3-1-1994

A Hybrid voice/text electronic mail system: an application of the integrated services digital network

Andrew McBride

Follow this and additional works at: http://scholarworks.rit.edu/theses

Recommended Citation

This Thesis is brought to you for free and open access by the Thesis/Dissertation Collections at RIT Scholar Works. It has been accepted for inclusion in Theses by an authorized administrator of RIT Scholar Works. For more information, please contact ritscholarworks@rit.edu.
A Hybrid Voice/Text Electronic Mail System: An Application of the Integrated Services Digital Network

by

Andrew McBride

A Thesis Submitted in Partial Fulfillment of the Requirements for the degree of MASTER OF SCIENCE in Computer Engineering

Approved by: Committee Member - Prof. James Heliotis

Committee Member - Prof. Robert Gayvert

Committee Member - Prof. Ronald Matteson

Department Head - Dr. Roy Czernikowski

DEPARTMENT OF COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
ROCHESTER INSTITUTE OF TECHNOLOGY
ROCHESTER, NEW YORK
MARCH 1994
THESIS RELEASE PERMISSION FORM

ROCHESTER INSTITUTE OF TECHNOLOGY
COLLEGE OF ENGINEERING

Title of Thesis: A Hybrid Voice/Text Electronic Mail System:
An Application of the Integrated Services Digital Network

I, Andrew McBride, hereby grant permission to the Wallace Memorial Library of
RIT to reproduce my thesis in whole or in part.

Date: 1 APRIL 1994
Abstract

The objective of this thesis is to present a useful application for the Integrated Services Digital Network (ISDN) that is expected to one day replace the analog phone system in use today. ISDN itself and its continuing evolution are detailed. The system developed as a part of this thesis involved the creation of an inexpensive "phone terminal" that can serve as an ISDN terminal and also as a bridge to a Local Area Network (LAN). The "phone terminal" provides a hybrid electronic mail system that allows the attachment of speech to text within a message. Messages created with this "phone terminal" could theoretically be sent locally using the LAN interface and globally using ISDN to other users with either "phone terminals" or multimedia personal computers. For this project, the two "phone terminals" created were interconnected via an Ethernet and using an 80486 PC to act as a Central Office System. This Central Office System provides speech/message storage for the "phone terminals." It makes use of speech compression techniques to minimize the storage requirements. The speech compression techniques used as well as the field of speech coding in general are discussed.
Preface

This report is divided into five chapters. Chapter 1 gives a brief overview of the thesis. Chapter 2 provides information on real-world areas related to the thesis. Chapter 3 provides information on the theory behind some of the algorithms used in the thesis project. Chapter 4 details the implementation of the thesis project. Chapter 5 contains a summary of the results produced by the project and concluding remarks. Additional information such as diagrams and source code can be found in the appendices.
## Table of Contents

List of Figures .................................................................................. i
List of Tables .................................................................................. ii
Glossary ......................................................................................... iii

### Chapter 1 Overview ..................................................................... 1
1.1 Thesis Concept ........................................................................ 1
1.2 The Project ............................................................................... 3

### Chapter 2 Concepts ..................................................................... 5
2.1 Integrated Services Digital Network ...................................... 5
   2.1.1 Introduction ..................................................................... 5
   2.1.2 Evolution ......................................................................... 6
   2.1.3 Inside ISDN ..................................................................... 10
   2.1.4 Broadband ISDN ........................................................... 12
   2.1.5 Technical Development of ISDN ................................... 14
   2.1.6 ISDN Applications and Services .................................. 18
2.2 Speech Coding .......................................................................... 21

### Chapter 3 Theory ........................................................................ 27
3.1 Adaptive Delta Modulation .................................................... 27
3.2 Adaptive Vector Quantization ............................................... 28
3.3 Linear Predictive Coding ......................................................... 29

### Chapter 4 Implementation ....................................................... 36
4.1 Hardware ................................................................................ 37
   4.1.1 Central Office System Hardware .................................. 37
   4.1.2 Node/Phone-Terminal Hardware .................................. 38
4.2 Software .................................................................................. 39
   4.2.1 MESS System Modes .................................................. 40
      4.2.1.1 The Main Menu .................................................. 41
      4.2.1.2 Text Editing Mode ............................................ 41
      4.2.1.3 Mail Message Processing System ..................... 42
   4.2.2 Node/Central Office Protocol ..................................... 43
   4.2.3 Mail Message Format .................................................. 46
   4.2.4 The Local Message Queue and SBC68K Dynamic Memory Allocation .... 47
   4.2.5 Central Office Message Storage .................................. 48
   4.2.6 Central Office System Speech Storage and Handle Maintenance ....... 50
List of Figures

Figure 1.1 Conceptual ISDN Phone Terminal ....................................................... 3
Figure 2.1 ISDN Access (Percentage of Lines) ....................................................... 9
Figure 2.2 ISDN Reference Configuration ............................................................ 10
Figure 3.1 Predictor Function for Pole System ...................................................... 30
Figure 3.2 Simultaneous Linear Eqs. for Optimal Predictor ................................. 31
Figure 3.3 Hamming Window .............................................................................. 31
Figure 3.4 Simultaneous Linear Eqs. for Autocorrelation Method ........................ 32
Figure 3.5 Levinson Recursion ......................................................................... 33
Figure 4.1 High Level System Diagram ................................................................. 36
Figure 4.2 Node Block Diagram ........................................................................ 38
Figure 4.3 Message Storage Structure ................................................................. 50
Figure 4.4 Data Storage Structure .................................................................... 53
Figure A.1 Screen Display ................................................................................. 78
Figure A.2 System Memory Map - RAM ............................................................. 79
Figure A.3 Modified rom68.a68 Startup Routine ............................................... 91
Figure A.4 u-Law Scheme used by DSC ............................................................... 93
Figure A.5 Graph of u-Law Scheme used by DSC ............................................. 93
Figure A.6 RIFF Chunk Structure ..................................................................... 94
Figure A.7 Special RIFF Chunk Structure ........................................................... 94
Figure A.8 Inverse u-Law Equation .................................................................... 95
Figure A.9 Macros for Format Conversion ......................................................... 96
Figure A.10 uLaw() Routine ............................................................................ 96
Figure A.11 uLawInvert() Routine ..................................................................... 96
Figure A.12 Globals used by fcvto() and sl_approx() Routines .......................... 97
Figure A.13 fcvto() Routine .............................................................................. 97
Figure A.14 sl_approx() Routine ....................................................................... 97
List of Tables

Table 2.1 ETSI-ISDN Coverage ................................................................. 7
Table 3.1 VQ/Run-Length Coding Scheme ............................................... 34
Table 4.1 Protocol Commands/Responses ................................................. 44
Table 4.2 AddressNode (AN) Structure .................................................... 49
Table 4.3 MessageNode (MN) Structure ................................................... 49
Table 4.4 HandleNode (HN) Structure ..................................................... 51
Table 4.5 DataNode (DN) Structure ......................................................... 51
Table 4.6 Interrupt Vectors ................................................................. 54
Table 4.7 DUART Registers ................................................................. 55
Table 5.1 Intelligibility Ratings .............................................................. 68
Table 5.2 Quality Ratings ................................................................. 69
Table 5.3 Utterances .......................................................... 69
Table 5.4 Phrase/Single-Word Compression Results ................................. 70
Table 5.5 Repeated Utterance Compression Results ................................. 70
Table 5.6 Volume Variation Compression Results .................................. 71
Table A.1 Regional Bell Holding Companies Area of Influence .................. 81
Table A.2 Independents Telco's Service Regions .................................... 81
Table A.3 Components of RIFF 'fmt ' Structure ................................... 94
Table A.4 Programmable Interrupt Controllers .................................... 98
Table A.5 Direct Memory Access Controllers ....................................... 98
Table A.6 Allied Telesis 1500 (Am79c960) ......................................... 98
Table A.7 DSC Register ................................................................. 99
Table A.8 LANCE Registers ............................................................ 99
Glossary

ACK
ACKnowledge

ADC
Analog to Digital Converter

ADM
Adaptive Delta Modulation

ADPCM
Adaptive Differential Pulse Coded Modulation

ADSL
Asynchronous Digital Subscriber Line

ANSI
American National Standards Institute

ARQ
Automatic Repeat Request

AT&T
American Telephone and Telegraph

ATM
Asynchronous Transfer Mode

autocorrelation
In statistics, the correlation between a function at two different times

AVQ
Adaptive Vector Quantization

B-ISDN
Broadband Integrated Services Digital Network

BABBLE
Babbling Transmitter (LANCE Interrupt Condition)

BC
Bearer Capability

Bellcore
Bell Communications Research Corporation

BIOS
Basic Input/Output System

BRA
Basic Rate Access

BTRL
British Telecom Research Laboratories

CAD
Computer Aided Design (alternate: Computer Aided Drafting)

CATV
Community Access Television (Cable Television)

CCITT
International Telephone and Telegraph Consultative Committee

CD
Compact Disc

correlation
In statistics, a measure of the dependence between two random variables

CPE
Customer Premise Equipment

DAC
Digital to Analog Converter

DCT
Discrete Cosine Transform

DMA
Direct Memory Access

DSC
Digital Subscriber Controller (Am79c30a)

DSL
Digital Subscriber Line (alternate: Digital Subscriber Loop)

DSP
Digital Signal Processing

DUART
Dual Universal Asynchronous Receiver Transmitter (mc68681)

EPROM
Erasable Programmable Read Only Memory

ergodic
In statistics, a function whose average is the same over time and repeated experiments

ETSI
European Telecommunications Standards Institute

FCC
Federal Communications Commission

FSK
Frequency Shift Keying

FTTC
Fiber-To-The-Curb

FTTH
Fiber-To-The-Home

GER
Gain Filter in Audio Output Path of DSC

HDSL
High bit-rate Digital Subscriber Line

HDTV
High Definition Television

IC
Integrated Circuit

IDN
Integrated Digital Network

IDON
Initialization Done (LANCE Interrupt Condition)

IP
Internetwork Protocol

IRQ
Interrupt Request (alternate: Interrupt Request Handling Routine)

ISA
Industry Standard Architecture

ISDN
Integrated Services Digital Network
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISO</td>
<td>International Standards Organization</td>
</tr>
<tr>
<td>ISPBE</td>
<td>Integrated Services Private Branch Exchange</td>
</tr>
<tr>
<td>ISR</td>
<td>Interrupt Service Routine</td>
</tr>
<tr>
<td>IXC</td>
<td>Interexchange Carrier</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>LANCE</td>
<td>Local Area Network Controller for Ethernet (Am7990)</td>
</tr>
<tr>
<td>LATA</td>
<td>Local Access and Transport Area</td>
</tr>
<tr>
<td>LEC</td>
<td>Local Exchange Carrier</td>
</tr>
<tr>
<td>LPC</td>
<td>Linear Predictive Coding</td>
</tr>
<tr>
<td>MAN</td>
<td>Metropolitan Area Network</td>
</tr>
<tr>
<td>MAP</td>
<td>Main Audio Processor (on Am79c30a)</td>
</tr>
<tr>
<td>MCI</td>
<td>Microwave Communications Incorporated</td>
</tr>
<tr>
<td>mean (1)</td>
<td>In statistics, the average value of a random variable</td>
</tr>
<tr>
<td>mean (2)</td>
<td>In statistics, the expected value of a random function at a given time</td>
</tr>
<tr>
<td>MERMAID</td>
<td>Multimedia Environment for Remote Multiple-Attendee Interactive Decision Making</td>
</tr>
<tr>
<td>MERR</td>
<td>Memory Access Error (LANCE Interrupt Condition)</td>
</tr>
<tr>
<td>MESS</td>
<td>e-Mail with Embedded Speech System</td>
</tr>
<tr>
<td>MFJ</td>
<td>Modified Final Judgment</td>
</tr>
<tr>
<td>MISS</td>
<td>Missed Packet (LANCE Interrupt Condition)</td>
</tr>
<tr>
<td>MPEG</td>
<td>Motion Picture Experts Group</td>
</tr>
<tr>
<td>MSN</td>
<td>Multiple Subscriber Network</td>
</tr>
<tr>
<td>Musicam</td>
<td>Masking-pattern Universal Sub-band Integrated Coding and Multiplexing</td>
</tr>
<tr>
<td>MUX</td>
<td>B-Channel Multiplexer (on Am79c30a)</td>
</tr>
<tr>
<td>N-ISDN</td>
<td>National ISDN</td>
</tr>
<tr>
<td>N-ISDN-n</td>
<td>National ISDN Level n</td>
</tr>
<tr>
<td>NACK</td>
<td>Negative Acknowledge</td>
</tr>
<tr>
<td>NII</td>
<td>National Information Infrastructure</td>
</tr>
<tr>
<td>NIST</td>
<td>National Institute of Standards and Technology</td>
</tr>
<tr>
<td>NIUF</td>
<td>North American ISDN Users' Forum</td>
</tr>
<tr>
<td>NT</td>
<td>Network Termination</td>
</tr>
<tr>
<td>NTT</td>
<td>Nippon Telephone and Telegraph</td>
</tr>
<tr>
<td>OLE</td>
<td>Object Linking and Embedding</td>
</tr>
<tr>
<td>PABX</td>
<td>Private Address Branch Exchange</td>
</tr>
<tr>
<td>PBX</td>
<td>Private Branch Exchange</td>
</tr>
<tr>
<td>PC</td>
<td>Personal Computer</td>
</tr>
<tr>
<td>PCB</td>
<td>Personal Computer Board</td>
</tr>
<tr>
<td>PCM</td>
<td>Pulse Coded Modulation</td>
</tr>
<tr>
<td>PIC</td>
<td>Programmable Interrupt Controller</td>
</tr>
<tr>
<td>PLCC</td>
<td>Plastic Leaded Chip Carrier</td>
</tr>
<tr>
<td>PRA</td>
<td>Primary Rate Access</td>
</tr>
<tr>
<td>PSC</td>
<td>Public Service Commission</td>
</tr>
<tr>
<td>PSPDN</td>
<td>Packet Switched Public Digital Network</td>
</tr>
<tr>
<td>QAM</td>
<td>Quadrature Amplitude Modulation</td>
</tr>
<tr>
<td>R&amp;D</td>
<td>Research and Development</td>
</tr>
<tr>
<td>RAM</td>
<td>Random Access Memory</td>
</tr>
<tr>
<td>RHC</td>
<td>Regional Bell Holding Company</td>
</tr>
<tr>
<td>RINT</td>
<td>Receiver Interrupt (LANCE Interrupt Condition)</td>
</tr>
<tr>
<td>SBC68K</td>
<td>Single Board Computer using the mc68000 microprocessor</td>
</tr>
<tr>
<td>SDH</td>
<td>Synchronous Digital Hierarchy</td>
</tr>
<tr>
<td>SIA</td>
<td>Serial Interface Adapter (Am7992)</td>
</tr>
<tr>
<td>SONET</td>
<td>Synchronous Optical Network</td>
</tr>
<tr>
<td>stationary</td>
<td>In statistics, a random function whose statistics do not change over time</td>
</tr>
<tr>
<td>stochastic</td>
<td>In statistics, a random process</td>
</tr>
<tr>
<td>STR</td>
<td>Secondary Tone Ringer (on Am79c30a)</td>
</tr>
<tr>
<td>SUB</td>
<td>Sub-addressing</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------</td>
<td>------------</td>
</tr>
<tr>
<td>TA</td>
<td>Terminal Adapter</td>
</tr>
<tr>
<td>TE</td>
<td>Terminal Equipment</td>
</tr>
<tr>
<td>TINT</td>
<td>Transmitter Interrupt (LANCE Interrupt Condition)</td>
</tr>
<tr>
<td>TRIP</td>
<td>Transcontinental ISDN Project</td>
</tr>
<tr>
<td>VCI</td>
<td>Virtual Call Identifier</td>
</tr>
<tr>
<td>VLSI</td>
<td>Very Large Scale Integration</td>
</tr>
<tr>
<td>VOCODER</td>
<td>Voice Encoder</td>
</tr>
<tr>
<td>VPI</td>
<td>Virtual Path Identifier</td>
</tr>
<tr>
<td>VQ</td>
<td>Vector Quantization</td>
</tr>
<tr>
<td>WYSIWYG</td>
<td>What-you-see-is-what-you-get</td>
</tr>
</tbody>
</table>
Chapter 1

Overview

Chapter 1 provides an overview of the concept behind this thesis and of the project developed as a part of this thesis. Each of the topics discussed here is presented in more detail in a later section of this document.

