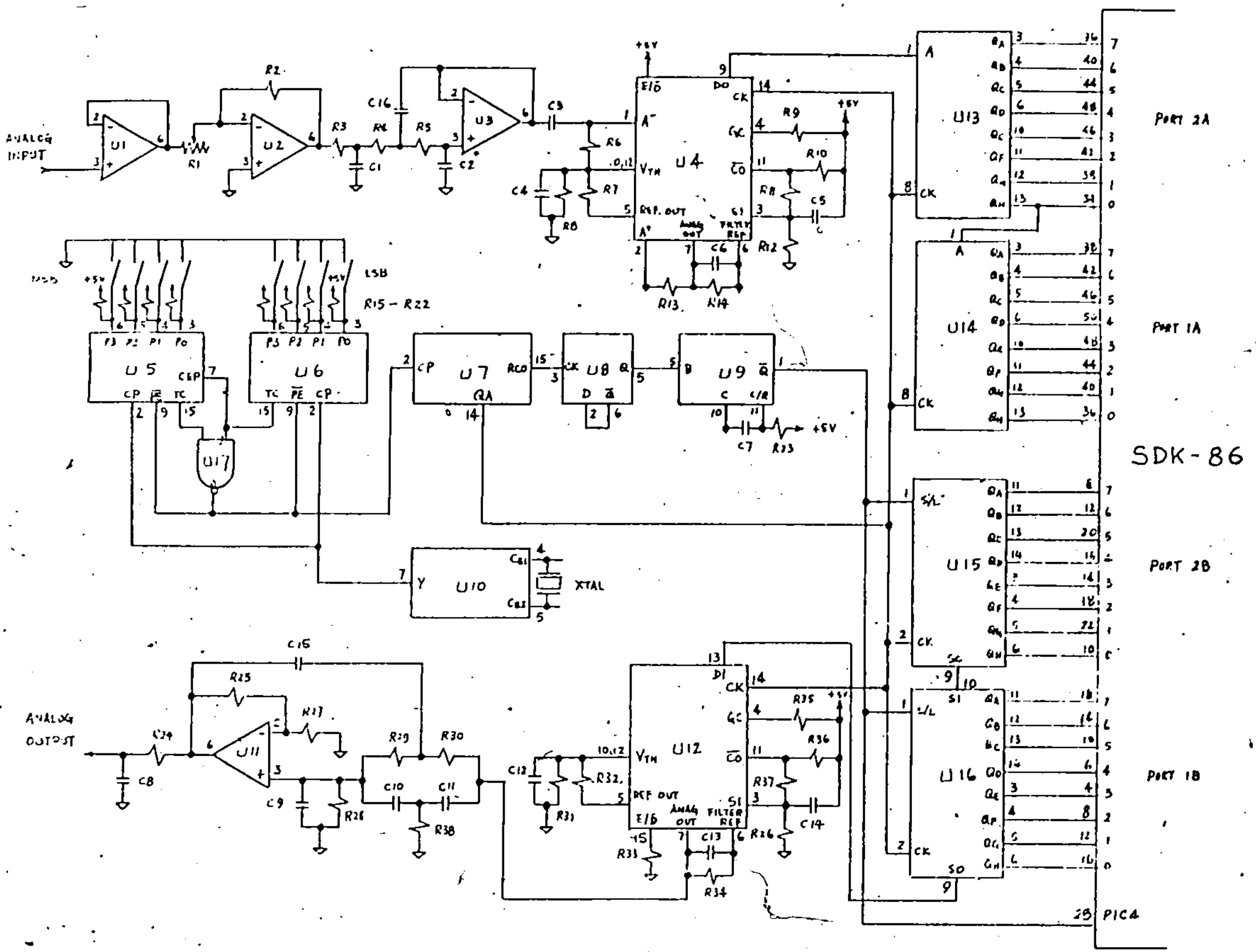
32. Shoch J.F., " Evolution of the Ethernet Local Computer Network", Local Networks for Computer Communications, North-Holland pub., 1981, pp.157.

33. Davis G.R., " The Local Network Handbook ", Data Communications Magazine, Mc-Graw Hill, 1982, pp.20-30.

Appendix A

SCHEMATIC DIAGRAM OF THE SUPPORT HARDWARE

The following two pages show the schematic diagram and the part list of the support hardware. U3 and U11 are the input and output low-pass filters respectively. Analog-to-digital conversion is done by U4 whose output is converted to parallel by U13 and U14. The processed digital output is converted to serial, by U15 and U16, which is then converted to analog signal by U12. The clock signal is generated by U5-U10.



SDK-86
PORT 2A
PORT 1A
PORT 2B
PORT 1B
ANALOG INPUT
ANALOG OUTPUT
XTAL
LSB
R15 - R22
U1
U2
U3
U4
U5
U6
U7
U8
U9
U10
U11
U12
U13
U14
U15
U16
U17

IC PART LIST

NUMBER	TYPE	+5V	GND	HIGH	LOW
U1,U2,U3,U11	741 (+12V-7, -12V-4)				
U4,U12	MC3417	16	8		
U5	74163	16	8	1,10	
U6	74163	16	8	1,7,10	
U8	7474	14	7	1,4	
U9	74121	14	7		3,4
U10	74S124	16,15	8,9		6
U13,U14	74164	16	8	2,9	
U15,U16	74165	16	8		10,15
U7	74163	16	8	1,3,4,5,6,7,9,10	

NOTE: ALL PINS CONNECT TO "HIGH" ARE VIA 2.2K RESISTORS

RESISTOR PART LIST

NUMBER	VALUE
R1,R7,R14,R23, R32,R33,R34	10 K
R3,R4,R5	6.8 K
R6,R13	600
R8,R25,R31	1 K
R9,R35	1.3 K
R10,R36	0.3 K
R11,R37	100 K
R12,R26	5 M
R15-R22	2.2 K
R30	87.6 K
R29	175 K
R38	247 K
R27	600 K
R28	1.11 M
R24	212 K

CAPACITOR PART LIST

NUMBER	VALUE
C1	7.2 nf
C2	2.2 nf
C3	4 µf
C4,C6,C12,C13	0.1 µf
C5,C14	0.05 µf
C7	200 pf
C8	380 pf
C9	220 pf
C10	78 pf
C11	157 pf
C15	1 nf
C16	9.3 nf

XTAL --- 6.144 MHz

Appendix B

SIMULATOR PROGRAM LISTING

The simulation program consists of five modules. They are initialization, computation, interrupt service, packet handling and data. There are about 700 lines of program instructions which occupy a memory space of 2.2 K bytes. The data module occupies another 2 K bytes of memory. These modules are assembled by a cross-assembler in a MDS-800 microcomputer development system. The program code, after linking, relocation and object-hex code conversion, is down loaded into the SDK-86 for execution.