1.1 Thesis Concept

The goal of this thesis is to present a useful application for the Integrated Services Digital Network (ISDN) that is expected to one day replace the analog phone system in use today. The application developed is a hybrid electronic mail system that allows the attachment of speech to text within a message.

Although economic realities have kept ISDN technology from rapidly taking its place as the communications network of the future, basic ISDN services are available to those (primarily in industry) willing to pay a premium for it. Today ISDN services in the United States are not perfect implementations of the standard and the process of standardizing the ISDN services offered by the many competing telephone carriers is far from over. In spite of the recent media interest in the "Information Super-Highway" and the National Information Infrastructure agenda put out by the White House, ISDN is no more a household term than eight years ago when telecommunications companies first began making preparations for its coming. The development of applications that will allow ordinary people to take advantage of its promise will help to speed its acceptance once it becomes economically feasible for residential use.

With the explosion of multimedia into the home computer market, it is now possible to merge sound, graphics and text into a single document. A good example of this capability is the Object Linking and Embedding (OLE) standard that is part of the Microsoft Windows operating systems. This allows standard image and sound files to be imported into a text document. The method for representing these can vary depending on the application; however, WYSIWYG displays for images and ICONs for sound are common. Such a document may easily be transmitted over a Local Area Network. Technically, such a message could also be transmitted over the analog phone network. This would be a painfully slow process for a document with non-text of any substance considering that there is a theoretical limit on the order of 28,800
bits/second and real transmission rates are typically much lower. Three things separate the system developed in this project from the above scenario. First, although the use of sound with home computers is popular, no business e-mail system has attempted to provide a specific application for the use of speech as proposed. Second, the addition of ISDN into the picture will provide easy transmission of e-mail not just between local nodes of a LAN, but eventually throughout the country. Third, the development of an inexpensive (as compared to a multimedia personal computer) "Phone Terminal" that can interface to ISDN and to a Local Area Network will allow modern telecommunication to be brought to people without the time or interest to use a personal computer. As an example, a busy newspaper editor could be set up with a simple display terminal attached to his/her telephone. Correspondents would electronically submit stories from throughout the country for review. The editor, while reviewing the story, would point to a specific section he wished to comment on and, using a microphone (or the telephone handset), attach a message. The journalist would then receive back his original with highlighted sections indicating where comments are. After making the appropriate changes the story could be resubmitted electronically. As shown by this example, a "network" would consist of personal computers and inexpensive "Phone Terminals" (depending on the user) both capable of communicating locally over a LAN and globally over ISDN.

The ideal finished product from this project would be a text-book sized black-box that would serve as an interface between a standard phone and terminal and both ISDN and a LAN. (See Figure 1.1.) There would also be a matching Microsoft Windows compatible e-Mail program to complement the "Phone Terminal" based program. All of the required logic for such a system is present on the completed interface boards constructed during this project; however, development of a system of the nature described above would require significant additional work. This includes porting the e-Mail software to other platforms, implementing the driver and hybrid circuit required for the interface to a standard telephone (a stand-alone speaker and microphone are used), and making use of the ISDN I/O channels (2B+D) provided by the ISDN digital subscriber controller (DSC) chip.
1.2 The Project

The "Phone Terminal" developed for this project consists of a Motorola 68000 single board computer, an ISDN/LAN interface board constructed for the project, and a speaker and microphone with related circuitry. A standard VT100 compatible terminal is used.

The choice of the 68000 board introduces a number of limitations. Its 12 MHz clock speed and lack of floating point co-processor make it inadequate for digital signal processing. Additionally, its 128k bytes of memory restricts the amount of uncompressed speech\(^1\) that can be stored on board.

To alleviate the storage problem associated with the 68000 board, a 80486 Personal Computer is interfaced to the Ethernet to serve as a Simulated ISDN Central-Office. Currently, a number of local phone companies provide voice-mail services as options to standard phone service. This service frees individuals from the limitations of tape-based answering devices. It would not be far-fetched to imagine similar services provided by ISDN carriers in the future to enable an inexpensive "Phone Terminal" to have access to the same features as expensive
personal computer based systems. For this project, passing the speech messages to the Simulated Central-Office for storage allows only a handle to each such message to be associated with a document, thus reducing tremendously the on-board storage requirements. This solution has the added benefit that speech compression can be done by the Central-Office computer rather than by the less powerful nodes.

The complete setup developed for this project consists of two functionally equivalent "Phone Terminals" and a Simulated ISDN Central Office. These are connected by thin-wire Ethernet. More "Phone Terminals" could be added to the setup if desired with no changes required to the hardware/software for the system. (A high-level schematic of this setup is shown in Appendix K).

1 Speech on the 68000 board is sampled at 8 kHz by 8 bits per sample. For example, if half of the available memory--or 64k bytes--were dedicated just to storing speech a total of (64k bytes)(1 second/8 k samples)(1 sample/8 bits)(8 bits/1 byte) = 8 seconds of speech could be stored.
Chapter 2 focuses on the real-world state of areas of technology that are addressed in this thesis. Two major areas are addressed: Integrated Services Digital Network (ISDN) and Speech Coding. ISDN is the term for a global project that is bringing point-to-point digital communications to many countries around the world and whose goal is to eventually provide world-wide, transparent digital communications to anywhere telephone access is available and beyond. ISDN will provide numerous services that take advantage of its all digital communications path. The application developed in this project is one example of such a service. Speech Coding refers to any of a number of methods of converting analog signals (speech) into a digital representation that can be manipulated/stored/transmitted as digital data. Speech Coding involves a trade-off between the quality of the speech reproduced, the amount of processing required to code the speech and the amount of digital data needed to store a certain length speech segment. Speech Coding is critical to any system that is forced to divide up limited transmission capability (or storage space) between digitized speech and other digital data. Speech Coding is important in this project to minimize the amount of storage space taken up by stored speech messages.

The first sections of this chapter introduce ISDN and provide information on its continuing development. Applications and services that currently exist or are expected are also presented. The development of broadband digital services, in particular Broadband ISDN (B-ISDN), are also discussed. The final section of this chapter describes commercial uses of speech coding techniques and outlines ongoing research and development in this field. A more technical discussion of the speech coding techniques used in this project can be found in the Chapter 3.

2.1 Integrated Services Digital Network

2.1.1 Introduction

The basic concept behind ISDN is simple: a digital communications network of the magnitude of today's analog telephone network providing a multitude of voice, data, and
multimedia services throughout the world. It is a concept that excites not just engineers and scientists but politicians, journalists, and scores of others who see its vast potential. Journalist and human rights advocate Leonard Sussman proclaims that "The new technologies [ISDN and associated applications] are the conduit for generating vast information power....[The new technologies will] enable questioners to confirm data, verify policies, and cross-check information instantaneously....-widen the horizons of individuals through far greater cultural and educational opportunities." [Ref 27, Pages 7-8] Unfortunately, political, economic, and technical realities make this simple concept a difficult one to implement. In the years that have passed since enthusiasm for ISDN reached its peak in 1984 with a preliminary international agreement on its technical specifications, progress has continued slowly while much of the enthusiasm dissipated. Today ISDN is a reality in the business world, but it will be some time before it is economical enough for residential uses.

ISDN was originally conceived of and named in 1971 by the International Telephone and Telegraph Consulting Committee (referred to by its French initials CCITT). It is this organization, responsible for developing/approving international communications, interfacing, and interworking standards, that continues to lead the way in the development of ISDN. The preliminary ISDN agreement, published by CCITT in 1984, specified several digital channels and two basic access arrangements that these would be configured in. The two most important digital channels are the B and D Channels. [The names for these channels originally came from the terms "Binary" channel (B) and the "Delta" channel (D).] Typically, the B channels with a data rate of 64 kbps are used for data transmission and the D channels with a data rate of 16 kbps (or 64 kbps in special cases) are used for signaling information. The two access arrangements defined by CCITT are Basic Rate Access (BRA) and Primary Rate Access (PRA). Basic Rate Access provides two B channels and one D channel (2B+D) and is intended for individual subscribers. Primary Rate Access provides either 23 B channels in North America or 30 B channels in Europe along with one D channel. This setup is intended for corporate customers with a need for more capability. Updates to the 1984 recommendations have specified a number of H channels that range in capacity from 384 kbps to 135 Mbps and introduced B-ISDN that would have data rates on the order of 150 Mbps and 600 Mbps. The BRA is intended for uses such as voice, data, and facsimile; whereas, B-ISDN data rates will be required for such things as the delivery of High Definition Television (HDTV).

2.1.2 Evolution

Although conceived of in 1971, efforts to develop and deploy ISDN have been underway only since the mid-1980's following the release of CCITT's initial recommendations in 1984.
These efforts have been centered in three areas: the United States, Japan, and Europe (with France and England leading the way). Efforts have been the most intense in Europe followed closely by Japan. The United States has lagged behind primarily because it does not have a centralized, government-controlled phone company. The initial phases of ISDN development did not offer enough economic incentive for the private telecommunications firms in the United States to move ahead rapidly.

In Europe, the deployment of ISDN has been underway since 1987. France first began offering commercial ISDN services (called Numeris) in 1987 and has now progressed to the point that BRA service is available to all business customers. In 1989 a plan was introduced that called for widespread European ISDN coverage by 1994. A set of standards to ensure compatibility was prepared by the European Telecommunications Standards Institute (ETSI). ETSI-ISDN coverage in January 1994 is summarized below:

<table>
<thead>
<tr>
<th>Level of Coverage</th>
<th>Countries</th>
</tr>
</thead>
<tbody>
<tr>
<td>100% of business customers</td>
<td>France/England/Belgium/Denmark/Luxembourg</td>
</tr>
<tr>
<td>80% of business customers</td>
<td>Germany/Portugal</td>
</tr>
<tr>
<td>60% of business customers</td>
<td>Netherlands/Ireland</td>
</tr>
</tbody>
</table>

One of the drivers of ISDN development is how up to data analog switching equipment was when the switch to digital began. For example, ISDN developed rapidly in France, whose phone system was in poor condition in the early 1980's. In contrast, progress was much slower in Germany where the analog system was in good condition.

In Japan, deployment of ISDN started in 1988. Since then, deployment has been aggressive, even extending outside of metropolitan areas in some cases. By the middle of 1991, ISDN service was able in Japan to 80% of the population and included 53,000 lines for nearly 9,000 subscribers. As in the other areas, ISDN in Japan serves almost exclusively corporate customers. Nippon Telephone and Telegraph (NTT), Japan's public telephone company that was privatized in 1985, plans to begin deploying fiber-optics in 1995 and cover the entire country by 2015. It will begin in central Tokyo and reach the 13 largest cities by 2000. This plan requires the approval of the Ministry of Posts and Telecommunications before it can go forward.

In the United States, as mentioned above, the problem of developing and deploying ISDN is a more complicated one. There are dozens of carriers local, regional, and nationwide interested in ISDN. Switching systems and Customer Premise Equipment (CPE) are provided to
these carriers and their subscribers by countless suppliers. As ISDN has evolved in North America, three main groups have emerged as focal points for its development. These groups are involved in setting standards and coordinating efforts so that nationwide compatible ISDN will become a reality. The first group is the American National Standards Institute (ANSI). Through a group known as committee T1, coordination with CCITT is maintained. Where necessary, committee T1 also develops standards specific to North America. The second group in the US-ISDN picture is Bellcore. This is the name given to the research arm of the regional bell operating companies and is short for Bell Communications Research. Bellcore provides technical advisories on the implementation of various aspects of ISDN. These advisories are not, however, always adhered to by the regional bell operating companies. The third player in the US-ISDN picture is the North American ISDN User's Forum (NIUF). This group, formed in 1988, is composed of industry users of ISDN and chaired by the National Institute of Standards and Technology (NIST). The goal of this group is to ensure that the eventual users of ISDN technology have a strong voice in its implementation. The NIUF now publishes "A Catalog of National ISDN Solutions for Selected NIUF Applications." This catalog describes over 30 applications and the equipment/services needed for each. It also lists 120 products for 60 suppliers. This, along with demonstrations held by NIST and others, help users to sort through the complex list of ISDN services and find applications that they can implement and use today.