ISIS-II MCS-86 MACRO ASSEMBLER V2.1 ASSEMBLY OF MODULE M1
OBJECT MODULE PLACED IN :F1:M1.OBJ
ASSEMBLER INVOKED BY: ASM86 :F1:M1.SRC NOMP EP DA(29-07-83)

```
LOC  OBJ          SOURCE

                  ;
                  ;***************************************
                  ;
                  ;         INITIALIZATION MODULE
                  ;
                  ;***************************************
                  ;
 FFF8             INPUT   EQU     0FFF8H  ;F2A-P1A         ;INPUT PORT ADDRESS
 FFFA             OUTPUT  EQU     0FFFAH  ;F2B-P1B         ;OUTPUT PORT ADDRESS
 FFFE             CONTROL EQU     0FFFEH                   ;CONTROL PORT ADDRESS
                  ;
 0089             CWORD   EQU     089H                     ;SET PORT A TO INPUT AND B TO OUTPUT
 0009             ENINT   EQU     09H                      ;ENABLE INTERRUPT
 0008             DIINT   EQU     08H                      ;DISABLE INTERRUPT
 00FF             MFLAG   EQU     0FFH
 AAAA             SQUARE  EQU     0AAAAH                   ;SILENCE DATA
                  ;
 FFF2             USARTST EQU     0FFF2H                   ;USART STATUS
 FFF0             USARTDT EQU     0FFF0H                   ;USART DATA
 0002             RXRDY   EQU     02H                      ;USART RX DATA READY FLAG
                  ;
 5555             SILENCE1        EQU         5555H  ;SILENCE PATTERN 1
 AAAA             SILENCE2        EQU         0AAAAH ;SILENCE PATTERN 2
 FFFF             LOSTHD          EQU         0FFFFH ;LOST PACKET HEADER
 00FF             BUFHD           EQU         0FFH   ;PACKET HEADER
 0003             SAREA           EQU         3H     ;PACKET DATA AREA
 000A             DEEPPROB        EQU         10     ;SIZE OF PDD & PND QUEUES
 0010             DEEPPTR         EQU         16     ;SIZE OF BUFFER-POOL, TX AND RX QUEUES
                  ;
 ----             STOR    STRUC
0000              TOPPTR  DW      ?    ;TOP POINTER OF QUEUE
0002              ENDPTR  DW      ?    ;BOTTOM POINTER OF QUEUE
0004              PUSHPTR DW      ?    ;PUSH POINTER OF QUEUE
0006              POPPTR  DW      ?    ;POP POINTER OF QUEUE
0008              BUFFER  DW      ?    ;START OF QUEUE
 ----             STOR    ENDS
                  ;
 ----             ISRSK   STRUC
0000              ISRBP   DW      ?    ;POINTER OF OUTPUT PACKET
0002              ISRDI   DW      ?    ;SIZE OF OUTPUT PACKET
0004              ISRSI   DW      ?    ;SIZE OF INPUT PACKET
0006              ISRBX   DW      ?    ;POINTER OF INPUT PACKET
 ----             ISRSK   ENDS
                  ;
                          PUBLIC  STK2,BEGIN
                  ;
                  ;
                  ;EMEG IS THE SAME AS PKHAND FLAG
   #              FLAG    RECORD  TPS:1,BM:1,TEMP2:1,TEMP3:1,RRS:1,EMEG:1,
                  &               SILMODE:1,DELOVER:1,TEMP1:1
```

```
LOC  OBJ              SOURCE

                      ;
----                  P6      SEGMENT AT 100H
                              ASSUME  DS:P6,ES:P6
                      ;
                              EXTRN   PROB:WORD,PROBCOUNT:WORD,DELAY:WORD,DELAYCOUNT:WORD,
                      &               HIBYTE:WORD,LOWBYTE:WORD,PERTHI:WORD,PERTLOW:WORD,
                      &               STACKPTR:WORD,OUTREG:WORD,INSER:WORD,
                      &               MINPROB:WORD,TOTPKTX:WORD,TX:WORD,TXTOP:WORD,TXEND:WORD,
                      &               BUF:WORD,BUFFEREND:WORD,BUFFERAREA:WORD,
                      &               HI:WORD,LOW:WORD,PACKSIZE:WORD,MINDELAY:WORD,
                      &               MAXTSD:WORD,MINTSD:WORD,DUMPPTR:WORD,TXTEMPPTR:WORD,
                      &               MAXSIL:WORD,MINSIL:WORD,COUNTER:WORD,MAXSZ:WORD
                      ;
----                  P6      ENDS
                      ;
                              EXTRN   POPFIFO:FAR,PUSHFIFO:FAR,REPP:FAR,ISP:DWORD,D0:WORD,
                      &               COMPUTE:FAR
                      ;
                      ;SET UP NON-MASKABLE INTERRUPT VECTOR ADDRESS
                      ;
----                  VECTOR  SEGMENT AT 0H
0008                          ORG     8H      ;NMI POINTER
0008                  NMIPTR  LABEL   WORD
                      ;
                      ;SET UP TYPE "44" VECTOR ADDRESS AT LOCATION 110H
                      ;
0110                          ORG     44H*4H  ;TYPE44 VECTOR
0110                  TYPE44  LABEL   WORD
                      ;
                      ;SET UP USER STACK
                      ;
0180                          ORG     180H
0180 (16                      DW      16 DUP(?)
     ????
     )
01A0                  STK2    LABEL   WORD
                      ;
                      ;SET UP SYSTEM STACK
                      ;
0150                          ORG     150H    ;STACK AREA
0150 (24                      DW      24 DUP(?)
     ????
     )
0180                  TOS     LABEL   WORD
                      ;
                      ;NON-MASKABLE INTERRUPT SERVICE ROUTINE
                      ;
01C0                          ORG     1C0H    ;NMI ROUTINE
01C0                  BREAK   LABEL   WORD
01C0 50                       PUSH    AX
01C1 52                       PUSH    DX
01C2 BAFAFF                   MOV     DX,OUTPUT       ;SET OUTPUT TO SILENCE
01C5 B8AAAA                   MOV     AX,SQUARE
01C8 EF                       OUT     DX,AX
01C9 BAFEFF                   MOV     DX,CONTROL      ;DISABLE INTERRUPT
```

```
LOC  OBJ                       SOURCE

01CC B008                              MOV     AL,DIINT
01CE 8AE0                              MOV     AH,AL
01D0 EF                                OUT     DX,AX
01D1 5A                                POP     DX
01D2 58                                POP     AX
                               ;
01D3 CC                                INT     3H              ;BREAK-POINT INTERRUPT
01D4 CF                                IRET
                               ;
----                           VECTOR  ENDS
                               ;
                               ;THIS ROUTINE SETS UP ALL SEGMENT ADDRESSES, VECTOR POINTERS
                               ;PORT A&B INITIALIZATION AND STACK POINTERS
                               ;
----                           P0      SEGMENT AT 200H
                               ;
                               ASSUME  CS:P0
                               ;
0000 2BC0                      BEGIN:  SUB     AX,AX           ;SET UP NMI AND TYPE44 POINTER
0002 8EC0                              MOV     ES,AX
0004 B8C001                            MOV     AX,OFFSET BREAK
0007 26A30800                          MOV     ES:NMIPTR,AX
000B 26C7060A000000                    MOV     ES:NMIPTR+2,VECTOR
0012 B80000            E               MOV     AX,OFFSET ISR
0015 26A31001                          MOV     ES:TYPE44,AX
0019 26C7061201----    E               MOV     ES:TYPE44+2,SEG ISR
                               ;
0020 B80001                            MOV     AX,SEG BUFFERAREA
0023 8ED8                              MOV     DS,AX
0025 8EC0                              MOV     ES,AX
0027 B80030                            MOV     AX,VECTOR
002A 8ED0                              MOV     SS,AX
002C BC0001                            MOV     SP,OFFSET TOS
                               ;
002F BAFEFF                            MOV     DX,CONTROL
0032 B089                              MOV     AL,CWORD
0034 8AE0                              MOV     AH,AL
0036 EF                                OUT     DX,AX
                               ;
                               ;VARIABLE INITIALIZATONS START HERE
                               ;
0037 FC                                CLD                     ;AUTO-INCREMENT
0038 B90000                            MOV     CX,0H           ;RESET ALL FLAGS
003B 81C90400                          OR      CX,MASK SILMODE ;SET SILENCE MODE
003F 51                                PUSH    CX
0040 C70600000001      E               MOV     STACKPTR,OFFSET STK2    ;SET UP USER STACK
0046 C70600005555      E               MOV     OUTREG,SILENCE1
004C C70600000000      E               MOV     INSER,0H
                               ;
                               ;CLEAR STORAGE IN DATA MODULE FROM VARIABLE "TOTPKTX" TO THE END
                               ;OF PACKET STORAGE AREA
                               ;
0052 BB0000            E               MOV     BX,OFFSET TOTPKTX
0055 BA0000            E               MOV     DX,OFFSET BUFFEREND
0058 42                                INC     DX
```

```
LOC  OBJ                  SOURCE

0059 C60700               H0:     MOV     BYTE PTR[BX],0H
005C 43                           INC     BX
005D 3BD3                         CMP     DX,BX
005F 75F8                         JNE     H0
                          ;
                          ;INIT MAX & MIN COUNTERS TO KEEP MAX & MIN OF VARIABLES
                          ;
0061 B8FFFF                       MOV     AX,0FFFFH
0064 A30000          E            MOV     MINPROB,AX
0067 A30000          E            MOV     MINDELAY,AX
006A A30000          E            MOV     MINTSD,AX
006D A30000          E            MOV     MINSIL,AX
                          ;
0070 B80000                       MOV     AX,0H
0073 A30000          E            MOV     MAXTSD,AX
0076 A30000          E            MOV     MAXSIL,AX
                          ;
                          ;CLEAR DELAY ELEMENT COUNT LOCATIONS
                          ;THE "DELAY ELEMENT COUNT" IS USED TO COUNT THE NUMBER OF TIMES
                          ;THE CORRESPONDING DELAY ELEMENT IS USED
                          ;
0079 8B0E0000        E            MOV     CX,COUNTER
007D 8CDA                         MOV     DX,DS           ;SAVE DATA SEGMENT
007F B8----          E            MOV     AX,SEG D0
0082 8ED8                         MOV     DS,AX
0084 BB0000          E            MOV     BX,OFFSET D0
0087 83C306                       ADD     BX,6H
008A C7070000             H1:     MOV     WORD PTR [BX],0H
008E 83C308                       ADD     BX,8H
0091 E2F7                         LOOP    H1
0093 8EDA                         MOV     DS,DX           ;RESTORE DATA SEGMENT
                          ;
                          ;SETUP UP ALL FIFO QUEUES - TOP,END,PUSH,POP PTRS
                          ;THE QUEUES ARE BUFFER-POOL, TRANSMIT, RECEIVE, PACKET DISCARD DECISION
                          ;AND RANDOM NETWORK DELAY
0095 B90300                       MOV     CX,3H
0098 BB0000          E            MOV     BX,OFFSET TX
009B B80000          E            MOV     AX,OFFSET TXTOP
009E BA0000          E            MOV     DX,OFFSET TXEND
00A1 BE2C00                       MOV     SI,12+2*DEEPPTR
00A4 E86800                       CALL    SETUP
00A7 83EA0C                       SUB     DX,(DEEPPTR-DEEPPROB)*2
00AA B90200                       MOV     CX,2H
00AD BE2000                       MOV     SI,12+2*DEEPPROB
00B0 E85400                       CALL    SETUP
                          ;
                          ;PUT PTRS IN BUFFER-POOL QUEUE, LEAVE 5 BYTES (0) BETWEEN PACKETS
                          ;
00B3 B91000                       MOV     CX,DEEPPTR
00B6 BA0000          E            MOV     DX,OFFSET BUFFERAREA
00B9 8B1E0000        E            MOV     BX,MAXSZ
00BD 83C305                       ADD     BX,5H
                          ;
00C0 B80000          E    I1:     MOV     AX,OFFSET BUF
00C3 9A0000----      E            CALL    PUSHFIFO
```

```
LOC  OBJ                    SOURCE

00C8 03D3                           ADD     DX,BX
00CA E2F4                           LOOP    I1
                            ;
                            ;SET UP INITIAL PACKET POINTERS AND SIZES IN USER STACK
                            ;
00CC BE0300                         MOV     SI,SAREA
00CF 8BFE                           MOV     DI,SI
00D1 B80000            E            MOV     AX,OFFSET BUF
00D4 9A0000----        E            CALL    POPFIFO
00D9 8EDA                           MOV     DS,DX
00DB BDFFFF                         MOV     BP,LOSTHD
                            ;
00DE C60700                         MOV     BYTE PTR [BX],0H
00E1 87260000          E            XCHG    STACKPTR,SP
00E5 53                             PUSH    BX
00E6 56                             PUSH    SI
00E7 57                             PUSH    DI
00E8 55                             PUSH    BP
00E9 87260000          E            XCHG    STACKPTR,SP
                            ;
                            ;FILL UP PDD AND PND QUEUES
                            ;
00ED B90000                         MOV     CX,DEEFPROB
00F0 9A0000----        E    I2:     CALL    COMPUTE
00F5 E2F9                           LOOP    I2
                            ;
                            ;INITIALIZE INTERRUPT
                            ;
00F7 59                             POP     CX
00F8 BAFEFF                         MOV     DX,CONTROL
00FB B80900                         MOV     AX,ENINT
00FE 8AE0                           MOV     AH,AL
0100 EF                             OUT     DX,AX
0101 FB                             STI
                            ;
0102 EA0000----        E            JMP     REPR     ;GO TO COMPUTATION ROUTINE
                            ;
                            ;THIS SUBROUTINE IS INVOKED FOR QUEUE INITIALIZATION
                            ;CX - # OF QUEUES
                            ;BX - ADDRESS OF FIRST QUEUE
                            ;AX - TOP OF QUEUE
                            ;DX - BOTTOM OF QUEUE
                            ;SI - SIZE OF QUEUE
                            ;
0107                        SETUP   PROC
0107 8907                   I0:     MOV     [BX],AX
0109 895702                         MOV     [BX+2],DX
010C 894704                         MOV     [BX+4],AX
010F 894706                         MOV     [BX+6],AX
0112 C747080000                     MOV     WORD PTR[BX+8],0H
0117 87DA                           XCHG    BX,DX
0119 C7070000                       MOV     WORD PTR[BX],0H
011D 87DA                           XCHG    BX,DX
011F 03DE                           ADD     BX,SI
0121 03C6                           ADD     AX,SI
```

```
LOC  OBJ                LINE     SOURCE

0123 03D6                267              ADD     DX,SI
0125 E2E0                268              LOOP    I0
0127 C3                  269              RET
                         270     SETUP    ENDP
                         271     ;
----                     272     P0       ENDS
                         273     ;
 0000                    274              END BEGIN

ASSEMBLY COMPLETE, NO ERRORS FOUND
```

```
ISIS-II MCS-86 MACRO ASSEMBLER V2.1 ASSEMBLY OF MODULE M2
OBJECT MODULE PLACED IN :F1:M2.OBJ
ASSEMBLER INVOKED BY:  ASM86 :F1:M2.SRC NOMR EP DA(29-07-83)


LOC  OBJ                 LINE     SOURCE

                           1      ;
                           2      ;********************************
                           3      ;
                           4      ;  * COMPUTATION MODULE
                           5      ;
                           6      ;********************************
 FFF8                      7      INPUT   EQU     0FFF8H  ;P2A-P1A
 FFFA                      8      OUTPUT  EQU     0FFFAH  ;P2B-P1B
 FFFE                      9      CONTROL EQU     0FFFEH
                          10      ;
 0389                     11      CWORD   EQU     089H
 0009                     12      ENINT   EQU     09H
 0008                     13      DIINT   EQU     08H
 00FF                     14      MFLAG   EQU     0FFH
 AAAA                     15      SQUARE  EQU     0AAAAH
                          16      ;
 FFF2                     17      USARTST EQU     0FFF2H
 FFF0                     18      USARTDT EQU     0FFF0H
 0002                     19      RXRDY   EQU     02H
                          20      ;
 5555                     21      SILENCE1        EQU             5555H
 AAAA                     22      SILENCE2        EQU             0AAAAH
 FFFF                     23      LOSTHD          EQU             0FFFFH
 00FF                     24      BUFHD           EQU             0FFH
 0003                     25      SAPER           EQU             3H
 000A                     26      DEEPPROB        EQU             10
 0010                     27      DEEPPTR         EQU             16
                          28      ;
 ----                     29      STOR    STRUC
 0000                     30      TOPPTR  DW      ?
 0002                     31      ENDPTR  DW      ?
 0004                     32      PUSHPTR DW      ?
 0006                     33      POPPTR  DW      ?
 0008                     34      BUFFER  DW      ?
 ----                     35      STOR    ENDS
                          36      ;
 ----                     37      ISRSK   STRUC
 0000                     38      ISRBP   DW      ?
 0002                     39      ISRDI   DW      ?
 0004                     40      ISRSI   DW      ?
 0006                     41      ISRBX   DW      ?
 ----                     42      ISRSK   ENDS
                          43      ;
                          44      ;FLAG EMEG IS THE SAME AS PKHAND
                          45      FLAG    RECORD  TPS:1,BM:1,TEMP2:1,TEMP3:1,RPS:1,EMEG:1,
                          46      &               SILMODE:1,DELOVER:1,TEMP1:1
                          47      ;
 ----                     48      P6      SEGMENT AT 100H
                          49              ASSUME  DS:P6,ES:P6
                          50      ;
```

```
LOC  OBJ                 LINE   SOURCE

                                        EXTRN   PROB:WORD, PROBCOUNT:WORD, DELAY:WORD, DELAYCOUNT:WORD,
                                &               HIBYTE:WORD, LOWBYTE:WORD, PERTHI:WORD, PERTLOW:WORD,
                                &               STACKPTR:WORD, STK2:WORD, OUTREG:WORD, INSER:WORD,
                                &               MINPROB:WORD, TOTPKTX:WORD, TX:WORD, TXTOP:WORD, TXEND:WORD,
                                &               BUF:WORD, BUFFEREND:WORD, BUFFERAREA:WORD,
                                &               HI:WORD, LOWI:WORD, PACKSIZE:WORD, MINDELAY:WORD,
                                &               MAXTSD:WORD, MINTSD:WORD, DUMPPTR:WORD, TXTEMPPTR:WORD,
                                &               MAXSIL:WORD, MINSIL:WORD, COUNTER:WORD, MAXSZ:WORD
                                ;
----                            P6      ENDS
                                ;
                                        EXTRN   POPFIFO:FAR, PUSHFIFO:FAR, BEGIN:FAR
                                ;
                                        PUBLIC  REPP, DO, COMPUTE
                                ;
----                            P1      SEGMENT AT 220H
                                        ASSUME  CS:P1
                                ;
0000                            REPP    LABEL   FAR
0000 9A1E002002                 REPP1:  CALL    FAR PTR COMPUTE
                                ;
                                ;THE FOLLOWING ROUTINE ALLOWS INTERACTION BETWEEN THE USER
                                ;AND THE PROGRAM. TYPE CTRL/S TO HALT PROGRAM EXECUTION
                                ;
0005 BAF2FF                             MOV     DX,USARTST      ;STOP IF USART RECEIVES A CTRL/S
0008 EC                                 IN      AL,DX
0009 2402                               AND     AL,RXRDY
000B 74F3                               JZ      REPP1
                                ;
000D BAF0FF                             MOV     DX,USARTDT
0010 EC                                 IN      AL,DX
0011 247F                               AND     AL,7FH
0013 3C13                               CMP     AL,13H          ;CTRL/S
0015 75E9                               JNZ     REPP1
                                ;
0017 CD02                               INT     2H              ;GOTO NMI ISR
                                ;
0019 EA0000----        E                JMP     BEGIN
                                ;
                                ;THE FOLLOWING ROUTINE CALCULATES A NEW POD ELEMENT IF POD QUEUE IS NOT FULL
                                ;IT ALSO CALCULATES A PND ELEMENT IF PND QUEUE IS NOT FULL
                                ;
001E                            COMPUTE PROC    FAR
001E BB0000            E                MOV     BX,OFFSET PROB
0021 8B4704                             MOV     AX,[BX].PUSHPTR
0024 050200                             ADD     AX,TYPE PUSHPTR
0027 3B4702                             CMP     AX,[BX].ENDPTR
002A 7502                               JNE     A1
002C 8B07                               MOV     AX,[BX].TOPPTR
                                ;
002E 3B4706                     A1:     CMP     AX,[BX].POPPTR
0031 7410                               JE      A2
0033 E83400                             CALL    GENPROB
0036 FA                                 CLI
0037 874704                             XCHG    AX,[BX].PUSHPTR
```

```
LOC  OBJ                 LINE     SOURCE

003A 8BD8                                 MOV     BX,AX
003C 8917                                 MOV     WORD PTR[BX],DX
003E FB                                   STI
003F FF060000       E                     INC     PROBCOUNT
                                  ;
0043 BB0000         E             A2:     MOV     BX,OFFSET DELAY
0046 8B4704                               MOV     AX,[BX].PUSHPTR
0049 050200                               ADD     AX,TYPE PUSHPTR
004C 3B4702                               CMP     AX,[BX].ENDPTR
004F 7502                                 JNE     A3
0051 8B07                                 MOV     AX,[BX].TOPPTR
                                  ;
0053 3B4706                       A3:     CMP     AX,[BX].POPPTR
0056 7501                                 JNE     A4
0058 C3                                   RET
                                  ;
0059 E85700                       A4:     CALL    GENDELAY
005C FA                                   CLI
005D 874704                               XCHG    AX,[BX].PUSHPTR
0060 8BD8                                 MOV     BX,AX
0062 8917                                 MOV     WORD PTR[BX],DX
0064 FB                                   STI
0065 FF060000       E                     INC     WORD PTR DELAYCOUNT
0069 C3                                   RET
                                  ;
                                  COMPUTE ENDP
                                  ;
                                  ;THIS SUBROUTINE GENERATES PACKET DISCARD DECISION
                                  ;
006A                              GENPROB PROC
006A 50                                   PUSH    AX
006B 53                                   PUSH    BX
006C BF0100                               MOV     DI,1H            ;GENERATE RANDOM NUMBER
006F BB0300                               MOV     BX,3H
0072 A10000         E                     MOV     AX,HIBYTE
0075 F7E3                                 MUL     BX
0077 A30000         E                     MOV     HIBYTE,AX
                                  ;
007A 8B360000       E                     MOV     SI,LOWBYTE
007E 8BC6                                 MOV     AX,SI
0080 F7E3                                 MUL     BX
0082 A30000         E                     MOV     LOWBYTE,AX
0085 01160000       E                     ADD     HIBYTE,DX
                                  ;
0089 8BC6                                 MOV     AX,SI
008B F7E7                                 MUL     DI
008D 01060000       E                     ADD     HIBYTE,AX
0091 81260000FF7F   E                     AND     HIBYTE,7FFFH
                                  ;
0097 2BD2                                 SUB     DX,DX            ;GENERATE PACKET DISCARD DECISION
0099 A10000         E                     MOV     AX,PERTHI
009C 39060000       E                     CMP     HIBYTE,AX
00A0 770E                                 JA      B0
                                  ;
00A2 7209                                 JB      B1
```

```
LOC  OBJ                LINE    SOURCE

                         161    ;
00A4 A10000         E    162            MOV     AX,PERTLOW
00A7 39060000       E    163            CMP     LOWBYTE,AX
00AB 7703                164            JA      B0
                         165    ;
00AD BAFFFF              166    B1:     MOV     DX,LOSTHD
                         167    ;
00B0 5B                  168    B0:     POP     BX
00B1 58                  169            POP     AX
00B2 C3                  170            RET
                         171    ;
                         172    GENPROB ENDP
                         173    ;
                         174    ;THIS SUBROUTINE GENERATES RANDOM NETWORK DELAY
                         175    ;
00B3                     176    GENDELAY        PROC
00B3 50                  177            PUSH    AX              ;GENERATE RANDOM NUMBER
00B4 53                  178            PUSH    BX
00B5 BF0100              179            MOV     DI,1H
00B8 BB0300              180            MOV     BX,3H
00BB A10000         E    181            MOV     AX,HI
00BE F7E3                182            MUL     BX
00C0 A30000         E    183            MOV     HI,AX
                         184    ;
00C3 8B360000       E    185            MOV     SI,LOWW
00C7 8BC6                186            MOV     AX,SI
00C9 F7E3                187            MUL     BX
00CB A30000         E    188            MOV     LOWW,AX
00CE 01160000       E    189            ADD     HI,DX
                         190    ;
00D2 8BC6                191            MOV     AX,SI
00D4 F7E7                192            MUL     DI
00D6 01060000       E    193            ADD     HI,AX
00DA 81260000FF7F   E    194            AND     HI,7FFFH
                         195    ;
00E0 E80300              196            CALL    DIST            ;GENERATE RANDOM DELAY
                         197    ;
00E3 5B                  198            POP     BX
00E4 58                  199            POP     AX
00E5 C3                  200            RET
                         201    ;
                         202    GENDELAY        ENDP
                         203    ;
                         204    ;THIS SUBROUTINE GENERATES A RANDOM NETWORK DELAY BASED ON THE RANDOM
                         205    ;NUMBER GENERATED BY MEANS OF A LOOK-UP TABLE
                         206    ;
00E6                     207    DIST    PROC
                         208    ;
00E6 2BD2                209            SUB     DX,DX
00E8 C3                  210            RET                     ;"RET" FOR ZERO NETWORK DELAY