The first ISDN trial in the United States took place in 1986 in Oak Brook, Illinois. Isolated trials conducted by individual vendors continued through 1989. In 1990, the first multi-vendor ISDN trials took place and the first deployments of nationwide compatible ISDN took place. In 1991, Bellcore released a proposal for the widespread availability of ISDN. By the middle of 1991, there were an estimated 120,000 BRA and 1,000 PRA commercial lines installed. Up until November of 1992, ISDN in the United States was composed of islands of ISDN not necessarily compatible with each other. On 16 November 1992, the National ISDN (N-ISDN) standard officially went into effect, and 22 N-ISDN Level 1 compatible digital switches were put on line. This occurred during the Transcontinental ISDN Project 1992 (TRIP 92). This was a large scale ISDN demonstration involving 149 North American and several international locations. Within six months of TRIP 92, two-hundred N-ISDN Level 1 compatible digital switches were in operation. There are three levels to the National ISDN standard. Level 1 is the standard protocols for BRA ISDN; Level 2 is the standard protocols for PRA ISDN; Level 3 provides for uniformity of services and standard frame relay. Currently, N-ISDN Level 1 is the only one that has been completed and is in use. Today, the transition from isolated islands of ISDN to a seamless nationwide interface is far from complete. Many carriers have a large investment in non-standard ISDN and are slow to change. The following chart shows the current and projected progress of the seven regional bell holding companies in their respective regions.
Ameritech is omitted in the 1995 columns because they have not announced their plans as of yet. SW Bell is omitted from the N-ISDN columns because they have not reported any N-ISDN compatibility data. (Remember that in addition to these carriers, there are many independent carriers providing local telephone service in the United States each progressing differently, if at all, on ISDN development and standardization.)

![Graph showing ISDN Access (Percentage of Lines)](image)

Figure 2.1 ISDN Access (Percentage of Lines)

There are still a number of problems with ISDN independent of N-ISDN Level 1 compatibility. First, the cost of ISDN equipment is still prohibitive for companies without a specific need that ISDN can fill. Both the Northern Telecom DMS-100 Private Branch Exchange (PBX) and the AT&T 5ESS PBX accept BRA-ISDN line cards that can be swapped for analog line cards; however, the cost of the ISDN cards are nearly five times as much. In addition, either a terminal adapter (costing from $1000-1500 each), an ISDN phone (that run from $650-850), or an ISDN PC Interface card (costing over $1000) is required. Costs for installation of a single ISDN line are similarly expensive. For example, Rochester Telephone's ISDN services, named Versatel, costs around $150 for installation, and $25 per month plus usage fees depending on the particular services requested. Versatel telephones cost between $300 and $1200. Second,
for small telephone companies the cost of upgrading their Central Office is too high. An ISDN upgrade for an AT&T digital switching system costs on the order of $700k. An ISDN upgrade for a similar Northern Telecom system costs around $500k. Worse yet, old Central Offices must be replaced entirely for ISDN. Third, long distance charges for ISDN calls cost about 30% more (AT&T rates) than analog voice calls. Fourth, ISDN does not currently offer any benefits for these voice calls that would justify the premium cost. In fact, power must be supplied from the local source, costing electricity, and only one person per line can use an ISDN phone - extension phones, for multiple parties on the same line, are not available. All of these problems should decrease or vanish over time; however, at present they slow the acceptance of ISDN.

Several events in the past few years have helped build momentum for ISDN's development. Products and services based on CCITT's 1988 updated ISDN recommendations have now been completed and are beginning to appear in larger quantities. Frame Relay (ISDN's version of packet switching -- which will be discussed in later sections) has gained acceptance as a replacement for x.25 packet switching. CCITT released an interim ISDN recommendation in 1990 that provided more detail on frame relay and introduced the increasingly talked about B-ISDN. This coupled with recent moves in the United States to remove the barriers between the Cable TV and Telephone industries has brought renewed public interest in the "information super highway." Although no one can predict when personal ISDN will become common-place, continuing advances in VLSI and Digital Signal Processing Technology (discussed in the section on Technical Development of ISDN) are making the economic barriers blocking the transition to ISDN less formidable.

2.1.3 Inside ISDN

In the previous section a history of ISDN and its evolution was presented. Here a more detailed description of what exactly makes up a BRA or PRA ISDN interface is presented. A number of common terms are used in describing a typical ISDN setup. These are shown below in the CCITT "Reference Configuration."

```
    S
 TE1 ----|----- T ---- U
    R     NT2  NT1 ...
 TE2 ---- TA ----
```

Figure 2.2 ISDN Reference Configuration
In the reference diagram, TE1 and TE2 each represent Terminal Equipment. TE1 is used to refer to an ISDN terminal while TE2 is used to refer to any non-ISDN terminal. The block TA represents a Terminal Adapter that is used to allow a non-ISDN terminal to interface to ISDN. An interface at point S has come to be called an S-bus interface. (This terminology is generally seen in reference to a [S-bus] card for a personal computer.) NT1 and NT2 represent Network Termination equipment. NT1 is a standard device that provides physical and electromagnetic termination for the network. (It is also the legally established dividing point between carrier equipment and customer equipment.) NT2 is an optional block that represents customer equipment such as a PABX (Private Address Branch Exchange) or a LAN. Point T is the interface between such equipment and the NT1. Point U is used to refer to the transmission line.

The S-bus is the key interface for users of ISDN. It is the equivalent of the modular jack in the analog phone system. The S-bus is a four wire (a Transmit pair and a Receive pair) bus whose data rate is 192 kbps. This data rate is derived from the BRA 2B+D configuration as follows: each B channel occupies 64 kbps, the D channel 16 kbps, and 48 kbps is devoted to bus management functions such as frame synchronization and collision avoidance. Part of this bus management includes an Echo Channel (E) that is simply a reflection of the D channel used for collision detection. In a BRA configuration two different modes are allowed. In Point-to-Point mode, one terminal is directly connected up to 1 km from NT2/NT1. In MultiPoint mode, up to eight terminals are connected in a bus configuration. Bus length is limited to approximately 200 m. In a PRA configuration, only point-to-point mode is allowed. In the PRA case, MultiPoint mode is not necessary because customer equipment such as a PABX typically is used to provide that functionality.

Today's analog telephone network supplies the power for customer telephone equipment. This enables phone service to often be maintained even under emergency situations where local power is lost. ISDN equipment is, on the other hand, powered by the local source under normal conditions. This setup has several advantages (mostly in cost to the phone company); however, the issue of communications in emergency situations must be addressed. This is done, in a ISDN network, by enabling the network to provide minimal power (a maximum of 420 mW) to allow the operation of a digital phone under emergency conditions. This provision could require additional wires if the ISDN service is to be provided using fiber optic cables.

The addressing of terminals under ISDN is not as simple as the assigning of telephone numbers today. Multiple terminals (representing possibly a wide variety of equipment from digital phone to fax machines and computers) can be connected to a single ISDN line. The more complex terminals may have multiple processes that perform different functions. For the foreseeable future an interface to the analog telephone world must be permitted. To deal with all
of the possibilities, a number of elements are associated with each terminal to control its addressing. The first of these is a Multiple Subscriber Number (MSN). This is used as follows: The last digit(s) of the public telephone number are broadcast on the S-bus to all terminals so that only the one with a matching MSN responds. The second method is called Sub-addressing (SUB). This refers to transparent (to the users) information that is passed from the calling terminal to the called terminal(s). This allows selection of a particular terminal and even a particular process on that terminal. (Although multiple processes are generally associated with a computer workstation, in this case it could mean something as simple as providing for distinctive rings on a shared digital phone.) The third piece of addressing information associated with each terminal is its Bearer Capability (BC). This is used in cases of interconnection between networks to identify the type of traffic expected. Examples of this are BC = speech, BC = 3.1 kHz Audio (analog modem traffic), and BC = 64 kbps, Unrestricted (normal ISDN traffic).

2.1.4 Broadband ISDN

In 1990, Broadband-ISDN (B-ISDN) was introduced as a part of CCITT's interim ISDN recommendations. B-ISDN is defined as the provision of channels with data rates above 64 kbps. The need for B-ISDN (as opposed to PRA setups that provide multiple 64 kbps channels) is due to the fact that simple concatenation of B channels into a higher data rate channel is not practical. The delay on a given 64 kbps channel is not standard and can vary greatly between B channels. Delays in the Integrated Digital Network (IDN) may be minor ones due to switching stages or significant ones due to satellite links. [Integrated Digital Network is the name given to the network that carries telephony between central offices.] In the IDN, it is possible for one B channel in a PRA setup to be routed via a completely terrestrial link and another via a satellite link or multiple satellite links. This results in delays that can differ by as much as .25 seconds. (Because of concerns of creating too noticeable a delay--in voice conversations--a maximum of two satellite hops per link is observed.)

There are two methods of providing B-ISDN data rates by using special techniques to concatenate multiple B channels and to avoid the effects of delay. The first of these is referred to as the terminal solution. It involves developing special B-ISDN terminals that have the ability to investigate the delay in each channel during call setup. This known delay can then be compensated for during the concatenation process. The second solution is known as the network solution. This solution requires that the central office exchange processors ensure that all of the channels of a multi-channel transfer are within a single multiplex channel, and therefore follow a common path. This solution is limited by the design of current systems which cannot guarantee the availability of an N x 64 kbps channel within a single multiplex channel. The
probability of the network being unable to provide such a channel becomes significant above \( N = 6 \); therefore, 384 kbps (6 x 64 kbps) is the maximum data rate typically provided in this manner.

The goals set for B-ISDN go far beyond the data rates achievable within the framework of today's network. In its initial B-ISDN proposal, CCITT defined a number of \( H \) channels that were to compose B-ISDN. These are as follows: \( H_0 \) - 384 kbps; \( H_1 \) - 1.536 Mbps, \( H_2 \) - 1.920 Mbps; \( H_3 \) - 34 Mbps, \( H_4 \) - 55 Mbps; and \( H_4 \) - 135 Mbps. Since that time B-ISDN has a momentum as preparation for its realization take place. The three data rates that are now considered key to B-ISDN are 155 Mbps, 622 Mbps, and 2.4 Gbps. These data rates reach the levels that will be required for the provision of future video services such as HDTV.

The biggest problem facing the developers of B-ISDN besides simply achieving the switching and processing speeds required by such high data rates is providing a means of multiplexing various ISDN and B-ISDN services (with very different data rates) together on a single channel. There are two techniques that are being used in the development of B-ISDN to solve this problem. The first is called Synchronous Digital Hierarchy (SDH). It is based on the Synchronous Optical NETwork (SONET) that is being developed in the United States to provide high data rate telecommunications services over fiber optic networks. The goal of SDH/SONET is to allow access to lower data rate channels without requiring the entire signal to be demultiplexed. This is done by creating virtual containers that consist of both the data to be transmitted and path overhead. These virtual containers are then interleaved in a systematic manner. The result of this complex process is that direct access to individual channels is now possible. A second, much simpler, technique for addressing the same problem is Asynchronous Transfer Mode (ATM). This system uses a combination of packet-switching techniques and asynchronous time division multiplexing. Data is divided into short, fixed length frames called cells. Each cell has a 5 byte header and a 48 byte data field. The header contains a virtual path identifier (VPI) and a virtual call identifier (VCI). Once an ATM virtual connection has been established the VPI is used by high speed switches (using hardware based routing tables to achieve the speed required by B-ISDN data rates) to route cells to a destination node. The VCI is then used to route cells to a destination terminal. The term virtual connection is used to indicate that no transmission capacity is used by the connection being in place but only when cells are actually being transmitted. Under ATM cells belonging to an individual channel can easily be identified and separated out as was the case with SDH. Due to its simplicity and greater flexibility, ATM is becoming the accepted standard for most future broadband networks; however, because ATM and SDH interworking can be easily achieved it is expected that both will play a role in the future of B-ISDN.

B-ISDN is now typically described as a combination of certain quantities of \( B \), \( D \), and \( H \) channels depending on the intended usage. The \( H_4 \) channel for example matches the data rate
requirements projected for future video services. Some of the applications foreseen for B-ISDN include videophones, color facsimile, video conferencing, cable TV and HDTV distribution, and High Fidelity audio transmission. (See also section 3.1.6 on ISDN applications.) There is currently much effort being put into the development of B-ISDN and experts have predicted that it will become commercially available (with limited access and capabilities) during the late 1990's.

2.1.5 Technical Development of ISDN

The development of ISDN is based in large part on pre-existing telephone network systems. Its continued evolution depends in large part on interworking with existing data communications networks and making use of existing equipment as ISDN is phased in. This section describes the features of the telephone network system from which ISDN was derived, details efforts to provide interworking between ISDN and other networks, and provides an overview of some of the techniques that will be used to decrease the cost of phasing in ISDN.

ISDN came about basically as an extension of the Integrated Digital Network (IDN) used by regional and long-distance telephone carriers. This network carries voice and data between central offices at 64 kbps. The interface between the central office and a subscriber is an analog local loop designed for bi-directional transmission of a 0-4 kHz signal. At the central office, each call consists of separate incoming and outgoing signals that are digitized at 64 kbps. All of the 64 kbps data channels are multiplexed together into a number of T1 channels. A T1 channel is referred to as a Primary Rate Multiplex. In North America it consists of 24 channels (1.544 Mbps) and in Europe it is composed of 32 channels (2.048 Mbps). The North American system currently embeds signaling within the data channels while the European system uses two of its 32 channels for signaling information. ISDN can be thought of as an extension to the IDN in that it simply extends the digital network to the subscriber. (Note that the bit-stealing signaling method used in North America cannot be used in an ISDN system as it was in a system that simply carried digitized 4 kHz audio.) The channel data rate of 64 kbps is preserved as the B channel data rate. BRA can be simply thought of as two phone lines. PRA data rates correspond to the primary rate multiplex: 23B+D in North America and 30B+D in Europe.

In the section on Speech Coding techniques, you will see that 64 kbps is no longer required for the transmission of telephone quality speech. The continued use of this data rate on the IDN (while it would seem economical for carriers to use a 32 kbps or better encoding scheme and essentially double their capacity) is required because Speech Coding techniques do not, in general, preserve facsimile and modem signals. Analog data transmission now accounts for a large portion of the traffic on the public telephone network. Although some speech coding
algorithms have been developed that preserve certain, specific modem signals, no general purpose algorithm exists that could be safely used on the IDN.

The major obstacle to making the extension from the IDN to ISDN is economics. In the early days of ISDN it was believed that fiber optic cables would be required to support the data rate needed for basic rate access ISDN. The cost of replacing the over 100 million individual copper-wire pairs in the United States today with fiber-optic cable is overwhelming. Currently, the incentive for phone companies to make such an investment in minimal. It was this impasse that, until recently, threatened the future of ISDN in the United States.

Fortunately, advances in VLSI and Digital Signal Processing (DSP) have combined to provide methods to make use of much of the existing telephone wiring to provide ISDN service. This allows carriers to offer ISDN service without as much of an investment. They can build demand for ISDN and other data services while gradually phasing-in the fiber-optic cabling that will support the higher data rate services. Of the existing copper-pair wiring in the United States, approximately 25% are equipped with loading coils that eliminate it from consideration for use as a Digital Subscriber Loop (DSL). The majority of the remaining wiring was installed following Carrier Service Area design rules that specify wire gauge and maximum lengths. All of the wiring is this category has the possibility of serving as a Digital Subscriber Loop. A number of problems must be overcome to use local-loop pairs as a DSL. These difficulties arise from the fact that no two wire pairs has the same characteristics and that the path from the central office to the customer is rarely a simple pair of wires. Typically, different gauge wires are spliced together resulting in impedance mismatches that distort higher frequency signals. In addition, bi-directional transmission on a single wire pair causes echoes (reflections) to be added to the signal. (This causes little problem for analog voice conversations but can wreck havoc on a digital data stream.) Finally, although the twisted pair wiring that is employed minimizes signal coupling and reduces stray noise signals, the balance is imperfect and the usable bandwidth is limited by both crosstalk and noise. In the past these problems seemed insurmountable; however, adaptive digital signal processors have reached the point where they can adapt in real-time to individual wire pairs, perform echo cancellation on the incoming signal based on what the outgoing signal looks like, and reshape and separate pulses that have been distorted and overlapped.

The first commercial VLSI chips for BRA ISDN came out in 1990. They demonstrated that the necessary line conditioning for using copper pair as a DSL could be provided in a compact, affordable package. It is now possible to provide bi-directional 144 kbps communications (for BRA 2B+D) over unloaded copper pair wires for distances of at least 3 miles. Given that businesses are typically expected to be within 2-3 miles of either a central office or a remote terminal, the deployment of ISDN has become less of a problem.
Two other techniques for proving higher data rates over existing wiring are under study. The first is called High-Bit-Rate Digital Subscriber Line (HDSL). It is intended as a transparent replacement for leased 1.544 Mbps T1 lines currently used by corporate subscribers. The biggest disadvantage of the existing lines is that they require frequent repeaters to maintain the signal - a costly and time consuming installation problem. The new HDSL would use advanced VLSI/DSP to provide the same capability over the same four wires but without any repeaters. The reduction of crosstalk and the use of echo canceling allow each pair to carry 784 kbps bi-directionally. Although this technology is still in the trial stage, it is hoped that it will be available to serve as a transition technology until fiber-optic cable installations become more widely available. A second technique, aimed more at residential applications such as the delivery of video services and for high speed retrieval of multimedia information, is called Asymmetrical Digital Subscriber Line (ADSL). Currently ADSL is in the study/trial phase. Its intent is to provide uni-directional high data rate (initially 1.544 Mbps) service with a low data rate control channel back to the central office. By restricting the system to operating in a uni-directional manner, a number of problems such as echo and crosstalk are reduced or eliminated. This will allow the service to be provided on existing copper pair wiring without interfering with the analog telephone channel also carried on the pair. In spite of continuing advances in video compression techniques it is expected that 3-4 Mbps will be necessary to carry true TV quality signals; however, a second generation of ADSL has already been proposed that would meet this requirement. All of the techniques for using twisted pair copper wiring to provide data services are considered transitional until the arrival of fiber-optics; however, economic realities may make that transitional period a long one for many people who will receive service only because of these techniques and others that continued advances in VLSI/DSP bring about.

The ability to promote ISDN by providing transitional services using existing wiring and equipment will prove vital to its continued development. Equally important is the capability of ISDN to work with in-place data networks such as Local Area Networks (LANs), Metropolitan Area Networks (MANs), and Packet-Switched Public Digital Networks (PSPDNs). A great deal of effort is underway to insure not only interworking between ISDN and the networks above, but also between ISDN and the coming B-ISDN. Such interworking will allow users of different networks to communicate, users of non-global networks to connect to other non-global networks, and users of networks under development to overcome initial limitations by using an existing network.

The differences between ISDN and B-ISDN technology are vast. Although B-ISDN will eventually support all ISDN services, the ability to interconnect the two will be important, especially during the development of B-ISDN. The leading solution for providing interworking between ISDN data channels and B-ISDN is to develop combined broadband/narrowband switch
modules. To allow users of ISDN's Frame Relay services to communicate with B-ISDN users, an additional InterWork Unit must be added to the B-ISDN switching setup. [CCITT is expected to make a recommendation on both of these shortly.]

ISDN support for packet-switching has been well defined by CCITT. Three different methods are possible. The first two insure compatibility with PSPDNs that use X.25 - a virtual circuit/packet-switching protocol established by CCITT in 1976. The first and simplest method is to use an ISDN terminal connected to an ISDN line and request that ISDN make the connection to the PSPDN. The second method is similar but uses an X.25 terminal connected to ISDN through a terminal adapter. The third method is to use an all ISDN network (the source and destination are both ISDN terminals). This method, referred to as "ISDN Virtual Circuit Bearer Services," allows more flexibility and services that are not possible when an X.25 interface is required. CCITT has defined two new services that fall under this category: Frame-Switching, a technique that uses an acknowledgment protocol; and Frame-Relay, a technique that does not acknowledge packets.

The interworking of B-ISDN with Local Area Networks (LANs) and Metropolitan Area Networks (MANs) has recently attracted much interest. A CCITT standard to address interfacing IEEE 802.6 compatible MANs and B-ISDN in under development. A server/bridge will be required to perform the protocol translation between the two systems. Header information will also have to be processed to establish virtual connections within B-ISDN. The required translations are not difficult; however, the best methods for establishing the virtual connections are still being studied. For several reasons, the interworking of B-ISDN with LANs is a more difficult problem. There are many different types of LANs in use today. Four categories have been defined for grouping the LANs of today and the future. These are Low/Medium speed (10-20 Mbps), High speed (100-140 Mbps), Supercomputer/Short Distance (~1 Gbps) and Ultragigabit (~1 Tbps). ISDN planners must also take into account the fact that although current LANs all use connectionless protocols, this may change in the future with the shift towards delivery of video and multimedia services. The problem has been divided into two main areas that are now being looked at: the connection of like LANs and the connection of dissimilar LANs. The first will require only a bridge to B-ISDN that maintains local and remote address tables. The second will require the use of Internetwork Protocol (IP) addresses and routers with semi-permanent connections to B-ISDN servers. ISDN BRA and PRA services are currently being used to interconnect LANs in proprietary setups but as of yet CCITT has not made recommendations on LAN interconnectivity with either ISDN or B-ISDN.

17
2.1.6 ISDN Applications and Services

The purpose of this section is to describe some of the applications that ISDN is being used for and to look at the types of applications and services that will take advantage of the power and flexibility offered by ISDN and B-ISDN. Current data applications will all benefit from ISDN. Some such as electronic mail and file transfer will simply be made easier and faster by the improved data rate. Others such as facsimile and videotex will be changed dramatically to a degree that new uses will become possible. Over basic rate ISDN, facsimile speeds will approach those of a slow photocopier and using B-ISDN concepts such as color facsimile will become reality. The change from today's 1200 bps text retrieval services to ISDN based services operating at 64 kbps and beyond will enable instant page creation, browsing, and the insertion of graphics. In addition to these services, others that are not currently possible will come into being. True videophones, high quality speech, and CD quality music transmission will be realizable. Services such as caller-id that recognizes the caller and retrieves and displays his/her medical, school, or customer record before the phone is answered will be easily implemented.

In France, where deployment of ISDN began in 1987 and in now available throughout the country, ISDN is used for a number of varied purposes. The French system, called Numeris, provides users with either BRA or PRA access. Interfacing options include S-bus interface cards for PCs, ISDN phone sets, Terminal Adapters, and Integrated Services Private Branch Exchanges (ISPBES). Services available include calling line ID and restriction, call forwarding, user to user messaging, terminal portability and subaddressing (described in the Inside ISDN section). ISDN applications are in use by a variety of types of companies. Both real estate companies and medical facilities use Numeris to provide access to remote image databases allowing information to be accessed in minutes instead of days. Security firms also use Numeris for image transfer to low resolution video surveillance from remote sites to be monitored at a central location. The configuration used for these image applications is typically a PC equipped with an S-bus interface card. Document transfer applications are used by oil, pharmaceutical, and engineering companies as well as by the public library system. A number of oil and pharmaceutical companies such as Shell and Glaxo use Numeris for commercial document transfer. Hewlett Packard uses Numeris for equipment order tracking. The French public library system uses Numeris to transfer student records between locations. Other applications that use Numeris include CAD file transfers for PCB manufacturing, remote teaching/training, the gathering of point of sale information and banking transfers.

An example of a system using ISDN today but truly designed for and awaiting B-ISDN is NEC's MERMAID. This system, the Multimedia Environment for Remote Multiple-Attendee
Interactive Decision-making, interconnects NEC-Japan and NEC-USA. This system is designed to work over ISDN, leased data lines, or satellite networks. Planning for its evolution to B-ISDN is well underway. It uses Unix-based workstations and provides simultaneous interchange of voice and video. The transfer and processing of text, graphics, images, and handwriting are also possible. Currently the system is setup between NEC research labs in Osaka and Kawasaki (and several other sites in Japan) and NEC-USA in Princeton, New Jersey. The nodes in Japan are interconnected by Japanese ISDN and use two B channels. From Japan they are connected via an international ISDN connection to AT&T's Accunet which provides two 56 kbps channels to the NEC-USA node.

A number of ideas along the lines of NEC's MERMAID system have been proposed as possibilities for ISDN/B-ISDN. One is the idea of a multipoint conference. This extends the idea of a video/audio/data conferencing capability by adding control signals that allow a chairman to control the proceedings. A different concept along these lines is that of a computer conference. In this scenario, two people jointly run an application that appears on each of their computer screens. Interaction is controlled by data passed over a B channel and audio communications is provided over a second B channel. The uses for this concept include on-site training, data-entry/verification, and remote maintenance/diagnostics.

A good amount of planning (and attempts to provide standardization) for future ISDN applications is underway. One area for future ISDN applications is quality speech, and the coding of speech for multimedia applications. Currently speech is coded using a logarithmic coding scheme (u-Law coding is the United States standard; A-Law coding is the European standard) into 64 kbps. In this scheme less than 4 kHz of audio bandwidth is preserved. Speech signals can actually cover a range of up to 7 kHz (only fricatives—hissing sounds such as s and f can produce—actually approach this maximum level); therefore, that is the limit set for high quality speech. CCITT has defined two standards for advanced speech coding: 3.1 kHz speech coded into 32 kbps with no subjective difference versus standard 64 kbps coding, and 7 kHz speech coded into 64 kbps. As you will see in the section on speech coding, the bit rates for coding speech can be reduced significantly beyond these standards depending on the quality of output required and the processing power available. The uses for high quality speech are clearer telephone service, transmission of news and sporting event broadcasts and the like. The coding of speech at 32 kbps and below allows the simultaneous transfer of either multiple conversations or multimedia information (including speech) over a single B channel. In the latter case, the more the data rate of the speech can be reduced the more additional information can be transferred along with it.

A similar area of research is music coding. Compact Disc quality audio covers the entire theoretical range of human hearing from 20 Hz to 20 kHz. This requires 16 bits/sample at a
sampling rate of 44.1 kHz. The result is a data rate of 705 kbps. Additionally, the data rate is doubled for a stereo signal. By taking advantage of certain characteristics of human hearing (the techniques involved will be discussed further in the section on Speech Coding) techniques have been developed that allow significant reductions in the bit rate without a noticeable change in quality. The current standard from the Motion Pictures Experts Group (MPEG, a section of the International Standards Organization [ISO]) produces a non-constant bit rate depending on the input signal. Using this method on a mono signal, a bit rate of 110 kbps is sufficient for most cases. Additional specifications have been developed for reducing the signal to 64 kbps with as little distortion as possible. Subjective tests of the system using 128 kbps (the equivalent of two B channels) have shown the output to be indistinguishable from CD quality.

A key area for ISDN/B-ISDN is facsimile. This technology has been around since the 1960's but only recently has become an indispensable business tool. CCITT released its first facsimile recommendations in 1968. These defined Group 1 facsimile. Group 1 facsimile is an analog FM transmission system for sending one gray scale page in six minutes over the telephone network. Group 2 recommendations, released in 1976, preserved the 3.85 lines/mm resolution of Group 1 but offered transmission times of three minutes per page. Group 3 standards proposed in 1980 created the first digital form of facsimile. In order to allow compression using a run-length coding scheme, gray scale was abandoned in favor of black and white. Vertical resolutions of 3.85 lines/mm and 7.7 lines/mm and a horizontal resolution of 8.03 pel/mm were adopted. Three transmission rates are defined: 2400 bps, 4800 bps, and 9600 bps. With the compression scheme yielding typically 7:1 to 12:1, transmission times of 20-30 seconds per page are possible at 9600 bps. The Group 4 standard released in 1984 covers facsimile, teletex, and telex services under ISDN. The system is fully digital and includes an error correction scheme and a compression scheme that uses two-dimensional run-length coding with Huffman codes. Resolutions from 200-400 lpi (roughly 7.87-15.75 lines/mm) are defined. Automatic fallback to communicate with Group 3 machines is supported. Because of the inclusion of teletex/telex services, three classes of Group 4 facsimile are defined. Class 1 indicates basic facsimile only; Class 2 indicates facsimile and teletex receive; Class 3 indicates facsimile and teletex send and receive. Both classes 2 and 3 include support for a mixed-mode that allows facsimile and character blocks to be combined on a single page. Higher-resolution and color facsimile are concepts that have been mentioned as B-ISDN applications, but no specifics exist as of yet.

An area where ISDN/B-ISDN hold great promise is photographic quality image transmission. Applications in this area include photographic videotex, still pictures for teleconferencing, slow-scan security pictures, medical images, newspaper pictures, and satellite weather maps. A studio quality photograph displayed on a typical 512 line x 512 pixel videotex
display requires 262k pixels. Each pixel represents one of 16 million colors. This requires 24 bits per pixel (8 bits x 3 primaries) which means that 786 kbytes is needed to represent a single picture. Using an ISDN-based interactive service this would take an unacceptable amount of time to transmit. There are a number of common image compression techniques that can be used to improve the transmission time: predictive coding, block coding, discrete cosine transform, and vector quantization. The current standard from ISO uses a discrete cosine transform (DCT) method. This technique is adaptable and can produce bit rates from 0.1 bits/pixel to 12 bits/pixel depending on the desired image quality. It also supports a scheme called progressive coding. Progressive coding is useful in cases where an interactive remote search is taking place. The scheme quickly produces an initial crude image, then improves on it sequentially until the final image is displayed. This allows a user to determine if he/she has found a desired image or to continue scrolling if not (terminating transmission of the partial image). In subjective testing using the ISO DCT technique, images of poor quality resulted from bit rates below 0.5 bits/pixel, good quality images resulted from bit rates on the order of 1.0 bit/pixel, and images transmitted using 2.0 bits/pixel were indistinguishable from the original. Images of the quality required for medical purposes required 8.0 bits/pixel to produce. Transmission times over an ISDN B channel for a 512 x 512 picture range from less than 2 seconds for a quick, poor quality image to around 4 seconds for a good quality image, 8 seconds for a picture indistinguishable from the original, and 32 seconds for a "medical" quality image.