                         211    ;                               ;CHANGE TO "NOP" CODE "90" FOR DELAY GENERATION
00E9 BB0000290           212            MOV     BX,OFFSET DA
00ED 2BF6                213            SUB     SI,SI
                         214    ;
00EF 2E8B00              215    C3:     MOV     AX,CS:[BX][SI]
```

```
LOC  OBJ                LINE    SOURCE

00F2 39060000       E    216           CMP     HI,AX
00F6 7715                217           JA      C0
00F8 720A                218           JB      C1
00FA 2E8B4002            219           MOV     AX,CS:[BX][SI+2]
00FE 39060000       E    220           CMP     LOWW,AX
0102 7709                221           JA      C0
                         222    ;
0104 2E8B5004            223    C1:    MOV     DX,CS:[BX][SI+4]
0108 2EFF4006            224           INC     WORD PTR CS:[BX][SI+6]
010C C3                  225           RET
                         226    ;
010D 83C60800            227    C0:    ADD     SI,D1-D0
0111 EBDC                228           JMP     C3
                         229    ;
                         230    DIST   ENDP
                         231    ;
                         232    ;THIS IS THE NETWORK DELAY LOOK-UP TABLE
                         233    ;THE TABLE IS LOADED INTO THIS LOCATION BY EXECUTING "L (TABLE NAME)"
                         234    ;IN THE SDK-86 MONITOR MODE
                         235    ;
0200                     236           ORG     200H
                         237    ;
0200                     238    DA     LABEL   WORD
0200 ????                239    D0     DW      ?               ;DELAY THRESHOLD, HIGH ORDER
0202 ????                240           DW      ?               ;DELAY THRESHOLD, LOW ORDER
0204 ????                241           DW      ?               ;DELAY VALUE
0206 ????                242           DW      ?               ;COUNT # OF TIME THIS DELAY VALUE IS USED
                         243    ;
0208 ????                244    D1     DW      ?
                         245    ;
----                     246    P1     ENDS
                         247    ;
                         248           END

ASSEMBLY COMPLETE, NO ERRORS FOUND
```

```
ISIS-II MCS-86 MACRO ASSEMBLER V2.1 ASSEMBLY OF MODULE M3
OBJECT MODULE PLACED IN :F1:M3.OBJ
ASSEMBLER INVOKED BY:  ASM86 :F1:M3.SRC NOXR PR(:LP:) DA(29-07-83)

LOC  OBJ         SOURCE

                 ;
                 ;**************************************************
                 ;
                 ;        INTERRUPT SERVICE ROUTINE MODULE
                 ;
                 ;**************************************************
                 ;
----             SERV    STRUC
0000             DCI     DW      ?
0002             SCI     DW      ?
0004             BCP     DW      ?
0006             BCX     DW      ?
0008             ACX     DW      ?
000A             DCX     DW      ?
000C             RETADD  DW      ?
000E             CODESEG DW      ?
----             SERV    ENDS
                 ;
  FFF8           INPUT   EQU     0FFF8H  ;P2A-P1A
  FFFA           OUTPUT  EQU     0FFFAH  ;P2B-P1B
  FFFE           CONTROL EQU     0FFFEH
                 ;
  0029           CWORD   EQU     029H
  0009           ENINT   EQU     09H
  0008           DIINT   EQU     08H
  00FF           MFLAG   EQU     0FFH
  AAAA           SQUARE  EQU     0AAAAH
                 ;
                 ;
  FFF0           USARTDT         EQU             0FFF0H          ;USART DATA PORT
  0055           SIL1            EQU             0055H           ;SILENCE PATTERN 1
  00AA           SIL2            EQU             00AAH           ;SILENCE PATTERN 2
  EEEE           SPACKET         EQU             0EEEEH          ;SILENCE PACKET HEADER
  DDDD           PREOFF          EQU             0DDDDH          ;LENGTH=0DDDDH INDICATES PRE-OS
  5555           SILENCE1        EQU             5555H
  AAAA           SILENCE2        EQU             0AAAAH
  FFFF           LOSTHD          EQU             0FFFFH
  00FF           BUFHD           EQU             0FFH
  0003           SAREA           EQU             3H
  0300           MAXSIZE         EQU             300H
                 ;
                 ;FLAG EMEG IS THE SAME AS PKHRND
                 FLAG    RECORD  TRS:1,BM:1,TEMP2:1,TEMP3:1,RRS:1,EMEG:1,
                 &               SILMODE:1,DELOVER:1,START:1
                 ;
----             P6      SEGMENT AT 100H
                         ASSUME  DS:P6,ES:P6
                 ;
                         EXTRN   STACKPTR:WORD,OUTREG:WORD,SERSEG:WORD,
                 &               SERVICE:WORD,INSER:BYTE,PREOS:WORD
```

```
LOC  OBJ                     SOURCE

                             ;
                                     EXTRN   VOICE:WORD, VHT:WORD, VLT:WORD, PACKSIZE:WORD, DELCOUNT:WORD,
                             &               TXTEMPPTR:WORD, TXTEMPSIZE:WORD, BUFPOP:WORD, DELAYPOP:WORD,
                             &               TXDELAY:WORD, SPK:WORD, SPKTH:WORD, DELZERO:WORD, PREOSPTR:WORD,
                             &               POSTOS:WORD, TSDURAT:WORD, MAXTSD:WORD, MINTSD:WORD, TKSPURT:WORD,
                             &               SILENCE:WORD, SILDURAT:WORD, MINSIL:WORD, MAXSIL:WORD, TXP:WORD,
                             &               TOTPKTX:WORD, SILPACK:WORD, BUFPUSH:WORD,
                             &               SPACKPTR1:WORD, SPACKPTR2:WORD, DELAYCOUNT:WORD
                             ;
                                     EXTRN   PXP:WORD, PKBUF:WORD, PSIZE:WORD,
                             &               PXTEMPPTR:WORD, PX:WORD,
                             &               BUFPK:WORD, DON:BYTE
                             ;
----                         P6 ENDS
                             ;
                                     EXTRN   SEP:WORD, BEGIN:FAR, POPFIFO:FAR
                             ;
                                     PUBLIC  ISR, PRE
                             ;
                             ;"SILENCE DETECTION"
                             ;ONCE 16-BIT DATA IS FORMED ON THE S/P CONVERTER, THE ROUTINE IS EXECUTED
                             ;TO READ THE DATA.  UPON COMPLETION OF A PACKET ASSEMBLY, SILENCE DETECTION
                             ;IS DONE.  DATA IS PREPARED FOR OUTPUT.  PACKET DELAY COUNTER IS DECREMENTED.
                             ;RETURN ADDRESS IS MODIFIED IF PACKET HANDLING ROUTINE IS REQUIRED SERVICE.
                             ;
----                         P2      SEGMENT AT 260H
                                     ASSUME  CS:P2
                             ;
0000                         ISR     PROC    FAR
0000 52                              PUSH    DX
0001 50                              PUSH    AX              ;SAVE EXT REGS IN SYSTEM STACK
0002 53                              PUSH    BX
0003 55                              PUSH    BP
0004 56                              PUSH    SI
0005 57                              PUSH    DI
                             ;
0006 87260000          E             XCHG    SP,STACKPTR
                             ;
000A 5D                              POP     BP              ;RESTORE INT REGS FROM USER STACK
000B 5F                              POP     DI
000C 5E                              POP     SI
000D 5B                              POP     BX
                             ;
000E BAFAFF                          MOV     DX,OUTPUT       ;OUTPUT AND INPUT 16-BIT DATA
0011 A10000            E             MOV     AX,OUTREG
0014 EF                              OUT     DX,AX
0015 BAF8FF                          MOV     DX,INPUT
0018 ED                              IN      AX,DX
                             ;
                             ;                               PREPARE SILENCE OUTPUT IN CASE OUTPUT
0019 813E00005555      E             CMP     OUTREG,SILENCE1 ;DOES NOT WORK PROPERLY
001F 740E                            JE      U0
0021 813E0000AAAA      E             CMP     OUTREG,SILENCE2
0027 7406                            JE      U0
0029 C70600005555      E             MOV     OUTREG,SILENCE1
```

```
LOC  OBJ                  LINE     SOURCE

                          106      ;
002F E85300               107      U8:     CALL    DETECT          ;GOTO SILENCE DETECT ROUTINE
                          108      ;
0032 E80401               109              CALL    REC
                          110      ;
0035 833E000000      E    111      U0:     CMP     DELCOUNT,0H     ;DELAY COUNTER = 0 ?
003A 7413                 112              JZ      U1
                          113      ;
003C FF0E0000        E    114      U4:     DEC     DELCOUNT        ;DEC DELAY COUNTER
0040 833E000000      E    115              CMP     DELCOUNT,0H
0045 7508                 116              JNZ     U1
0047 81C90800             117              OR      CX,MASK EMEG    ;SET EMEG AND DELOVER FLAGS
004B 81C90200             118              OR      CX,MASK DELOVER
                          119      ;
004F 53                   120      U1:     PUSH    BX              ;SAVE INT REGS
0050 56                   121              PUSH    SI
0051 57                   122              PUSH    DI
0052 55                   123              PUSH    BP
                          124      ;
0053 87260000        E    125              XCHG    SP,STACKPTR
                          126      ;
0057 803E0000FF      E    127              CMP     INSER,0FFH
005C 741F                 128              JE      U2
                          129      ;
005E F7C10800             130              TEST    CX,MASK EMEG            ;CHANGE RETURN ADD IF EMEG IS SET
0062 7419                 131              JZ      U2
0064 C6060000FF      E    132              MOV     INSER,0FFH
0069 8BEC                 133              MOV     BP,SP
006B B80000          E    134              MOV     AX,OFFSET SER
006E 87460C               135              XCHG    [BP].RETADD,AX
0071 A30000          E    136              MOV     SERVICE,AX
0074 B8----          E    137              MOV     AX,SEG SER
0077 87460E               138              XCHG    [BP].CODESEG,AX
007A A30000          E    139              MOV     SERSEG,AX
                          140      ;
007D 5F                   141      U2:     POP     DI              ;RESTORE EXT REGS
007E 5E                   142              POP     SI
007F 5D                   143              POP     BP
0080 5B                   144              POP     BX
0081 58                   145              POP     AX
0082 5A                   146              POP     DX
                          147      ;
0083 FB                   148              STI
0084 CF                   149              IRET
                          150      ;
                          151      ISR     ENDP
                          152      ;
                          153      ;PACKET ASSEMBLY AND SILENCE DETECTION
                          154      ;SILENCE PACKETS ARE NOT TRANSMITTED
                          155      ;
0085                      156      DETECT  PROC
0085 8900                 157              MOV     [BX][SI],AX     ;STORE INPUT
0087 46                   158              INC     SI
0088 46                   159              INC     SI
                          160      ;
```

```
LOC  OBJ                LINE    SOURCE

0089 50                                 PUSH    AX
008A 25FF00                             AND     AX,00FFH        ;CHECK FOR VOICE SAMPLE
008D 3D5500                             CMP     AX,SIL1
0090 7409                               JE      T30
0092 3DAA00                             CMP     AX,SIL2
0095 7404                               JE      T30
0097 FF060000       E                   INC     VOICE
                                ;
009B 58                         T30:    POP     AX
009C 2500FF                             AND     AX,0FF00H
009F 3D5555                             CMP     AX,SILENCE1
00A2 7409                               JE      T1
00A4 3DAAAA                             CMP     AX,SILENCE2
00A7 7404                               JE      T1
                                ;
00A9 FF060000       E           T0:     INC     VOICE
                                ;
00AD 3B360000       E           T1:     CMP     SI,PACKSIZE     ;DO SILENCE DETECTION ONCE A PACKET IS FORMED
00B1 7301                               JAE     T2
00B3 C3                                 RET
                                ;
00B4 A10000         E           T2:     MOV     AX,VOICE
00B7 F7C10400                           TEST    CX,MASK SILMODE ;SILENCE MODE?
00BB 7574                               JNZ     T3              ;YES, SILENCE
00BD 3B060000       E                   CMP     AX,VLT          ;VOICE MODE
00C1 733D                               JAE     T20             ;JMP IF BIGGER THAN LOWER THRESHOLD
                                ;
00C3 FF060000       E                   INC     SPK             ;INC SILENCE PACKET IN VOICE MODE
00C7 A10000         E                   MOV     AX,SPK
00CA 3B060000       E                   CMP     AX,SPKTH        ;EXCEED THE SILENCE THRESHOLD?
00CE 7236                               JB      T15             ;NO, JUMP
00D0 C70600000000   E                   MOV     SPK,0H          ;YES, CHANGE TO SILENCE MODE
                                ;
00D6 81C90400                           OR      CX,MASK SILMODE ;SET SILENCE MODE
00DA 8B360000       E                   MOV     SI,POSTOS       ;POST-OFFSET SIZE
00DE FF060000       E                   INC     SILENCE         ;# OF SILENT GAP
00E2 A10000         E                   MOV     AX,TSDURAT
00E5 3B060000       E                   CMP     AX,MAXTSD       ;GET MAX TALKSPURT LENGTH
00E9 7603                               JBE     T16
00EB A30000         E                   MOV     MAXTSD,AX
00EE 3B060000       E           T16:    CMP     AX,MINTSD       ;GET MIN TALKSPURT LENGTH
00F2 7303                               JAE     T17
00F4 A30000         E                   MOV     MINTSD,AX
00F7 C70600000100   E           T17:    MOV     TSDURAT,1H
00FD E90D00                             JMP     T4
                                ;
0100 C70600000000   E           T20:    MOV     SPK,0H
0106 FF060000       E           T15:    INC     TSDURAT         ;TALKSPURT DURATION + 1
010A E90000                             JMP     T4
                                ;
010D FF060000       E           T5:     INC     SILDURAT        ;INCREMENT SILENT DURATION
0111 A10000         E                   MOV     AX,SPACKPTR2    ;PUT 2 IN 1 AND TO BE SAVED IN BUF
                                ;
0114 833E000000     E                   CMP     SPACKPTR1,0H    ;CHECK SPACKPTR1 OVERWRITTEN
0119 740A                               JZ      T83
```

```
LOC  OBJ                LINE   SOURCE

011B B82529                            MOV   AX,2925H
011E CD02                              INT   2H
0120 EA0000----      E                 JMP   BEGIN
                               ;
0125 A30000          E         T83:    MOV   SPACKPTR1,AX
0128 891E0000        E                 MOV   SPACKPTR2,BX    ;SAVE SIL PACKET POINTER
012C BBEEEE                            MOV   BX,SPACKET      ;HEADER INDICATES SILENCE
012F EB5C                              JMP   SHORT T4
                               ;
0131 3B060000        E         T3:     CMP   AX,VHT          ;SILENCE MODE
0135 7206                              JB    T5
                               ;
                               ;
0137 F7C10100                          TEST  CX,MASK START
013B 7510                              JNZ   T28
013D 81C90100                          OR    CX,MASK START   ;SET START MODE, CLEAR TOTPKTX AND SILPACK
                                                             ;ONCE SPEECH IS STARTED
0141 C70600000000    E                 MOV   TOTPKTX,0H
0147 C70600000000    E                 MOV   SILPACK,0H
                               ;
                               ;
014D 81E1FBFF                  T28:    AND   CX,NOT MASK SILMODE   ;CHANGE TO VOICE MODE
0151 FF060000        E                 INC   TKSPURT         ;# OF TALKSPURT
0155 A10000          E                 MOV   AX,SPACKPTR2    ;STORE LAST PTR IN PREOSPTR
                               ;
0158 833E000000      E                 CMP   PREOSPTR,0H     ;CHECK PREOSPTR OVERWRITTEN
015D 740A                              JZ    T62
015F B86824                            MOV   AX,2468H
0162 CD02                              INT   2H
0164 EA0000----      E                 JMP   BEGIN
                               ;
0169 A30000          E         T62:    MOV   PREOSPTR,AX
016C C70600000000    E                 MOV   SPACKPTR2,0H    ;CLEAR IT
0172 A10000          E                 MOV   AX,SILDURAT     ;UPDATE MAX AND MIN SILENT DURATION
0175 3B060000        E                 CMP   AX,MAXSIL
0179 7603                              JBE   T18
017B A30000          E                 MOV   MAXSIL,AX
017E 3B060000        E         T18:    CMP   AX,MINSIL
0182 7303                              JAE   T19
0184 A30000          E                 MOV   MINSIL,AX
0187 C70600000100    E         T19:    MOV   SILDURAT,1H
                               ;
018D 833E000000      E         T4:     CMP   TXTEMPPTR,0H
0192 740A                              JZ    T47
0194 B8CDAB                            MOV   AX,0ABCDH       ;CHECK TXTEMPPTR ERASED
0197 CD02                              INT   2H
0199 EA0000----      E                 JMP   BEGIN
                               ;
019E 891E0000        E         T47:    MOV   TXTEMPPTR,BX
01A2 89360000        E                 MOV   TXTEMPSIZE,SI
01A6 81C90001                          OR    CX,MASK TPS
01AA 81C90800                          OR    CX,MASK EMEG
                               ;
01AE 833E000000      E                 CMP   DELCOUNT,0H
01B3 7520                              JNZ   S10
```

```
LOC  OBJ                LINE     SOURCE

                                 ;
01B5 81FBEEEE                            CMP    BX,SPACKET      ;NO DELAY FOR SILENCE PACKET
01B9 7508                                JNE    T6
                                 ;
01BB C70600000100     E          T12:    MOV    DELCOUNT,1H     ;TRIGGER DELOVER
01C1 EB12                                JMP    SHORT S10
                                 ;
01C3 FF060000         E          T6:     INC    DELZERO         ;KEEP TRACK # OF PACKET GOES TO HERE
                                 ;
01C7 8B1E0000         E                  MOV    BX,DELAYPOP
01CB 8B17                                MOV    DX,[BX]
01CD 03160000         E                  ADD    DX,TXDELAY
01D1 89160000         E          T7:     MOV    DELCOUNT,DX
                                 ;
01D5 8B1E0000         E          S10:    MOV    BX,BUFPOP
                                 ;
01D9 3B1E0000         E                  CMP    BX,BUFPUSH      ;CHECK BUF FIFO OVERFLOW
01DD 750A                                JNE    T39
                                 ;
01DF B83412                              MOV    AX,1234H
01E2 CD02                        T35:    INT    2H
01E4 EA0000----       E                  JMP    BEGIN
                                 ;
                                 ;
01E9 8B1F                        T39:    MOV    BX,[BX]
                                 ;
                                 ;
01EB 3B1E0000         E                  CMP    BX,TXTEMPPTR    ;CHECK
01EF 7505                                JNE    T31
01F1 B87856                      T59:    MOV    AX,5678H
01F4 EBEC                                JMP    T35
                                 ;
                                 ;
01F6 3B1E0000         E          T31:    CMP    BX,PREOSPTR
01FA 74F5                                JE     T59             ;CHECK SAME PTR
                                 ;
01FC BE0300                              MOV    SI,SAREA
01FF C70600000000     E                  MOV    VOICE,0H
0205 C60700                              MOV    BYTE PTR[BX],0H
0208 C3                                  RET
                                 ;
                                 DETECT  ENDP
                                 ;
                                 ;THIS SUBROUTINE "DE-PACKETIZATION" PREPARES DATA FOR OUTPUT.
                                 ;IF A PACKET IS FINISHED OUTPUTTING, A NEW PACKET IS PREPARED FOR OUTPUT.
                                 ;
0209                             REC     PROC
0209 833E000000       E                  CMP    RXP,0           ;ANY PACKET IN RXFIFO?
020E 7501                                JNZ    V8
0210 C3                                  RET
                                 ;
0211 F7C18000                    V8:     TEST   CX,MASK BM
0215 7415                                JZ     V0
                                 ;
0217 A10000           E          V10:    MOV    AX,RXP          ;RXBUF=RXP?
```

```
LOC  OBJ                  SOURCE

021A 3E060000      E                CMP   AX, PKBUF
021E 7301                           JAE   V9
0220 C3                             RET
                            ;
0221 81E17FFF               V9:     AND   CX, NOT MASK BM
0225 3E8B03                         MOV   AX, DS:[BP][DI]   ;PREPARE OUTPUT
0228 A30000        E                MOV   OUTREG, AX
022B C3                             RET
                            ;
022C 47                     V0:     INC   DI
022D 47                             INC   DI
022E 3B3E0000      E                CMP   DI, PSIZE         ;PACKET OVER?
0232 7316                           JAE   V2
                            ;
0234 B85555                 V11:    MOV   AX, SILENCE1      ;OUTPUT SILENCE PATTERN FOR LOST PACKET
0237 81FDFFFF                       CMP   BP, LOSTHD        ;AND SILENCE PACKET
023B 7409                           JE    V3
023D 81FDEEEE                       CMP   BP, SPACKET
0241 7403                           JE    V3
0243 3E8B03                         MOV   AX, DS:[BP][DI]
0246 A30000        E        V3:     MOV   OUTREG, AX
                            ;
0249 C3                             RET
                            ;
024A FF0E0000      E        V2:     DEC   RXP
                            ;
024E 81FDEEEE                       CMP   BP, SPACKET       ;NO RX SERVICE FOR SIL PACKET
0252 7423                           JE    V49
                            ;
0254 81FDFFFF                       CMP   BP, LOSTHD        ;NO RX SERVICE
0258 741D                           JE    V49
                            ;
025A 833E000000    E                CMP   RXTEMPPTR, 0H     ;CHECK RXTEMPPTR OVERWRITTEN
025F 740A                           JZ    V47
0261 B85713                         MOV   AX, 1357H
0264 CD02                           INT   2H
0266 EA0000----    E                JMP   BEGIN
                            ;
026B 892E0000      E        V47:    MOV   RXTEMPPTR, BP
026F 81C90080                       OR    CX, MASK EMEG
0273 81C91000                       OR    CX, MASK RPS
                            ;
0277 BF0300                 V49:    MOV   DI, SAREA
027A 833E000000    E                CMP   RXP, 0
027F 7504                           JNZ   V13
0281 BDFFFF                         MOV   BP, LOSTHD
0284 C3                             RET
                            ;
0285 B80000        E        V13:    MOV   AX, OFFSET RX     ;PREPARE A NEW PACKET FOR OUTPUT
0288 9A0000----    E                CALL  POPFIFO
028D 8BEA                           MOV   BP, DX
028F 3E817E01DDDD                   CMP   DS:[BP+1], PREOFF
0295 7508                           JNE   V44
0297 8B3E0000      E                MOV   DI, PACKSIZE
029B 2B3E0000      E                SUB   DI, PREOS
```

```
LOC  OBJ                  SOURCE

029F 9A05025002           V44:      CALL    FAR PTR PREPX
02A4 C3                   V7:       RET
                          ;
                          REC       ENDP
                          ;
                          ;
02A5                      PRE       LABEL   FAR
02A5                      PREPX     PROC    FAR
02A5 81FDFFFF                       CMP     BP,LOSTHD        ;PREPARE PACKET POINTER AND SIZE
02A9 7406                           JE      V6
02AB 81FDEEEE                       CMP     BP,SPACKET
02AF 7500                           JNE     V4
                          ;
02B1 A10030          E    V6:       MOV     AX,PACKSIZE
02B4 A30030          E              MOV     PSIZE,AX
02B7 EB28                           JMP     SHORT DISPLAY
                          ;
02B9 3E8B4601             V4:       MOV     AX,DS:[BP+1]
02BD 3DDDDD                         CMP     AX,PREOFF        ;PRE-OFFSET PACKET
02C0 7503                           JNE     V41
02C2 A10030          E              MOV     AX,PACKSIZE
02C5 A30030          E    V41:      MOV     PSIZE,AX
02C8 3E8A4600                       MOV     AL,DS:[BP]
02CC 3CFF                           CMP     AL,BUFHD
02CE 7509                           JNZ     V5
                          ;
02D0 81C90000             V14:      OR      CX,MASK BM
02D4 FF060000        E              INC     BUFPK
02D8 CB                             RET
                          ;
02D9 3E8B03               V5:       MOV     AX,DS:[BP][DI]
02DC A30000          E              MOV     OUTREG,AX
02DF EB00                           JMP     SHORT DISPLAY
                          ;
02E1 803E0000FF      E    DISPLAY:          CMP     DON,0FFH      ;DISPLAY "*" ON THE TERMINAL FOR EVERY
02E6 7401                           JE      Z0                    ;PACKET OUTPUTTED. DISPLAY " " FOR LOST
02E8 CB                             RET                           ;PACKET
                          ;
02E9 81FDEEEE             Z0:       CMP     BP,SPACKET
02ED 7304                           JAE     Z2
02EF B04F                           MOV     AL,'O'
02F1 EB0C                           JMP     SHORT SHOW
                          ;
02F3 81FDEEEE             Z2:       CMP     BP,SPACKET
02F7 7504                           JNE     Z4
02F9 B020                           MOV     AL,' '
02FB EB02                           JMP     SHORT SHOW
                          ;
02FD B02A                 Z4:       MOV     AL,'*'
                          ;
02FF 52                   SHOW:     PUSH    DX
0300 BAF0FF                         MOV     DX,USARTDT
0303 EE                             OUT     DX,AL
0304 5A                             POP     DX
0305 CB                             RET
```

```
LOC  OBJ                LINE     SOURCE

                         436     ;
                         437     PPERX    ENDP
                         438     ;
                         439     P2       ENDS
                         440     ;
                         441              END

ASSEMBLY COMPLETE, NO ERRORS FOUND
```