The range of applications and services possible under ISDN and B-ISDN is immense. Many of the concepts/techniques presented in this section will play a role in the development of future applications for ISDN/B-ISDN. The continuing research in the fields of speech and video coding promise shrinking requirements for transmission while B-ISDN promises to provide increasing capacities. This combination makes it likely that even those applications that are dreams today will one day be dwarfed by what is possible.

2.2 Speech Coding

The purpose of this section is to provide information on the state of the art in Speech Coding, how it is used in real-world applications, and where current research and development efforts are headed. Little technical detail on specific algorithms will be presented in this section. For a detailed description of the algorithms used in this project refer to the Implementation section.

Digital audio data is generated from analog audio signals using a technique known as Pulse Coded Modulation (PCM). An analog to digital converter (ADC) is used to sample an audio waveform thousands of times per second. Based on the amplitude of the waveform at a
given instant, the ADC selects the closest digital representation. The sampling rate must be
twice the highest frequency that will be encoded. Telephone quality speech consists of
frequencies of less than 4 kHz; therefore, a sampling rate of 8 kHz is standard for digital
telephone equipment. Human hearing extends as high as 20-22 kHz so sampling rates up to the
44.1 kHz used for Compact Discs are common in digital audio systems. The number of bits per
sample used in the digital representation for PCM range from 7/8 for telephone systems to 16 for
Compact Disc quality systems. The 64 kbps data rate found in the IDN and in ISDN represents 8
bits per sample at 8 kHz. At its destination the digital data is played back through a digital to
analog converter (DAC) to generate an audio waveform.

The term Speech Coding refers to both PCM and many digital signal processing
techniques that are used in addition to PCM to reduce the amount of data that must be
stored/transmitted in order to reproduce the original. Selecting a method of speech coding for a
given system involves a trade-off between the quality and intelligibility of the speech at the
destination, the processing power required to code/decode the speech, and the amount of data
required to represent a speech sample. The quality of a speech sample refers to how pleasant
(subjectively) it sounds while the intelligibility refers to how accurately (objectively) the speech is
interpreted.

One of the simplest forms of speech coding is to use a logarithmic scaling of the digital
values. Two common examples of this technique are u-Law and A-Law coding. (These are
discussed further in Appendix F.) A technique widely used in modern speech coding systems is
Linear Prediction. (See Chapter 4 for one example of its use.) A Linear Prediction Coding (LPC)
system uses a model to predict values of the waveform based on past values. A basic LPC
system works as follows: the incoming waveform is divided into short segments over which the
signal can be assumed to be fairly consistent; a model (or prediction coefficients) is computed
for each of these segments; the predicted waveform is subtracted from the original waveform
creating a residual (or error) signal; the residual signal and the prediction coefficients are coded
and transmitted in place of the original waveform; at the destination the process is undone
resulting in a close approximation of the original. Depending on how the residual and prediction
coefficients are coded a large amount of compression can be achieved using this method. A
number of different systems that are based on LPC techniques are presented later.

Another technique used in speech coding systems (but general to any type of data
compression) is Vector Quantization (VQ). A reduction in the data to be transmitted is achieved
by grouping samples into vectors that are represented by special codes. Based on common
signals, a codebook with the 2k most likely vectors is generated. During transmission each
incoming vector is matched to the closest vector in the codebook and the k-bit address of the
vector is sent instead of sending the entire vector. The size of the codebook, the size of the

vectors, and the algorithm for selecting the closest vector for non-matches are selected based on the quality/processing-time/compression trade-off mentioned above.

One other class of techniques for speech coding is filter-bank or sub-band coding. Here the frequency range of the incoming signal is divided up into channels each of which is processed separately. Note that the signal at the destination will sound like the original (to the human ear) but with some of these techniques it will not match the original waveform. (The term VOCODER for a device that performs voice encoding--as opposed to waveform encoding--is given to speech coding systems that make no attempt to preserve the original waveform.) There are several variations of this technique that select different types of information to preserve. The two items used in most all VOCODERS are pitch and voicing. Pitch is critical to producing an output that sound like the original speaker. Voicing information specifies whether a specific sample of speech was produced using the vocal cords (voiced) or produced using forced air (unvoiced) - such as the hissing sounds that s and f often cause. Several systems using voice encoding techniques are described below.

The most common speech coding systems in use today are Adaptive Differential Pulse Coded Modulation (ADPCM) systems that achieve compression rates of 2:1 and 4:1 with no noticeable change in quality/intelligibility. These are found in digital answering machines, PC sound cards, and other inexpensive applications. ADPCM uses straight-forward LPC-based techniques: the term Adaptive refers to the fact that the prediction coefficients are recalculated for each speech segment; the term Differential refers to the fact the residual signal actually transmitted is generated from the difference between the actual value and a prediction based on previous values.

Advanced systems, often developed for specific data communications applications, have reached the point where they can code telephone quality speech (originally 64 kbps) into 600 bps and below while maintaining reasonable intelligibility. The majority of these systems require specialized DSP processor(s) to operate in real-time. The following list describes several low and very-low bit rate systems and the data rates they can achieve. Code-Excited Linear Prediction (CELP) systems use a VQ-like codebook to compress the LPC residual. Current systems can reach 3000 bps and preserve a natural sounding signal. Multipulse systems eliminate the residual, but try to match it with a series of carefully generated pulses. These systems can also reach 3000 bps and retain naturalness; however, they are computationally intensive due to the pulse selection/matching process. An LPC VOCODER combines two techniques by sending the LPC prediction coefficients (but not the residual) and pitch and voicing information. Data rates of 1200-2400 bps are achievable with these systems; however, much of the naturalness is lost. The combination of VQ and LPC VOCODER techniques has been demonstrated to produce a data rate of 800 bps with good intelligibility but little naturalness.
One of the latest systems under development is aimed at producing 450-600 bps while maintaining both intelligibility and naturalness. A recently released speech coding IC achieves up to 16:1 compression (resulting in 4096 bps given a 64 kbps input signal) using a proprietary algorithm called QCELP. The device, the Qualcomm Q4400 Variable Rate Vocoder, is a 100-pin PLCC VLSI circuit. Currently, Code-Excited Linear Prediction (CELP) is the leading technique used by industry. The leading CELP algorithms are complex and required specialized DSP processors to run them or Application Specific Integrated Circuits as with the Qualcomm device. LPC VOCODER based techniques offer promise for future very low-bit rate systems.

In addition to low-bit rate coding of speech signals, several other speech/audio coding topics are generating increasing interest as the multimedia revolution continues. Research into coding higher bandwidth audio signals is a major area. Some research into speech coding algorithms that are transparent to both voice and modem signals is being done. The coding of higher bandwidth audio signals can be divided into high quality voice transmission systems and music transmission systems. A CCITT standard for high-quality voice coding specifies a method of coding 7 kHz into 64 kbps. A technique called sub-band ADPCM is used for this purpose. A low-band (from 50-4000 Hz) and a high-band (from 4000-7000 Hz) are separately coded using ADPCM. Based on the fact that the majority of important speech signals are found in the low-band, 6-bits per samples are assigned to it and 2-bits per samples are assigned to the high-band. This results in a low-band data rate of 48 kbps (8k samples/second x 6 bits/sample) and a high-band data rate of 16 kbps (8k samples/second x 2 bits/sample) for a total of 64 kbps.

The compression of 20 kHz audio (Compact Disc quality) for transmission of music (or creating higher capacity CDs) is another area where attention has been focused recently. Three groups currently offer 4:1 compression algorithms that maintain CD quality sound: Dolby Laboratories, Inc.; Scientific Atlanta, Inc.; and the Motion Picture Experts Group (MPEG) of the International Standards Organization (ISO). These algorithms all run on DSP processors from Crystal Semiconductor, Texas Instruments, and Motorola. MPEG also has developed the Musicam (Masking-pattern Universal Sub-band Integrated Coding and Multiplexing) compression standard. This is a non-constant bit-rate compression scheme that typically results in a bit-rate on the order of 110 kbps (or 220 kbps for a stereo signal). All of these compression techniques rely on the psychoacoustical model of hearing and on the principal of masking. The important facts from this model are that lower frequencies are heard with greater precision than higher frequencies, and in the presence of a high amplitude tone, the human ear is not sensitive to lower amplitude signals of similar frequencies. In order to take advantage of these facts, the audio signal is divided into sub-bands (typically 32). Within each sub-band, the number of bits required to quantize (represent) that portion of the signal that is audible to the human ear is
determined. Further compression is achieved by reducing the number of bits allocated to higher frequencies that cannot be heard with great precision.

Although speech coding algorithms have improved to the point that 16 kbps or 32 kbps could be used to code standard telephone conversations transparently to the users, the telephone carriers continue to use standard 64 kbps PCM for telephone traffic. This is because of all of the non-voice traffic, such as facsimile and modems, that now use the telephone network. These devices make use of the 3.1 kHz audio bandwidth provided by the carriers. Thus, until ISDN eliminates the need for analog transmission of digital data, the carriers are forced to continue to use 64 kbps to transfer analog encodings of 1200, 2400, 9600, etc. bps data streams. Some research has gone into providing audio coding algorithms that would be transparent not only to voice traffic but to certain types of modem traffic as well. In 1988, CCITT approved an algorithm that codes audio at 32 kbps and is transparent to standard 4800 bps data modems. Researchers believe that theoretically up to 9600 bps data modems could be encoded safely in 18 kbps. An ideal coder would be one that could synchronize with any modem signal and identify it so that it could be coded at a value near its own data rate. The continuing development of modern technology (the v.fast standard that supports up to 28.8 kbps over a 3.1 kHz audio channel is now nearing final approval) combined with the impending coming of ISDN make the goal of creating an ideal coder one that will likely never be realized.

One real life example where low bit-rate speech coding is being used is in British Airway's SkyPhone. This service, which allows airline passengers to connect to the public telephone network while in flight, was first introduced as a trial service in 1988. For this application, the selection of a speech coding algorithm had to take into account the fact that the bit-rate available over an aircraft to Inmarsat satellite to ground station link is limited. Other problems unique to this system are high background noise (from the aircraft engines) and the likelihood of errors in transmission: a bit-error rate of 1 per 1000 was expected along with burst errors resulting from an inability to keep the aircraft antenna perfectly aligned. After a great deal of testing, a 9.6 kbps speech codec produced by British Telecom Research Laboratories (BTRL) was selected. The algorithms that produced data-rates below 9.6 kbps were all deemed to be of too low quality. The BTRL codec uses a Multipulse algorithm which turned out to be the most resistant to background noise and transmission errors. [The codec uses a Multipulse algorithm with a 10th order LPC on a 20 ms short-term prediction window, an 8 kHz sampling rate, and a WE-DSP32 microprocessor.] It also successfully passed 300 bps FSK (Frequency Shift Keying) V.21 modem signals. The delay due to the codec of 40 ms is negligible when compared to the satellite delay. Due to the fact that the transmission delay associated with a satellite makes the use of an Automatic Repeat Request (ARQ) protocol impractical, a Forward Error Correction (FEC) protocol was chosen. (A straight forward ARQ protocol can create undesirable delays
when used over a satellite link; however, a properly designed sliding-window ARQ protocol could be used with minimal delays resulting.) The combination of the FEC protocol and the BTRL Multipulse codec has produced a good quality SkyPhone system for British Airways.

The field of speech coding will continue to grow in importance as more of the world enters the digital age and as demand increases for the coding of higher bandwidth audio signals. Along with video compression, speech coding will have a profound effect on what future ISDN applications are capable of. As speech coding techniques continue to advance and VLSI-based DSP processors grow more powerful, the trade-off between data rate, quality, and complexity may become easier to negotiate, but especially as higher bandwidth options enter the picture, it will remain at the heart of the design of speech coding systems.
Chapter 3

Theory

Chapter 3 contains an in-depth description of the techniques used for coding speech in this project. This includes the three major speech coding techniques used: Adaptive Delta Modulation, Adaptive Vector Quantization, and Linear Predictive Coding. These three techniques reflect different choices in the traditional speech compression trade-off between complexity, compression, and quality. Adaptive Delta Modulation (ADM) is a simple, fast technique that yields good compression. The quality of its output is relatively poor: audible distortion should be expected. Adaptive Vector Quantization (AVQ) is more complex than ADM. It produces a higher quality result, but does not match the compression achieved by ADM. Linear Predictive Coding (LPC) is the most complex of the techniques. It has the potential to achieve compression rates better than that of ADM while maintaining good signal quality. The LPC technique used in this project is not as complex as the low bit-rate coders discussed in Chapter 2 which require specialized DSP hardware to run the algorithms; however, it achieves compression similar to that of AVQ and produces good quality output.

3.1 Adaptive Delta Modulation

The simplest form of speech compression used in this project is Adaptive Delta Modulation (ADM). This is a primitive method of coding speech used for its lack of computational intensity. It uses a differential coding technique meaning that the values saved represent the change in the signal between two samples instead of the samples themselves. Adaptive Delta Modulation achieves a constant 8:1 compression ratio, but creates audible distortion of the signal. The general technique and the implementation used in this project are described in this section.

Delta Modulation refers to using a simple 1-bit differential value to code the signal. This means that each sample is compared to the previous value and only the sign of the difference is preserved. Reproducing an approximation of the signal is done by adding or subtracting a constant delta to the value of the signal based on the current 1-bit code. Adaptive Delta Modulation improves on the technique by allowing the delta value to change to track rapid changes in the signal better. The delta begins at a low value. It is incremented (up to a certain
maximum) if multiple differences in the same direction occur. It is reduced once the string of differences ends.

The ADM algorithm used in this system allows three delta values (1,2,4). The delta is normally one. Two consecutive, equal differences will cause the delta to be increased to two. A third matching difference will cause the delta to be raised to four. Additional matching differences will keep the delta at four. Consecutive, non-matching differences at any point will return the delta to one. This method allows the algorithm to better reproduce areas where the signal changes rapidly, thus reducing the noise added by the coding.