ISIS-II MCS-86 MACRO ASSEMBLER V2.1 ASSEMBLY OF MODULE M4
OBJECT MODULE PLACED IN :F1:M4.OBJ
ASSEMBLER INVOKED BY: ASM86 :F1:M4.SRC NOHR EP(:LP:) DA(29-07-83)

```
LOC  OBJ           LINE     SOURCE

                            ;
                            ;************************************
                            ;
                            ;        PACKET HANDLING MODULE
                            ;
                            ;************************************
                            ;
----                        STOR    STRUC
0000                        TOPPTR  DW      ?
0002                        ENDPTR  DW      ?
0004                        PUSHPTR DW      ?
0006                        POPPTR  DW      ?
0008                        BUFFER  DW      ?
----                        STOR    ENDS
                            ;
----                        ISPSK   STRUC
0000                        ISPBP   DW      ?
0002                        ISPDI   DW      ?
0004                        ISPSI   DW      ?
0006                        ISPBX   DW      ?
----                        ISPSK   ENDS
                            ;
----                        SERV    STRUC
0000                        DCI     DW      ?           ;DI INDEX REGISTER
0002                        DCX     DW      ?           ;DX DATA REGISTER
0004                        SCI     DW      ?           ;SI INDEX REGISTER
0006                        BCP     DW      ?           ;BP BASE REGISTER
0008                        BCX     DW      ?           ;BX DATA REGISTER
000A                        ACX     DW      ?           ;AX REGISTER
000C                        RETADD  DW      ?           ;RETURN ADDRESS
000E                        CODESEG DW      ?           ;RETURN CODE SEGMENT
----                        SERV    ENDS
                            ;
  FFF8                      INPUT   EQU     0FFF8H  ;P2A-P1A
  FFFA                      OUTPUT  EQU     0FFFAH  ;P2B-P1B
  FFFE                      CONTROL EQU     0FFFEH
                            ;
  0089                      CWORD   EQU     089H
  0089                      ENINT   EQU     89H
  0008                      DIINT   EQU     08H
  00FF                      MFLAG   EQU     0FFH
  AAAA                      SQUARE  EQU     0AAAAH
                            ;
  5555                      SILENCE1        EQU             5555H
  AAAA                      SILENCE2        EQU             0AAAAH
  FFFF                      LOSTHD          EQU             0FFFFH
  00FF                      BUFHD           EQU             0FFH
  0003                      SAFEA           EQU             3H
  0010                      DEEPPTR         EQU             16
  EEEE                      SPACKET         EQU             0EEEEH
```

```
LOC  OBJ                 LINE    SOURCE

 0000                            PREOFF        EQU          0C000H
                                 ;
                                 ;FLAG EMEG IS THE SAME AS PKHAND
   ****                          FLAG    RECORD  TPS:1,BM:1,TEMP2:1,TEMP3:1,PPS:1,EMEG:1,
                                 &               SILMODE:1,DELOVER:1,START:1
                                 ;
 ----                            P6      SEGMENT AT 100H
                                         ASSUME DS:P6,ES:P6
                                 ;
                                         EXTRN   TXP:WORD,RXP:WORD,STACKPTR:WORD,INSEP:BYTE,RXPZERO:WORD,
                                 &               TOTRXRX:WORD,MAXRX:WORD,TXDELAY:WORD,TXPOP:WORD,RXPUSH:WORD,
                                 &               DELCOUNT:WORD,PTDIS:WORD,TXTEMPPTR:WORD,RXPOP:WORD,
                                 &               TXTEMPSIZE:WORD,PACKSIZE:WORD,NOTFULL:WORD,TOTRXTX:WORD,
                                 &               MAXTX:WORD,RXTEMPPTR:WORD,SERVICE:WORD,TX:WORD,RX:WORD,
                                 &               BUF:WORD,PROB:WORD,DELAY:WORD,DELAYCOUNT:WORD,MINDELAY:WORD,
                                 &               PROBCOUNT:WORD,MINPROB:WORD,SEPSEG:WORD,SILPACK:WORD,
                                 &               TEMPRX:WORD,PREOSPTR:WORD,BUFTOP:WORD,STACKPTR1:WORD,
                                 &               PREOS:WORD,DISDY:WORD,STOP:WORD,STK2:WORD

                                 ;
 ----                            P6      ENDS
                                 ;
                                         EXTRN   PRE:FAR,BEGIN:FAR
                                 ;
 ----                            P5      SEGMENT AT 200H
                                         ASSUME  CS:P5
                                 ;
                                         PUBLIC  SEP,POPFIFO,PUSHFIFO
                                 ;
                                 ;THIS ROUTINE CHECKS WHICH FLAGS, PPS, TPS AND DELOVER, AND EXECUTES
                                 ;THE CORRESPONDING SUBROUTINE
                                 ;
 0000                            SEP     PROC    FAR
 0000 9C                                 PUSHF
 0001 9C                                 PUSHF                   ;RESERVE SPACE FOR RETURN CODE SEGMENT
 0002 9C                                 PUSHF                   ;RESERVE SPACE FOR RETURN ADDRESS
 0003 50                                 PUSH    AX
 0004 53                                 PUSH    BX
 0005 55                                 PUSH    BP
 0006 56                                 PUSH    SI
 0007 52                                 PUSH    DX
 0008 57                                 PUSH    DI
                                 ;
 0009 8BEC                               MOV     BP,SP
 000B A10000          E                  MOV     AX,SERVICE   ;SET UP RETURN ADDRESS
 000E 89460C                             MOV     [BP].RETADD,AX
 0011 A10000          E                  MOV     AX,SEPSEG    ;SET UP RETURN CODE SEGMENT
 0014 89460E                             MOV     [BP].CODESEG,AX
                                 ;
 0017 F7C11000                   Z3:     TEST    CX,MASK PPS        ;PPS REQUIRES SERVICE?
 001B 7403                               JZ      Z0
 001D E81202                             CALL    PSEP
                                 ;
 0020 F7C10001                   Z0:     TEST    CX,MASK TPS        ;TPS REQUIRES SERVICE?
 0024 7405                               JZ      Z1
```

```
LOC  OBJ                     SOURCE

0026 E83701                            CALL    TSEP
0029 EB09                              JMP     SHORT Z2
                             ;
002B F7C10200                Z1:       TEST    CX,MASK DELOVER   ;DELOVER REQUIRES SERVICE?
002F 7403                              JZ      Z2
0031 E81200                            CALL    DOVER
                             ;
0034 F7C11201                Z2:       TEST    CX,1000100100B
0038 75DD                              JNZ     Z3
                             ;
003A 5F                                POP     DI
003B 5A                                POP     DX
003C 5E                                POP     SI
003D 5D                                POP     BP
003E 5B                                POP     BX
003F 58                                POP     AX
                             ;
0040 C606000000       E                MOV     INSER,0H
0045 CF                                IRET
                             ;
                             SER       ENDP
                             ;
                             ;THIS SUBROUTINE IS EXECUTED WHEN THE PACKET DELAY IS OVER
                             ;A PACKET IS TRANSMITTED TO THE RECEIVER IF IT IS NOT DISCARDED.
                             ;IF THE RECEIVER IS EMPTY, A PACKET IS PREPARED FOR OUTPUT
                             ;IF THE TRANSMITTER IS NOT EMPTY, A PACKET WILL BE DELAYED AND DISCARDED
                             ;ACCORDINGLY
                             ;
0046                         DOVER     PROC
0046 B80000           E      W4:       MOV     AX,OFFSET TX       ;GET TRANSMIT PACKET POINTER
0049 9A6F021D002                       CALL    FAR PTR POPFIFO
004E 8BDA                              MOV     BX,DX
                             ;
0050 81FBEEEE                W11:      CMP     BX,SPACKET         ;SIL PACKET
0054 7506                              JNE     W16                ;NO, JMP
0056 FF060000         E                INC     SILPACK            ;YES
005A EB34                              JMP     SHORT W0
                             ;
005C B80000           E      W16:      MOV     AX,OFFSET PROB  ;GET PACKET DISCARD DECISION
005F 9A6F021D002                       CALL    FAR PTR POPFIFO
                             ;
0064 FF0E0000         E                DEC     PROBCOUNT          ;UPDATE MINPROB
0068 A10000           E                MOV     AX,PROBCOUNT
006B 3B060000         E                CMP     AX,MINPROB
006F 7303                              JAE     W9
0071 A30000           E                MOV     MINPROB,AX
                             ;
0074 87DA                    W9:       XCHG    BX,DX
0076 81FBFFFF                          CMP     BX,LOSTHD
007A 7406                              JE      W7
                             ;
007C FF060000         E      W12:      INC     TOTPKRX
0080 EB0E                              JMP     SHORT W0
                             ;
0082 FF060000         E      W7:       INC     PKDIS
```

```
LOC  OBJ                LINE   SOURCE

0086 B80000          E         H17:    MOV    AX,OFFSET BUF      ;RETURN DISCARDED PACKET TO BUFFER-POOL
0089 9A4D020002                        CALL   FAR PTR PUSHFIFO
008E 87DA                              XCHG   BX,DX
                               ;
0090 B80000          E         W0:     MOV    AX,OFFSET RX
0093 89160000        E                 MOV    TEMPRX,DX
0097 9A4D020002                        CALL   FAR PTR PUSHFIFO  ;TRANSMIT PACKET TO RECEIVER
009C FF0E0000        E                 DEC    TXP
                               ;
                               ;
00A0 FA                                CLI
00A1 833E000000      E                 CMP    RXP,0H             ;IS RECEIVE QUEUE EMPTY?
00A6 753E                              JNZ    W2
00A8 FB                                STI
00A9 81FAEEEE                  W5:     CMP    DX,SPACKET         ;NO UPDATE RXPZERO FOR SIL AND LOST PACKET
00AD 740A                              JE     W21
00AF 81FAFFFF                          CMP    DX,LOSTHD
00B3 7404                              JE     W31
                               ;
00B5 FF060000        E                 INC    RXPZERO            ;COUNT # OF PACKETS RECEIVE WHEN RXP IS EMPTY
                               ;
00B9 B80000          E         W21:    MOV    AX,OFFSET RX
00BC 9A5F020002                        CALL   FAR PTR POPFIFO ;GET PACKET POINTER
                               ;
00C1 8B2E0000        E                 MOV    BP,STACKPTR
00C5 895600                            MOV    [BP].ISRBP,DX
00C8 BF0300                            MOV    DI,SAREA
00CB 8BDA                              MOV    BX,DX
00CD 817F01DDDD                        CMP    [BX+1].PREOFF     ;PRE-OFFSET PACKET?
00D2 7508                              JNE    W52
00D4 8B3E0000        E                 MOV    DI,PACKSIZE
00D8 2B3E0000        E                 SUB    DI,PREOS
00DC 897E02                    W52:    MOV    [BP].ISRDI,DI
00DF 8BEA                              MOV    BP,DX
00E1 9A0000          E                 CALL   PRE
                               ;
00E6 FF060000        E         W2:     INC    RXP
00EA FB                                STI
00EB 8B1E0000        E                 MOV    BX,TEMPRX          ;NO DELAY FOR SILENCE PACKETS
00EF 81FBFFFF                          CMP    BX,LOSTHD
00F3 7412                              JE     H10
00F5 81FBEEEE                          CMP    BX,SPACKET
00F9 740C                              JE     H10
                               ;
00FB A10000          E                 MOV    AX,RXP
00FE 3B060000        E                 CMP    AX,MAXRX
0102 7603                              JBE    H10
                               ;
0104 A30000          E                 MOV    MAXRX,AX
                               ;
0107 81E1FDFF                  H10:    AND    CX,NOT MASK DELOVER
010B 81E1F7FF                          AND    CX,NOT MASK EMEG
                               ;
                               ;
010F 833E000000      E                 CMP    DELCOUNT,0H
```

```
LOC  OBJ                  SOURCE

0114 7549                         JNZ    W8
0116 833E000000      E            CMP    TXP,0H           ;ANY PACKETS IN TRANSMIT QUEUE?
011B 7442                 W15:    JZ     W8
                          ;
011D 8B1E0000        E    W3:     MOV    BX,TXPOP         ;DO NOT INTRODUCE FOR SIL PACKETS
0121 8B1F                         MOV    BX,[BX]
                          ;
0123 81FBEEEE                     CMP    BX,SPACKET
0127 7507                         JNE    W22
0129 C70600000100    E    W23:    MOV    DELCOUNT,1H
012F C3                           RET
                          ;
0130 81FBFFFF             W22:    CMP    BX,LOSTHD               ;DISCARDED PACKET?
0134 7509                         JNE    W23
0136 8B160000        E            MOV    DX,DISDY                ;GET AVE. DISCARDED DELAY
013A 89160000        E            MOV    DELCOUNT,DX
013E C3                           RET
                          ;
013F B80000          E    W23:    MOV    AX,OFFSET DELAY
0142 9A5F020002                   CALL   FAR PTR POFFIFO
0147 03160000        E            ADD    DX,TXDELAY
014B 89160000        E    W19:    MOV    DELCOUNT,DX
014F FF0E0000        E            DEC    DELAYCOUNT
0153 A10000          E            MOV    AX,DELAYCOUNT
0156 3B060000        E            CMP    AX,MINDELAY
015A 7303                         JAE    W8
015C A30000          E            MOV    MINDELAY,AX
                          ;
015F C3                   W8:     RET
                          ;
                          DOVER   ENDP
                          ;
                          ;THIS SUBROUTINE IS EXECUTED WHEN THE TRANSMITTER IS REQUIRING SERVICE
                          ;STOP PROGRAM EXECUTION IF # OF PACKETS GENERATED EXCEEDS "STOP".
                          ;SAVE TRANSMIT PACKET POINTER IN TRANSMIT QUEUE AND ALSO THE PRE-OFFSET IF
                          ;REQUIRED.
0160                      TSER    PROC
0160 8B1E0000        E    X3:     MOV    BX,TXTEMPPTR
0164 C70600000000    E            MOV    TXTEMPPTR,0H
016A 8B360000        E            MOV    SI,TXTEMPSIZE
016E 3B360000        E            CMP    SI,PACKSIZE
0172 7304                         JAE    X0
                          ;
0174 FF060000        E    X13:    INC    NOTFULL
                          ;
0178 F7C10100             X0:     TEST   CX,MASK START
017C 7413                         JZ     X18
017E A10000          E            MOV    AX,STOP          ;STOP EXECUTION AFTER 'STOP' NUMBER OF
0181 39060000        E            CMP    TOTPKTX,AX       ;PACKETS ARE GENERATED
0185 720A                         JB     X18
0187 B82614                       MOV    AX,1426H
018A CD02                         INT    2H
018C EA0000----      E            JMP    BEGIN
                          ;
                          ;
```

```
LOC  OBJ                 LINE    SOURCE

0191 81FBEEEE             271    X18:   CMP    BX,SPACKET       ;NO UPDATE AND SAVE PTR FOR SIL PACKET
0195 7521                 272           JNE    X8
0197 8B160000       E     273           MOV    DX,SPACKPTR1
019B 23D2                 274           AND    DX,DX
019D 7446                 275           JZ     X5
019F C70600000000   E     276           MOV    SPACKPTR1,0H
01A5 87DA                 277           XCHG   BX,DX
01A7 C747010000           278           MOV    WORD PTR[BX+1],0H       ;CLEAR LENGTH
01AC 87DA                 279           XCHG   BX,DX
01AE B80000         E     280           MOV    AX,OFFSET BUF
01B1 9A4D02D002           281           CALL   FAR PTR PUSHFIFO
01B6 EB2D                 282           JMP    SHORT X5
                          283    ;
01B8 897701               284    X8:    MOV    [BX+1],SI
01BB 8B160000       E     285           MOV    DX,PREOSPTR      ;SAVE PRE-OFFSET IN TXBUF
01BF 23D2                 286           AND    DX,DX
01C1 7422                 287           JZ     X5
                          288    ;
01C3 C70600000000   E     289           MOV    PREOSPTR,0H
01C9 87DA                 290           XCHG   BX,DX
01CB C607FF               291           MOV    BYTE PTR[BX],BUFHD
01CE C747010DDD           292           MOV    [BX+1],PREOFF    ;INDICATE PRE-OFFSET
01D3 87DA                 293           XCHG   BX,DX
                          294    ;
01D5 B80000         E     295           MOV    AX,OFFSET TX
01D8 9A4D02D002           296           CALL   FAR PTR PUSHFIFO ;PUT PRE-OFFSET IN TRANSMIT QUEUE
01DD FF060000       E     297           INC    TOTPKTX
01E1 FF060000       E     298           INC    TXP
                          299    ;
01E5 B80000         E     300    X5:    MOV    AX,OFFSET TX
01E8 8BD3                 301           MOV    DX,BX
01EA 9A4D02D002           302           CALL   FAR PTR PUSHFIFO ;PUT PACKET IN TRANSMIT QUEUE
                          303    ;
01EF FF060000       E     304           INC    TOTPKTX
01F3 FF060000       E     305           INC    TXP
01F7 81FAEEEE             306           CMP    DX,SPACKET
01FB 740C                 307           JE     X32
01FD A10000         E     308           MOV    AX,TXP
0200 3B060000       E     309           CMP    AX,MAXTX
0204 7603                 310           JBE    X32
0206 A30000         E     311           MOV    MAXTX,AX
                          312    ;
0209 B80000         E     313    X32:   MOV    AX,OFFSET BUF    ;RELEASE PACKET FROM BUFFER-POOL QUEUE
020C 9A6F02D002           314           CALL   FAR PTR POPFIFO
                          315    ;
0211 B80000         E     316           MOV    AX,OFFSET DELAY
0214 9A6F02D002           317           CALL   FAR PTR POPFIFO
0219 FF0E0000       E     318           DEC    DELAYCOUNT
                          319    ;
021D A10000         E     320           MOV    AX,DELAYCOUNT
0220 3B060000       E     321           CMP    AX,MINDELAY
0224 7303                 322           JAE    X23
                          323    ;
0226 A30000         E     324           MOV    MINDELAY,AX
                          325    ;
```

```
LOC  OBJ                  LINE   SOURCE

0229 81E1FFFE                    X23:   AND    CX,NOT MASK TPS
022D 81E1F7FF                           AND    CX,NOT MASK EMEG
                                 ;
0231 C3                                 RET
                                 ;
                                 ;
                                 TSER   ENDP
                                 ;
                                 ;THIS SUBROUTINE IS ENTERED WHEN THE RECEIVER REQUIRES SERVICE
                                 ;AN EMPTY PACKET IS RETURNED TO THE BUFFER-POOL
                                 ;
0232                             PSER   PROC
                                 ;
0232 8B160000          E         Y3:    MOV    DX,PXTEMPPTR
0236 C70600000000      E                MOV    PXTEMPPTR,0H
                                 ;
023C B80000            E                MOV    AX,OFFSET BUF
023F 9A4D020032                         CALL   FAR PTR PUSHFIFO
                                 ;
0244 81E1EFFF                    Y1:    AND    CX,NOT MASK PPS
0248 81E1F7FF                           AND    CX,NOT MASK EMEG
                                 ;
024C C3                                 RET
                                 ;
                                 PSER   ENDP
                                 ;
                                 ;THIS IS THE SOFTWARE FIFO QUEUE "RELEASE" OPERATION
                                 ;A PACKET IS RELEASED IN A QUEUE IF THE "PUSHPTR" AFTER INCREMENTED BY ONE
                                 ;IS NOT EQUAL TO THE "POPPTR".  THE "PUSHPTR" IS CHANGED TO THE ADDRESS
                                 ;"TOPPTR" IF THE "ENDPTR" IS REACHED.
                                 ;
024D                             PUSHFIFO       PROC   FAR
024D 53                                 PUSH   BX
024E 50                                 PUSH   AX
024F FA                                 CLI
0250 8BD8                               MOV    BX,AX
0252 8B4704                             MOV    AX,[BX].PUSHPTR
0255 050200                             ADD    AX,TYPE PUSHPTR
0258 3B4702                             CMP    AX,[BX].ENDPTR
025B 7502                               JNE    S0
                                 ;
025D 8B07                               MOV    AX,[BX].TOPPTR
025F 3B4706                      S0:    CMP    AX,[BX].POPPTR
0262 7407                               JE     S1
                                 ;
0264 874704                             XCHG   AX,[BX].PUSHPTR
0267 8BD8                               MOV    BX,AX
0269 8917                               MOV    [BX],DX
026B FB                          S1:    STI
026C 58                                 POP    AX
026D 5B                                 POP    BX
                                 ;
026E CB                                 RET
                                 ;
                                 PUSHFIFO       ENDP
```

```
LOC  OBJ                LINE    SOURCE

                        381     ;
                        382     ;THIS IS THE SOFTWARE FIFO QUEUE "FETCH" OPERATION
                        383     ;A PACKET POINTER IS FETCHED IF THE "PUSHPTR" IS NOT EQUAL TO "POPPTR"
                        384     ;THE "POPPTR" IS INCREMENTED BY ONE AFTER A FETCH OPERATION
                        385     ;THE "POPPTR" IS CHANGED TO "TOPPTR" IF THE "ENDPTR" IS REACHED
                        386     ;
026F                    387     POPFIFO         PROC    FAR
026F 53                 388             PUSH    BX
0270 50                 389             PUSH    AX
0271 FA                 390             CLI
0272 8BD8               391             MOV     BX,AX
0274 8B4706             392             MOV     AX,[BX].POPPTR
0277 3B4704             393             CMP     AX,[BX].PUSHPTR
027A 7411               394             JE      S2
                        395     ;
027C 93                 396             XCHG    AX,BX
027D 8B17               397             MOV     DX,[BX]