Another consideration in the system is how differences of zero are handled. For equal values, the delta is selected such that it will be opposite the last delta selected in such a situation. This is important because, if care is not taken, an unwanted signal can be created from a string of zeros, or other constant values. In ADM, there is no zero change indication; therefore, a string of zeros will not be exactly reproduced. Using the technique described above insures that a minimal signal will be produced. This minimal signal should not produce audible distortion.

The compression achieved by the ADM algorithm used in this project is constant. Results of testing the ADM algorithm with different speech samples for subjective quality can be found in Chapter 5.

3.2 Adaptive Vector Quantization

Vector Quantization is a general compression technique. It is used for both lossless compression algorithms such as computer data compression and for lossy compression algorithms such as image compression. In this section, the basics of Vector Quantization are presented followed by a description of the algorithm used for the project.

The basic principle of vector quantization is to represent a group of samples using an index into a table of likely vectors. The compression achieved depends on the size of the table, the length of the vectors, and the number of matches that are made with the input data. The table, which is referred to as a codebook, is generated using samples of common data. For lossless compression algorithms, an exact match with the codebook is required for an index value to be substituted for the vector. For lossy compression algorithms, a fuzzy matching technique--dependent on the type of data being processed--is substituted for the exact matching requirement. Vector Quantization is often used in speech coding in combination with Linear Predictive Coding (see Section 2.2).

For this project, an Adaptive Vector Quantization (AVQ) algorithm [Ref 5] is used to provide moderate compression without the computational complexity of a Linear Predictive
Coding system. The term adaptive is used because there is no static codebook. Instead a unique codebook is generated on the fly for each speech signal. A lossy technique is used. This algorithm uses a fuzzy matching of vectors: two vectors are considered matching if they differ by less than a selected tolerance value. Increasing the tolerance value decreases the quality of the reproduced signal but enables greater compression of the signal. The AVQ algorithm for this project uses eight byte vectors with a one byte index into a 255 element codebook. Two index values and one codebook entry are reserved. One of the reserved index values points to an all zero vector that is located in a reserved entry at the start of the codebook. The other reserved index value is used to indicate a vector that could not be matched and is included in the data in the eight bytes that follow the special index. This vector is automatically added to the codebook. If a vector is successfully matched, only the index to its codebook location is output.

The AVQ algorithm used differs from a traditional VQ approach in several ways. The adaptive nature of the solution allows it to work for any signals, instead of the VQ approach that assumes its codebook has been generated such that it will be good for any signal encountered. The trade-off for this is that the codebook must be included in the compressed signal. The codebook also must be generated on-the-fly increasing processing time. Therefore, the AVQ provides greater flexibility at the cost of slightly reduced performance and compression.

The compression achieved by the AVQ algorithm used in this project varies depending on the input signal. Results of testing the AVQ algorithm with different speech samples for compression and subjective quality can be found in Chapter 5.

3.3 Linear Predictive Coding

The third method used for speech compression in this project is a Linear Predictive Coding (LPC) based technique. LPC is a general technique used in many different speech compression (and recognition) systems. This section describes LPC techniques and then how they are used to compress speech in this project.

LPC divides a speech signal into two basic components: the sound generating function and the vocal tract transfer function that shapes the sound. The sound generating function can either be voiced (created by the vocal cords) or unvoiced (created by forced air). The vocal tract transfer function can be modeled well by a 10th order, all-pole linear predictor. This choice has been found to achieve a good balance between model accuracy and computational complexity. [Ref 20] The predictor function for a value of the signal \( S_n \) based on its previous values \( S_{n-1}, S_{n-2}, \ldots, S_{n-p} \) is shown below.
\[ \hat{S}_n = \sum_{k=1}^{p} A_k S_{n-k} \]

**Figure 3.1** Predictor Function for \( p \) Pole System

When the predicted values \( \hat{S}_n \) for the signal, based on the vocal tract model, are subtracted from the original values \( S_n \), the resulting residual (or error) signal represents the sound generating function. The signal can now be represented as a residual signal and a set of coefficients for the vocal tract model. During speech, the vocal tract changes as different sounds are produced; therefore, the vocal tract model must be computed every 10-20 ms to accurately model typical speech.

The calculation of the prediction coefficients \( A_k \) used above is based on certain principles of statistics. Speech is considered to be a stochastic (random) process. A random function derived from a stochastic process is written as a probability function: \( p(x_1, x_2, x_3, \ldots, t_1, t_2, t_3, \ldots) \) that equals the probability that the value of the function will be \( x_1 \) at time \( t_1 \), and \( x_2 \) at time \( t_2 \), and so on. The mean (or average value) of a random variable \( x \) is written as \( \bar{x} \) and is equal to \( \sum_x x p(x) \). The mean of a random function \( x(t) \) is the expected value of \( x \) at time \( t \). It is written as \( \bar{x}(t) \) and is equal to \( \sum_x x \cdot p(x, t) \). The correlation of two random variables \( x \) and \( y \) is a measure of their dependence. It is written as \( \langle xy \rangle \) and is equal to \( \sum_{x,y} xy \cdot p(x, y) \). The autocorrelation is the correlation between the same function at two different times. It is written as \( r(t_1, t_2) \) and is equal to \( \langle x(t_1) x(t_2) \rangle = \sum_{x_1, x_2} x_1 x_2 p(x_1, x_2, t_1, t_2) \).

A random function is considered stationary if its statistics do not change over time. For a stationary function, \( \bar{x} = \bar{x}(t) \) and \( p(x_1, x_2, t_1, t_2) = p(x_1, x_2, \Delta) \) where \( \Delta = t_2 - t_1 \). The autocorrelation for a stationary signal becomes \( \sum_{x_1, x_2} x_1 x_2 p(x_1, x_2, \Delta) \). A special case of this type of function occurs when the average of one function over time is the same as the value computed by averaging many functions derived from sampling the same stochastic process many times. This special case is called ergodic and results in the following:

\[ \bar{x} = \lim_{N \to \infty} \left( \frac{1}{N} \sum_{t=-N}^{N} x(t) \right) \]

and

\[ r(\Delta) = \lim_{N \to \infty} \left( \frac{1}{N} \sum_{t=-N}^{N} x(t) x(t-\Delta) \right) \]

It is this autocorrelation function, a sum of lagged products, that is important in calculating the prediction coefficients \( A_k \) as we will see below.

The objective in choosing the prediction coefficients \( A_k \) is to minimize the mean squared prediction error:

\[ \bar{e}_n^2 = \left( \bar{S}_n - \hat{S}_n \right)^2 = \left( S_n - \sum_{k=1}^{p} A_k S_{n-k} \right)^2 \]

In theory, this could be done...
by setting the partial derivative of $\varepsilon_n^2$ with respect to each $A_k$ equal to zero, and then solving the resulting set of simultaneous linear equations for the unknown $A_k$'s:

$$\sum_{k=1}^{p} S_{n-1}S_{n-k}A_k = S_{n-1}S_n$$
$$\sum_{k=1}^{p} S_{n-2}S_{n-k}A_k = S_{n-2}S_n$$
$$\vdots$$
$$\sum_{k=1}^{p} S_{n-p}S_{n-k}A_k = S_{n-p}S_n$$

Figure 3.2 Simultaneous Linear Equations for Optimal Predictor

In fact, the optimal linear predictor determined as above cannot be used. This is because, for speech synthesis, it is essential that the vocal tract model transfer function by stable. This requires that the denominator of the vocal tract model transfer function have all of its roots inside of the unit circle. There are a number of methods that have been developed to find a set of coefficients that meet the above requirement. Two of the most common methods are referred to as the autocorrelation method and the covariance method. Both of these methods were used during the development of the system; however, the autocorrelation method was selected for use in the final system so it is the one discussed here.

The autocorrelation method uses the sum of lagged products autocorrelation function shown above. Although this is valid only for signals that are stationary over all time, it can be used for segments of a speech signal during which the signal is stationary. This is done by zeroing are values outside of a finite time interval (10-20 ms) that is short enough so that the signal will be stationary for that period. A sharp truncation of a speech segment at the ends of each finite time interval could cause prediction errors at each end of a segment; therefore, a tapered windowing function, whose amplitude falls gradually to zero, is typically used. In this project, a Hamming Window, shown below, is used. $N$ is the size of the window in samples.

$$x' = 1 + 0.84 \cos\left(2\pi x / N \right)$$

Figure 3.3 Hamming Window

Using the autocorrelation method the set of simultaneous linear equations that must be solved becomes:
\[
\sum_{k=1}^{p} S_{n-[1-k]} S_n A_k = S_{n-1} S_n \\
\sum_{k=1}^{p} S_{n-[2-k]} S_n A_k = S_{n-2} S_n \\
\vdots \\
\sum_{k=1}^{p} S_{n-[p-k]} S_n A_k = S_{n-p} S_n
\]

Figure 3.4 Simultaneous Linear Equations for Autocorrelation Method

These equations can be solved efficiently by a technique known as Levinson Recursion. First, the autocorrelation coefficients \( R_k \) are computed using the following (sum of lagged products) equation: \( R_k = \sum_{i=0}^{N} S_i S_{i+j} \). [Note that no other values of \( S_n \) contribute since it was zeroed outside of the finite time window.] Next, the values of the two sets of coefficients (the prediction coefficients \( A_k \) and the reflection coefficients \( K_k \) - whose usefulness if discussed later) are computed using the following recursive relationship.
Stage = 0
E = R_0
A_0 = 1
Stage > 0

\[ K_{\text{Stage}} = \frac{R_{\text{Stage}} - \sum_{j=1}^{\text{Stage}-1} (A'_j)(R_{\text{Stage}-j})}{E'} \]

\[ A_{\text{Stage}} = K_{\text{Stage}} \]

\[ A_{x \in \text{Stage}-1} = A'_x - (K_{\text{Stage}})(A'_{\text{Stage}-x}) \]

\[ E = E'(1 - (K_{\text{Stage}})(K_{\text{Stage}})) \]

Variables

\[ A_k = \text{Prediction Coefficients for Current Stage} \]
\[ A'_k = \text{Prediction Coefficients for Previous Stage} \]
\[ K_k = \text{Reflection Coefficients} \]
\[ R_k = \text{Autocorrelation Coefficients} \]
\[ E = \text{Prediction Error} \]
\[ E' = \text{Prediction Error for Previous Stage} \]

Figure 3.5 Levinson Recursion

Once the prediction coefficients have been computed by the autocorrelation method, they are used to compute the predictor values \( \hat{S}_n \), which are then subtracted from the original signal \( S_n \) to produce the residual signal. This process has not achieved any compression to this point. In fact, the amount of information has been increased by 20-40% because now the prediction coefficients and the residual signal are both required to reproduce the signal. The advantage that has been gained is that the residual signal can be compressed to a much greater degree than the original. The coefficients can also be quantized into much less storage space without distorting the reproduced signal. As mentioned in Chapter 2, LPC-based systems range from LPC Vocodersthat remove the residual entirely and substitute additional information computed about the signal—to Code Excited Linear Predictors (CELP)—that attempt to store the important parts of the residual in as little space as possible. In this project the methods of compressing the residual and the coefficients are less complex than those of modem low-bit rate speech coding systems, but operate on some of the same principles.

The two main principles used to compress the residual are as follows. One, the important part of the residual are the peaks; these need to be reproduced as accurately as possible to preserve signal quality. Two, the residual can be center clipped severely without
significant information loss. To preserve the peaks, a VQ scheme (see Section 3.2) is used. A static, 32-element codebook was generated by selecting peaks from a set of residuals derived from a number of different recorded signals. A count was maintained for the vectors surrounding each peak within a certain tolerance of each other. Ultimately, the 32 most frequently occurring vectors were selected for the codebook. During the encoding process, the residual is searched for peaks above or below a minimum threshold. The vector surrounding the peak is then matched to the codebook. Matched vectors are replaced by a 5-bit index into the codebook. Elements of the signal that are not matched during the VQ process are encoded into a 5-bit value (1-bit sign/4-bit magnitude) using center clipping. A run-length coding scheme is then used to further compress the residual. The stored signal is divided into 8-bit elements coded as follows:

<table>
<thead>
<tr>
<th>b0</th>
<th>b1</th>
<th>b2</th>
<th>b3</th>
<th>b4</th>
<th>b5</th>
<th>b6</th>
<th>b7</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>Value coded in xxxxx occurs n times*</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>Value coded in xxxxx occurs once</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>Value coded in xxxxx occurs twice</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>Value coded in xxxxx occurs three times</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>Value coded in xxxxx occurs four times</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>Value coded in xxxxx occurs five times</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>Value coded in xxxxx occurs six times</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>xxxx is an index into the VQ Codebook</td>
</tr>
</tbody>
</table>

*The repeat count n is an 8-bit value stored in the next byte.

When the residual is decoded, the 8-bit values are expanded into strings of repeated values or VQ vectors. The restored residual is then passed to the LPC decoder to generate the output signal. [Other experimental methods for compressing the residual as well as the effectiveness of the above method are discussed in Chapter 5.]

The quantization of the coefficients takes advantage of the fact that the Levinson Recursion method used to generate them also produces reflection coefficients that are guaranteed to be between +1 and -1. (This is the criteria required for the stability of the transfer function.) The prediction coefficients used in the actual LPC encoding/decoding can be easily computed from the reflection coefficients. It is a simple matter to quantize the reflection coefficients to 8-bit values. A non-linear quantization approach that represents the values close
to +/-1 with the most precision yields the best results. For this project, both a linear quantization approach and a non-linear approach--using the $\sin^{-1}(\ )$ of the reflection coefficients and scaling that result to 8-bits--were used. The results of different methods of quantizing the coefficients is discussed in Chapter 5. Although it has been demonstrated that the coefficients can be quantized to less than 8-bits each, using more advanced techniques that treat each coefficient individually, that was not attempted in this project. It is important that the encoding and decoding LPC processes use the same prediction coefficients. To accomplish this, the reflection coefficients are calculated and quantized before the encoding process. The prediction coefficients are then generated for the encoding process from the quantized reflection coefficients in the same manner as will be done in the decoding process.

The compression achieved by the LPC algorithm used in this project varies depending on the input signal. Results of testing the LPC algorithm with different speech samples for compression and subjective quality can be found in Chapter 5.
Chapter 4 focuses on the design of the system developed for this thesis project. The system is named the e-Mail with Embedded Speech System (MESS). It is intended to demonstrate an application of ISDN. Two "Phone Terminals" (referred to as system nodes throughout much of the document) that enable the creation, modification, and review of hybrid voice/text mail messages were developed. Conceptually, a "Phone Terminal" is an inexpensive device that interfaces to both a local area network for local communications and to ISDN for global communications. The "Phone Terminal" provides connectivity for people either not interested in or not able to afford a multimedia personal computer. For this system, the "Phone Terminals" are connected via an ethernet. An 80486 personal computer connected to the ethernet provides the simulated function of an ISDN Central Office System. The general system setup is shown below.