027F 93                 398             XCHG    AX,BX
0280 050200             399             ADD     AX,2H
0283 3B4702             400             CMP     AX,[BX].ENDPTR
0286 7502               401             JNE     S3
                        402     ;
0288 8B07               403             MOV     AX,[BX].TOPPTR
028A 894706             404     S3:     MOV     [BX].POPPTR,AX
                        405     ;
028D 81FC0000      E    406     S2:     CMP     SP,OFFSET STK2
0291 7401               407             JE      S5
0293 FB                 408             STI
0294 58                 409     S5:     POP     AX                  ;DO NOT ENABLE INTERRUPT IN THE ISR
0295 5B                 410             POP     BX
0296 CB                 411             RET
                        412     ;
                        413     POPFIFO ENDP
                        414     ;
----                    415     P5      ENDS
                        416     ;
                        417             END

ASSEMBLY COMPLETE, NO ERRORS FOUND
```

```
ISIS-II MCS-86 MACRO ASSEMBLER V2.1 ASSEMBLY OF MODULE M5
OBJECT MODULE PLACED IN :F1:M5.OBJ
ASSEMBLER INVOKED BY:  ASM86 :F1:M5.SRC NOMR EP DA(29-07-83)


LOC  OBJ                LINE    SOURCE

                                ;*****************************
                                ;
                                ;          DATA MODULE
                                ;
                                ;*****************************
                                ;
 000A                           DEEPPROB        EQU     10      ;SIZE OF PDD AND PND FIFO QUEUES
 0010                           DEEPPTR         EQU     16      ;SIZE OF BUFFER-POOL, TRANSMIT & RECEIVE QUEUES
 008C                           MAXSIZE         EQU     140     ;MAX SIZE OF A PACKET, PREPARE BUFPTR
                                ;
                                            PUBLIC  PACKSIZE, PKBUF, STACKPTR, OUTREG, VHT, VLT, COUNTER, DISDY, DON,
                                &                   PERTHI, PERTLOW, SERVICE, SERSEG, TXDELAY, MINPROB, MAXSZ, STOP,
                                &                   MINDELAY, HI, LOWW, HIBYTE, LOWBYTE, INSER, SPKTH, PREOS, POSTOS
                                ;
                                            PUBLIC  TOTPKTX, TXP, MAXTX, TXTEMPPTR, TXTEMPSIZE, RXP, NOTFULL,
                                &                   MAXRX, BUFPK, TOTPKRX, RXTEMPPTR, RXDIS, SILPACK, VOICE,
                                &                   PSIZE, PROBCOUNT, DELAYCOUNT, DELCOUNT, SPK, DELZERO, RXPZERO,
                                &                   TEMPRX, PREOSPTR, TXSPURT, SPACKPTR1, SPACKPTR2,
                                &                   MAXTSD, MINTSD, TSDWAT, SILENCE, SILWAT, MAXSIL, MINSIL
                                ;
                                            PUBLIC  TX, BUF, RX, PROB, DELAY, BUFFERAREA, BUFFEREND, TXTOP, TXEND,
                                &                   BUFPOP, BUFPUSH, RXPOP, DELAYPOP, TXPOP, RXPUSH, DUMPPTR, BUFTOP
                                ;
                                ;THE FOLLOWING LOCATIONS DEFINES VARIABLE AND THEIR VALUES
                                ;
 ----                           P6      SEGMENT AT 100H
                                        ASSUME  DS:P6, ES:P6
                                ;
0000 8A00                       PACKSIZE        DW      138     ;PACKET SIZE IN BYTES
0002 0000                       PKBUF           DW      0       ;# OF PACKETS NEED BUFFERING
0004 ????                       STACKPTR        DW      ?       ;TEMP STORAGE OF STACK POINTER
0006 ????                       OUTREG          DW      ?       ;DATA OUTPUT
0008 FF                         DON             DB      0FFH    ;FF-ENABLE DISPLAY
0009 FF                                         DB      0FFH
000A ??                         INSER           DB      ?       ;FLAG INDICATES SER IS IN SERVICE
000B FF                                         DB      0FFH
000C FFFF                       STOP            DW      0FFFFH  ;NUMBER OF PACKETS GENERATED BEFORE STOPPING
000E 4C00                       VLT             DW      76      ;SILENCE THRESHOLD
0010 6700                       VHT             DW      103     ;VOICE THRESHOLD
0012 1E00                       PREOS           DW      30      ;PRE-OFFSET SIZE
0014 1E00                       POSTOS          DW      30      ;POST-OFFSET SIZE
0016 0000                       SPKTH           DW      0       ;HANG-OVER SIZE
0018 4701                       PERTHI          DW      147H    ;PROBABILITY THRESHOLD - 1 %
001A 14AE                       PERTLOW         DW      0AE14H  ;LOWER ORDER
001C 0100                       DISDY           DW      1H      ;AVE DISCARD DELAY
001E 2000                       COUNTER         DW      32      ;NO OF INTERVALS TO BE CLEARED
0020 ????                       SERVICE         DW      ?       ;ADDRESS OF PKHAND ROUTINE
0022 ????                       SERSEG          DW      ?       ;CODE SEGMENT OF PKHAND ROUTINE
0024 0100                       TXDELAY         DW      1H      ;PROPAGATION DELAY
0026 8C00                       MAXSZ           DW      MAXSIZE ;MAX PACKET SIZE USE TO PREPARE BUFPTR
```

```
LOC  OBJ        SOURCE

0028 0000       HI          DW    0H      ;INIT VALUE FOR PDD GENERATIONS
002A 2F13       LOWW        DW    4911    ;LOWER ORDER
002C 0000       HIBYTE      DW    0H      ;INIT VALUE FOR PND GENERATIONS
002E 9F07       LOWBYTE     DW    1951    ;LOWER ORDER
                ;
                ;THE FOLLOWING LOCATIONS DEFINES VARIABLES WHICH ARE CLEARED
                ;EVERYTIME THE PROGRAM IS RUN
                ;
0030 ????       TOTPKTX     DW    ?       ;TOT # OF PACKETS GENERATED
0032 ????       TOTPKRX     DW    ?       ;:::::::::::::::: RECEIVED
0034 ????       PKDIS       DW    ?       ;:::::::::::::::: DISCARDED
0036 ????       SILPACK     DW    ?       ;# OF SILENCE PACKETS
0038 ????       TXP         DW    ?       ;# OF PACKETS IN TXFIFO
003A ????       MAXTX       DW    ?       ;MAX # OF PACKETS IN TXFIFO
003C ????       RXP         DW    ?       ;# OF PACKETS IN RXFIFO
003E ????       MAXRX       DW    ?       ;MAX # OF PACKETS IN RXFIFO
0040 ????       TXTEMPPTR   DW    ?       ;TEMP STORAGE OF TRANSMIT PACKET POINTER
0042 ????       TXTEMPSIZE  DW    ?       ;TEMP STORAGE OF TRANSMIT PACKET SIZE
0044 ????       RXTEMPPTR   DW    ?       ;TEMP STORAGE OF RECEIVE PACKET POINTER
0046 ????       NOTFULL     DW    ?       ;# OF NOT FULL-SIZE PACKETS
0048 ????       BUFPK       DW    ?       ;# OF PACKETS BUFFERED
                ;
004A ????       VOICE       DW    ?       ;# OF VOICE SAMPLE
004C ????       SPK         DW    ?       ;TEMP # OF SILENCE PACKET
004E ????       SPACKPTR1   DW    ?       ;1ST OF 2 TEMP FIFO STORAGE
0050 ????       SPACKPTR2   DW    ?       ;2ND OF 2 TEMP FIFO STORAGE
0052 ????       DELZERO     DW    ?       ;# OF TIMES TX BUF IS EMPTY
0054 ????       RXPZERO     DW    ?       ;# OF TIMES RX BUF IS EMPTY
0056 ????       PSIZE       DW    ?       ;RECEIVE PACKET SIZE
0058 ????       TEMPRX      DW    ?       ;TEMP STORAGE FOR PTR PUSH IN RXFIFO
005A ????       PREOSPTR    DW    ?       ;INDICATES NO DELAY AND PROB
005C ????       TKSPURT     DW    ?       ;NUMBER OF TALKSPURTS
005E ????       MAXTSD      DW    ?       ;MAX TALKSPURT DURATION
0060 ????       MINTSD      DW    ?       ;MIN TALKSPURT DURATION
0062 ????       SILENCE     DW    ?       ;NUMBER OF SILENT GAP
0064 ????       MAXSIL      DW    ?       ;MAX SIL GAP
0066 ????       MINSIL      DW    ?       ;MIN SIL GAP
0068 ????       MINPROB     DW    ?       ;MIN PROB IN PROBFIFO
006A ????       MINDELAY    DW    ?       ;MIN DELAY IN DELAYFIFO
006C ????       TSDURAT     DW    ?       ;TEMP TALKSPURT DURATION
006E ????       SILDURAT    DW    ?       ;TEMP SILENT DURATION
0070 ????       DELCOUNT    DW    ?       ;PACKET DELAY COUNTER
0072 ????       PROBCOUNT   DW    ?       ;# OF ELEMENTS IN PDD QUEUE
0074 ????       DELAYCOUNT  DW    ?       ;# OF ELEMENTS IN PND QUEUE
                ;
                ;FIFO QUEUES STORAGE AREA
                ;
0076 ????       TX          DW    ?       ;TRANSMIT QUEUE, TOP POINTER OF QUEUE
0078 ????                   DW    ?       ;BOTTOM POINTER OF QUEUE
007A ????                   DW    ?       ;PUSH POINTER OF QUEUE
007C ????       TXPOP       DW    ?       ;POP POINTER OF QUEUE
007E ????                   DW    ?
0080 (16        TXTOP       DW    DEEPPTR DUP(?)  ;ELEMENT STORAGE OF QUEUE
      ????
      )
```

```
LOC  OBJ              LINE    SOURCE

00A0 ????              104    TXEND        DW     ?
                       105    ;
00A2 ????              106    BUF          DW     ?        ;BUFFER-POOL QUEUE
00A4 ????              107                 DW     ?
00A6 ????              108    BUFPUSH      DW     ?
00A8 ????              109    BUFPOP       DW     ?
00AA ????              110                 DW     ?
00AC (16               111    BUFTOP       DW     DEEPPTR DUP(?)
     ????
     )
00CC ????              112    BUFEND       DW     ?
                       113    ;
00CE ????              114    RX           DW     ?        ;RECEIVE QUEUE
00D0 ????              115                 DW     ?
00D2 ????              116    RXPUSH       DW     ?
00D4 ????              117    RXPOP        DW     ?
00D6 ????              118                 DW     ?
00D8 (16               119    RXTOP        DW     DEEPPTR DUP(?)
     ????
     )
00F8 ????              120    RXEND        DW     ?
                       121    ;
00FA ????              122    PPOB         DW     ?        ;PACKET DISCARD DECISION QUEUE
00FC ????              123                 DW     ?
00FE ????              124                 DW     ?
0100 ????              125                 DW     ?
0102 ????              126                 DW     ?
0104 (10               127    PPOBTOP      DW     DEEPPPOB DUP(?)
     ????
     )
0118 ????              128    PPOBEND      DW     ?
                       129    ;
011A ????              130    DELAY        DW     ?        ;RANDOM NETWORK DELAY QUEUE
011C ????              131                 DW     ?
011E ????              132                 DW     ?
0120 ????              133    DELAYPOP     DW     ?
0122 ????              134                 DW     ?
0124 (10               135    DELAYTOP     DW     DEEPPPOB DUP(?)
     ????
     )
0138 ????              136    DELAYEND     DW     ?
                       137    ;
                       138    ;BUFFER AREA
                       139    ;
013A ????              140    DUMPPTR      DW     ?        ;DUMMY POINTER FOR FIRST PRE-OFFSET PTR
013C ????              141                 DW     ?
                       142    ;
                       143    ;
                       144    ;PACKET STORAGE AREA
                       145    ;
013E (2240             146    BUFFERAREA   DB     MAXSIZE*DEEPPTR DUP(?)
     ??
     )
09FE                   147    BUFFEREND    LABEL  WORD
                       148    ;
```

```
LOC  OBJ                  LINE     SOURCE

----                       149     P6          ENDS
                           150     ;
                           151                 END

ASSEMBLY COMPLETE, NO ERRORS FOUND
```