![High Level System Diagram](image)

This chapter is divided into two sections. Section 4.1 is a description of the hardware used for the system. Section 4.2 is a discussion of the software concepts used in developing the
system. Neither of these sections go into detail on the low-level aspects of the system. Circuit diagrams and source code can be found in the appendices if desired.

4.1 Hardware

This section focuses on the hardware used to implement the system developed for this project. The hardware consists of both commercial-off-the-shelf items and items designed specifically for this project. The hardware is divided into two main categories: that used for the Central Office System and that used for the Nodes. This section contains only a general description of the hardware used. The low-level code developed to control the hardware is discussed in the Software section. Circuit diagrams and source code can be found in the appendices.

4.1.1 Central Office System Hardware

The Central Office System consists of a 80486 personal computer (PC) and two commercial ISA-bus card. The first card, the Allied Telesis 1500, is a commercial ISA-bus Ethernet card. It is used to interface to the nodes. This card is controlled by the Central Office system software. The second card used as a part of the central office is the TyIN 2000. This card is used for input/output of speech samples on the 80486 system and did not require controlling software to be written for it. The process of converting the audio format used by the nodes to the format used by the TyIN 2000 is described in Appendix F.

The Allied Telesis 1500 card uses the Am79c960 Single-Chip Ethernet Controller for ISA. This is largely software compatible with the Am7990 Local Area Network Controller for Ethernet that is used as the controller for the 68000 board Ethernet interface. Although much of the low-level interface code for the controller was ported from the code for the 68000 controller, additional low-level code was required to allow the board to function in the more complicated PC environment. Code to setup the 8237 DMA Controllers and 8259 Programmable Interrupt Controllers (PIC) has been developed (one problem encountered with the PIC is discussed below). In order to preserve as closely as possible the commonality between the PC and 68000 basic interface code the I/O to the controller chips has been coded as macros that hide the difference between the PC's separate I/O space and the 68000's memory mapped I/O. Appendix G contains I/O addressing information for components that are that are important to the system.

The combination of the AT1500 board and the PIC(s) on the PC appear to be working incorrectly under certain conditions. Allied Telesis has acknowledged that a problem has been observed under certain conditions. In order to avoid this problem, a timer interrupt is used to
trigger a polling routine that controls the Ethernet board in place of the IRQ driven method. No problems were encountered using this method. (See Section 5.3 - Difficulties.)

The second ISA-bus card used as a part of the central office, the TyIN 2000, is a Microsoft Windows compatible sound card. It is not required for basic functioning of the central office system, but is used to play properly converted speech samples on the 80486 system. This allows testing of Speech Coding algorithms independent of the rest of the system and provides added diagnostics for the overall system.

4.1.2 Node/Phone-Terminal Hardware

A Node/Phone-Terminal consists of an SBC68K Single Board Computer, an interface board, audio circuitry/components, and a Thick-to-Thin Ethernet Transceiver. The SBC68K Single Board computer is built by Arnewsh, Inc. and consists of a 12 MHz mc68000, 128 kbytes of RAM, 32 kbytes of EPROM for firmware, and additional components to provide necessary functions such as serial interfaces to a host and terminal. The interface board, designed specifically for this project, connects to the SBC68K bus. It contains the necessary circuitry to support an ISDN Digital Subscriber Controller (DSC) and a Local Area Network Controller for Ethernet (LANCE). The audio part consists of a speaker/microphone and amplification/isolation circuits designed for the project. The Ethernet Thick-to-Thin Transceiver is a commercially produced device. A block diagram for a node is shown below.

![Block Diagram of a Node/Phone-Terminal](image)

Figure 4.2 Node Block Diagram
The DSC circuitry is used to provide audio functions. The Am79C30A is the main part of the DSC circuitry. It is memory-mapped into the SBC68K address space using 7FFFF0-7FFFFF. The directly addressable registers are shown in Appendix H. All of the registers are 8 bits wide. In addition to the directly addressable registers there are some 56 indirectly addressable registers. These registers are accessed by writing an address to the Command Register. The Data Register is then used to read/write the selected indirect register. The indirect registers control the function of the various subsections of the DSC. These are the Line Unit Interface (the ISDN S-bus connection), the B-Channel Multiplexer, the Main Audio Processor, the Data Link Controller, and the Peripheral Port. The Line Unit Interface and Data Link Controller would be used if the system were to be connected to an ISDN line. The Peripheral Port is designed as an interface to an ISDN terminal. In this project the Main Audio Processor (MAP) and the B-Channel Multiplexer (MUX) are used. The MAP is used to perform analog-to-digital and digital-to-analog conversion using standard u-Law coding (discussed in Appendix F). It also provides other functions such as tone generation and volume control. The MUX is used to route the MAP's B-Channel to the B-Channel #1 (see chart above) interface to the microprocessor. The DSC provides a data ready interrupt every 0.125 ms so that the system can read/write the B-Channel at the correct 64 kbps data rate.

The LANCE circuitry is used to interconnect the Nodes and the Central Office System over a Thin-Wire Ethernet (also called Cheapernet). The main components are the Am7990 LANCE and the Am7992 Serial Interface Adapter (SIA). The Am7990 is memory-mapped into the SBC68K address space using 7FFE0-7FFEF. The directly addressable registers are listed in Appendix H. The registers are 16 bits wide. The Address Register is used to selected between four Control/Status Registers. The selected Control/Status Register is then accessed through the Data Register. All additional interaction with the Am7990 is done via direct memory access (DMA). The Am7990 is interfaced to the 68000 bus in a Bus-Mastering configuration. It is capable of taking control of the bus to read an initialization block, to poll its receive and transmit descriptor rings, and to read/write packets of data in memory. The SIA is used by the LANCE to drive an external Ethernet Thick-to-Thin Transceiver that is connected to the interface board through a set of pulse transformers for isolation.

4.2 Software

This section focuses on the software used to implement the system developed for this project. The topics in this section fall into three different categories: issues relevant to the Central Office system, issues relevant to the nodes, and issues that affect both the Central
Office and the nodes. This section contains only a Central Office conceptual discussion of the software concepts used. The actual source code can be found in the appendices.

Section 4.2.1 focuses on the system modes seen by a user and the commands available to the user in each mode. It provides a high-level description of the implementation of commands. Section 4.2.2 describes the Node/Central Office protocol that is key to enabling node requests to be processed by the Central Office. Section 4.2.3 details the format of the key element of the system—the mail message—as it travels throughout the system. Section 4.2.4 explains the local message queue used for temporary mail message storage on the nodes. It also goes into the memory limitations of the SBC68K Single Board Computer and the simple memory allocation scheme used to address these limitations. Sections 4.2.5/4.2.6 cover the storage of messages and speech data by the Central Office system. Section 4.2.7 describes the interrupt handling mechanism used on the SBC68K system to support a number of devices. Section 4.2.8 details the implementation of the terminal interface for the system. It describes both the low-level interface details and the higher-level routines for standardizing input/output. Section 4.2.9 focuses on the interface to the ISDN Digital Subscriber Controller (DSC) that provides audio capabilities for the nodes. Section 4.2.10 details the programming interface to the Local Area Network Controller for Ethernet (LANCE) that is used by the nodes. Section 4.2.11 continues the discussion of programming an ethernet controller but focuses on the Central Office ethernet interface.

4.2.1 MESS System Modes

The MESS system has three major modes that a user can be in: the main menu, text editing, and mail message processing. This section describes how each of these modes is implemented and how they are tied together. The focus of this section is on high level functionality; details on the node/central office protocol, the audio system, and the local message queue are given in later parts.

The main menu provides a command based interface for performing simple tasks, selecting another mode, and displaying system information. The creation/modification of a text message and the addition/playback/removal of voice messages are separated into two different modes. The text editing mode performs just the creation/modification of the text portion of a message. The mail message processing mode provides both the voice addition/playback/removal interface and the message distribution options: send, forward, save, and discard.
4.2.1.1 The Main Menu

Once the user has "logged in," he/she is placed in the main menu. A list of valid commands is displayed in the text window. There are three general commands: Help, Version, and Quit. The additional five commands deal with mail messages. The Edit command is used to enter text editing mode to create a new mail message. The Read command is used to load the top message from the local message queue and enter mail message processing mode. The remaining commands Display Pending Messages, Get Message held by Central Office, and Look at Message held by Central Office all return the user to the main menu after performing a simple function. The Display Pending Messages command prints the number of messages in the local message queue and the number of message held by the Central Office for the user. (Appendix I contains a Quick Reference Guide for the system.) Both of the messages pending counts are maintained by the node. The count of messages held by the Central Office is initially obtained from the Central Office when a user first logs in. The Central Office also updates the message count for a user anytime a new message--for that user--is received. The node decrements the count each time it instructs the Central Office to delete a message. The final two commands both retrieve the top message pending from the Central Office system and add it to the end of the local message queue. They differ in that Get Messages held by Central Office instructs the Central Office to delete the message once it has been retrieved and Look at Message held by Central Office instructs the Central Office to move the message to the end of its queue once it has been retrieved.

4.2.1.2 Text Editing Mode

The text editing mode (or the editor) is used for the creation of or modification of the text portion of a message. This mode may be entered from the main menu--creation of a new message--or from the mail message processing mode--modification of a saved or received message. Voice messages are not valid in text editing mode since the text to which they are attached may be moved, changed, or deleted. For this reason, editing a saved or received message causes old voice messages to be automatically removed.

In text editing mode, the user creates a text message using standard editing and cursor movement keys. The editor provides only overtype mode. It supports a simple form of word wrapping; while text is being entered at the end of the document word wrapping is performed. Other than at the end of the document, however, word wrapping is not performed. The editor stores the message in the virtual edit window it shares with the mail message processing system. Exiting text editing mode is accomplished by either aborting, which returns the user to the main
menu, or by continuing (saving) which preserves the contents of the virtual edit window and passes control to the mail message processing system.

4.2.1.3 Mail Message Processing System

The mail message processing system serves a number of functions. It is used to attach voice messages to newly created text messages. It is used to read/modify saved or received messages. This includes playback of voice messages, removal of voice messages, and the addition of new voice messages. It is also possible to pass any message to the editor for modification of the message text. The mail message processing system provides the interface that allows a user to send/forward a message, to discard a message, or the save the message to the local message queue for later retrieval.

When a message is read from the local message queue it is removed from the front of the queue. Its contents are transferred to the virtual edit window. The discard command thus simply returns the user to the main menu. The quit command actually must rewrite the message to the end of the local message queue. The send/forward command sends a copy of the message, in the virtual edit window, (with the userid of the addressee) to the Central Office System. The edit command removes all old voice messages, but otherwise preserves the contents of the virtual edit window, and passes control to text editing mode. After a message has been edited it is returned to the mail message processing system for the addition of new voice messages and/or distribution.

The voice message manipulation commands require interaction with the audio portion of the system, the terminal control portion of the system, and the Central Office System. The audio portion of the system provides the ability to record and playback speech samples. The terminal control portion of the system allows the marking of text within a message. The Central Office System provides storage for speech samples.

In order to attach a voice message, a valid handle must be obtained; a speech sample must be recorded, transmitted to the Central Office and stored under the handle; and the appropriate text must be highlighted and marked. The first step in the process is to verify that a user has selected a valid location for a voice message. Three checks are made: a voice message must be attached to text, not white space; only one voice message per line is allowed; and a maximum of four voice messages per document are allowed. If all of these conditions are satisfied, a handle is requested from Central Office. Next, the speech sample is recorded. The speech data is then divided up into multiple packets that are assigned sequential ids, labeled with the handle obtained earlier, and sent to the Central Office. A flag is set in the Id of the final packet to indicate the end of the data to a node retrieving it. Once all of the speech data has
been successfully stored, the text to which the speech is to be attached must be marked. A marked "word" is identified by searching in each direction (from the current cursor position) until white space is found or until the start/end of the line is reached. The selected "word" is limited to a maximum of 16 characters. The characters in this "word" are then marked by setting to one the most significant bit in their eight-bit code. The terminal interface routines automatically will display marked characters in bold/highlighted text. The final step is to record the handle value and position into the appropriate arrays.*

Playback of a voice message requires retrieval of the speech data, associated with its handle, from the Central Office and playing of the speech sample through the audio system. The first step in the process is to check that the text is marked. If it is, the handle associated with the text is identified by matching the current row with a value in the HandlePosition[] array. The handle is then extracted from the corresponding row of the HandleValue[] array.* The handle is used to retrieve the speech data the Central Office. At this point, the speech data is played through the audio system.

Removal of a voice message requires that the marked text be returned to normal and that the HandlePosition[] and HandleValue[] arrays be updated. The marked text is cleared by setting to zero the most significant bit of the eight-bit code for each character. The HandlePosition[] line corresponding to the row of the marked text is identified. This line of the HandlePosition[] and HandleValue[] arrays are cleared by moving the subsequent lines forward and setting the last line in use to null values.*

This overview of how each of the three major modes of the MESS system operates is not meant to cover details of lower-level routines. If desired, these details can be found in individual parts on the specific area addressed.

* For details on the storage of handle positions/values on the nodes see Section 4.2.3.

4.2.2 Node/Central Office Protocol

The job of the Central Office System is to provide services and perform actions based on requests from any of the nodes. The primary service provided by the Central Office is storage of messages and speech data. Communications between the nodes and the Central Office system take place over the ethernet. The physical ethernet hardware itself simply provides a means of transmitting packets of data between connected systems. An addressing scheme to distinguish the recipient of a packet and a collision detection/recovery capability to take care of interference problems are also provided. The protocol described below is required
to augment these capabilities to insure that even when packets are lost or corrupted during transmission, the node and the Central Office can maintain sensible communications.

The first job of the protocol is to provide a set of commands for nodes to issue and a set of responses for the Central Office to use in answering. The commands and responses are listed below.

Table 4.1 Protocol Commands/Responses

<table>
<thead>
<tr>
<th>Command</th>
<th>Expected Response</th>
<th>Alternate Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>RequestHandle</td>
<td>AssignHandle</td>
<td>NotFound</td>
</tr>
<tr>
<td>RequestMsgCount</td>
<td>MsgCount</td>
<td>n/a</td>
</tr>
<tr>
<td>StoreMsg</td>
<td>ACK</td>
<td>n/a</td>
</tr>
<tr>
<td>RetrieveMsg</td>
<td>IncomingMsg</td>
<td>NotFound</td>
</tr>
<tr>
<td>MoveMsg</td>
<td>ACK</td>
<td>NotFound</td>
</tr>
<tr>
<td>DeleteMsg</td>
<td>ACK</td>
<td>NotFound</td>
</tr>
<tr>
<td>StoreData</td>
<td>ACK</td>
<td>n/a</td>
</tr>
<tr>
<td>AppendData</td>
<td>ACK</td>
<td>n/a</td>
</tr>
<tr>
<td>RetrieveData</td>
<td>IncomingData</td>
<td>NotFound</td>
</tr>
</tbody>
</table>

There is one additional response, NACK, whose use will be discussed later in describing the error detection and recovery aspects of the protocol. ACK is short for Acknowledge and NACK is short for Negative-Acknowledge.

Assuming—for the moment—that nothing goes wrong, the protocol functions as follows. A node sends a command packet (which may contain message/speech data to be stored) to the Central Office. The Central Office receives and attempts to process the command packet. If the command packet instructs the Central Office to simply perform an action (such as storing the data included in the packet), the Central Office does so and replies with an ACK indicating successful completion. If the command packet requires information to be sent to the node (such as a count of message pending or a message itself), a response packet containing the requested information is sent in place of the ACK. In the case where a node requests an impossible action (such as retrieving a message when the message queue is empty), a special NotFound response packet is sent to the node. In all cases, the Central Office sends a response packet of some sort to a node from which it receives a command packet.

An important job of the protocol is to make sure that the same results as above occur even when something does go wrong. There are two major problems that must be considered.
First, a packet could be corrupted during transmission. Second, a packet could be lost and never reach its destination. The functions performed by the nodes and the Central Office dictate that they approach solving these problems in different ways. A node deals only with the user and the Central Office. It can, therefore, afford to wait while attempting to communicate with the Central Office. A node initiates a request and knows what responses are possible. The Central Office, on the other hand, must be ready to process any command from any node at all times. It must process each command packet as quickly as possible and be ready for another incoming packet. The nodes have, therefore, been assigned the task of monitoring requests to see that they are completed satisfactorily.

Consider first the case of a packet being lost or corrupted on its trip from a node to the Central Office. In the case of a lost packet, the Central Office will not respond since it has no idea that a request was made. The node simply times out after waiting a certain amount of time for a Central Office response, and retransmits the request. In the case of a corrupted packet, the Central Office will receive the packet, but detect an error using a CRC check. The Central Office could simply discard the packet forcing the node to time out and retransmit the request; however, instead a NACK response is used to speed up the process. Upon reception of a NACK packet the node immediately retransmits its request. The node maintains a count of time outs and NACKs and will give up--reporting that the Central Office is temporarily down--after repeated failures.

The case of a packet lost or corrupted as it travels from the Central Office to a node is a more serious one. In this case, the Central Office may have already performed the requested action and is trying to inform the node. Take, for example, the case of a StoreMsg command packet. The Central Office stores the message and sends an ACK response packet. If this ACK packet is lost, the node will eventually time out, having received no response, and retransmit the request. The Central Office will then store a duplicate of the message! To prevent problems such as this, the protocol has been structured so that the Central Office is able to determine if it is looking at a first time or repeated request. A one-bit sequence number added to each command packet is used for this purpose. Each node toggles this sequence number each time it sends out a new request, but not when repeating a request. The Central Office maintains a record in the AddressNode associated with each node (see Section 4.2.5 for message storage details) of the sequence number value of the last command packet processed. Thus, if it receives a request it has already processed--the sequence number value has not changed--it knows that it need only re-send the response packet instead of repeating the entire command.

The only exception to the orderly process described above occurs when the notification of a new message is sent to a node. A node initiates the process by sending a message, addressed to another node, to the Central Office for storage. The Central Office will then
broadcast a MsgCount packet to the addressee node to let it know, if it is up and running, that it has received a new message. (Nodes also request automatically an updated MsgCount when first coming up so that missing this notification is not critical.) This could potentially cause a problem with the protocol if this unsolicited packet is misinterpreted as the response to a lost command packet. The chance of this happening in such a way as to cause a problem, especially on a system with a limited number of nodes and limited activity, is extremely small and is considered negligible. If it is determined that this approach is undesirable, the problem can be eliminated without changing the protocol by requiring nodes to periodically request an updated message count instead of allowing the Central Office to take the initiative to notify them.

The Node/Central Office System protocol is robust and provides all of the capabilities needed for nodes to make use of the Central Office system. To port the protocol from an ethernet based system to an ISDN BRA based system would require some adaptation. Requests of the Central Office system would be made using D channel signaling and would have to fit in with requests for the many other services provided by an ISDN Central Office. Once a request had been validated using the D channel, the data would be transferred using a B channel. The basic commands/responses would be the same, but their implementation would be different because of the separation of signaling information and data.

4.2.3 Mail Message Format

The basis of the system is the mail message. The two key aspects of any mail message are handles and text. A handle represents a stored speech sample that is attached to some text. This section describes the format of a mail message and how it is stored as it moves between the editor/mail system on one node to the Central Office system and on to another node.

There are several size and format limitations imposed on mail messages. The virtual edit window is 80 columns by 20 rows. (The physical display window where text is displayed is 80 columns by 12 rows.) This is as large as mail messages can be. Mail messages (unlike speech samples) are required to fit within a single ethernet packet, so the total number of characters (including up to 20 characters for handles) permitted is 1500. Limits are also placed on the number and positioning of handles. Only one handle per row of text is allowed. A maximum of four handles per mail message is also imposed. It is required that handles be attached to text, not white space.

The editor and mail message processing mode share the 80 by 20 (1600 byte) virtual edit window. (See System Memory Map - RAM in Appendix B, for more detail.) Text is stored using eight bits per character consisting of a seven-bit ASCII value and a one-bit flag. The flag is used to indicate if the character is part of a marked text block. Two ASCII values are reserved
for control characters: ASCII_ETX (03h) marks the end of a line and ASCII_EOT (04h) marks the end of a mail message. Handles are stored using separate HandlePosition[] and HandleValue[] arrays. The HandlePosition[] array contains the rows where each handle is located. The HandleValue[] array contains the 32-bit handle assigned to a particular speech sample by the Central Office system. These arrays are maintained in order by HandlePosition. The handle associated with a certain marked text block is found by matching its row in the HandlePosition[] array then extracting the handle from the HandleValue[] array.

When a message is written to the local message queue, its format is changed. A message begins with a list of 32-bit handle values (maintained in row order) terminated by a null handle (0h). The text of the message follows with variable-length rows separated by ASCII_ETX characters. After the last row in the message, an ASCII_EOT character is appended. Restoring a message from the local message queue is a simple process. The HandlePosition[] and HandleValue[] arrays are restored from the handle list by searching sequentially through the text for marked blocks. The text is then expanded back into the virtual edit window.

The format used for the local message queue is the same as that used when a mail message is transmitted to the Central Office system. The message in the local queue is copied to a transmit buffer and inserted into a complete ethernet packet for transmission (see Section 4.2.10 for LANCE interface details). The Central Office system stores mail messages unchanged and forwards them when requested.

The structure of the local message queue for nodes and the message storage approach of the Central Office system are each the subject of a separate section.

4.2.4 The Local Message Queue and SBC68K Dynamic Memory Allocation

The SBC68K Single Board Computer that is used for the system nodes has a total of 128 kbytes of RAM. The Aztec C cross-compiler used to generate code for the SBC68K does not provide dynamic memory allocation and is not aware of the 128k byte limitation. Therefore, a simple memory allocation scheme was developed for this project.

The Aztec C cross-compiler automatically generates a Code Segment and Initialized and Uninitialized Data Segments. These are followed in memory by USER and SUPERVISOR stacks and the System Heap. (Modifications made to the compiler to control the size and placement of the stacks and heap are shown in Appendix E.) The remaining memory, up to the 128k byte limit, is dynamically allocated to the virtual edit window, the LANCE receive/transmit buffers, and the local message queue. The size of the first three dynamically allocated items remain constant; therefore, the local message queue can make use of all free memory remaining after their allocation. (A System Memory Map - RAM is shown in Appendix B.)
The local message queue is implemented as a continuous wrap-around buffer. Two routines, alloc and dealloc maintain ReadMailPosition and WriteMailPosition pointers. Access to the local message queue is performed through a set of macros, RMAILPOS and WMAILPOS, that insure that address values are properly wrapped around. ReadMailPosition points to the oldest message in the queue. WriteMailPosition leads ReadMailPosition and points to the next available free memory within the queue. The amount of free memory available can be determined by the distance WriteMailPosition may be advanced before catching ReadMailPosition. Deleting a message (and deallocating its memory) is accomplished by advancing the ReadMailPosition pointer to the next message. Storing a message (and allocating memory for it) is accomplished by advancing the WriteMailPosition pointer.

In order to allow access to all messages on the local message queue, the following practice was adopted. When a message is stored, it is placed at the end of the queue. Messages that are read or edited are automatically removed from the local message queue. In the mail message processing system, therefore, the discard command actually does nothing. The quit and send command actually save the message by writing it back to the end of the queue. This allows a user to scroll through all pending messages sequentially rather than having to act on each message before being able to display the next one.

4.2.5 Central Office Message Storage

The Central Office System maintains dynamic message queues. There are separate queues for each addressee for whom messages are pending. At startup, a list of AddressNodes* is created. It is initially null. Each time the Central Office system receives a request from a node or a message addressed to a node, it checks its list of AddressNodes. If the node is not found, then a new AddressNode is created for it. (In addition to message storage, AddressNodes are also used for the Node/Central Office protocol.) Each AddressNode has a message queue (actually another linked list composed of MessageNodes) associated with it. The standard library functions malloc and free are used to allocate and deallocate storage for MessageNodes as they are created and deleted. The format of AddressNodes and MessageNodes are shown below.
The Central Office system provides five commands for nodes to manipulate the message queues. The first two do not affect existing messages. These are RequestMsgCount which returns the number of messages pending in the queue for a node, and StoreMsg which adds another message to the queue of the addressee. The final three commands all operate on the message at the top of the queue: RetrieveMsg sends a copy of the top message, but does not alter the queue; DeleteMsg removes and discards the top message from the queue; and MoveMsg places the top message at the end of the queue. The general appearance of the Address/Message queue follows:
List of AddressNodes

```
\[ \]
\[ \]
AddressNode 1 --> AddressNode 2 --> ... --> AddressNode N --> NULL
\[ \]
\[ \]

MessageNode 1.1 --> MessageNode 2.1 --> ... --> MessageNode N.1
\[ \]
\[ \]

MessageNode 1.2 --> MessageNode 2.2 --> ... --> MessageNode N.2
\[ \]
\[ \]

\[ \]
\[ \]
NULL NULL NULL
```

Figure 4.3 Message Storage Structure

* The terms AddressNode and MessageNode refer to Nodes of a linked list and should not be confused with systems nodes (written in small letters in this section).

4.2.6 Central Office System Speech Storage and Handle Maintenance

The Central Office System provides storage for speech data recorded by a node. Each stored speech sample is assigned a handle by the Central Office system. Handles (unique 32-bit values) are used by the nodes within messages and when retrieving a speech sample for playback. (See Section 4.2.5 for details on the mail message format.) At startup, a linked list of HandleNodes is created. Initially this list is empty. HandleNodes are created and added to the
list only when data to be stored under that handle is received. Each HandleNode has a usage count and a data field associated with it. The usage count tracks the number of stored messages that contain a given handle. The data field points to a linked list of DataNodes that contain the actual speech data. The format of the data stored in each of the DataNodes is specified by a data format field associated with each HandleNode. When new data is added it is stored in the format specified by the CurrentDataFormat variable that is set from the Central Office System's control screen. The data formats include Normal (uncompressed) and several compressed formats. These formats are discussed in Chapter 3. Data is automatically converted from its storage format to the format expected by the nodes when it is transmitted out. A buffer is used for processing/temporary storage of the data and storage space for the compressed data is allocated dynamically. The structure of the HandleNodes and DataNodes are shown below.

Table 4.4 HandleNode (HN) Structure

<table>
<thead>
<tr>
<th>FieldName</th>
<th>FieldType</th>
</tr>
</thead>
<tbody>
<tr>
<td>Handle</td>
<td>32-bit Value</td>
</tr>
<tr>
<td>UsageCount</td>
<td>Number</td>
</tr>
<tr>
<td>DataFormat</td>
<td>Number</td>
</tr>
<tr>
<td>Data</td>
<td>Pointer to DN list</td>
</tr>
<tr>
<td>Next</td>
<td>Pointer to HN list</td>
</tr>
</tbody>
</table>

Table 4.5 DataNode (DN) Structure

<table>
<thead>
<tr>
<th>FieldName</th>
<th>FieldType</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>Number</td>
</tr>
<tr>
<td>Data</td>
<td>Pointer to a memory block</td>
</tr>
<tr>
<td>Next</td>
<td>Pointer to DN list</td>
</tr>
</tbody>
</table>

The DataNodes for a given HandleNode each have an ID associated with them (the first byte of data). The IDs are assigned by the node requesting storage. The Central Office system maintains the list of DataNodes in ID order.

The Central Office system provides three commands for nodes to use in storing and retrieving speech data. In addition two message manipulation commands affect HandleNodes. To store a speech sample, the node uses the following command sequence. First, a new handle is assigned using RequestHandle. Second, the speech sample is broken up into packet sized
blocks, assigned sequential IDs, and sent to the Central Office using the StoreData command. The final ID is marked with a flag that indicates it is the last in the sequence. The assigned handle and IDs are used with the RetrieveData command to get and rebuild the speech sample for playback. The usage count (whose use is described below) is set to zero when a HandleNode is created. It is incremented each time a message containing its handle is received by the Central Office in a StoreMsg command. It is decremented each time a message containing its handle is deleted by the Central Office in response to a DeleteMsg command. No other commands affect the usage count.

The number of possible handle values is limited by the Central Office. (This limit is set based on system requirements and usages. Typically, it should be well below it maximum value of $2^{32}$.) For this reason, the Central Office uses a specific procedure to assign handle values and free old, no longer used handles. Initially, handle values are assigned sequentially until the maximum is reached. If all handles have been assigned, the Central Office first searches (from oldest to newest) for a handle that does not have any data stored under it. If this search fails, the Central Office then searches (again from oldest to newest) for HandleNodes with a usage count of zero. If this search succeeds, the HandleNode and associated data are discarded (reclaiming the used memory) and the handle value is reassigned. If all of the handle values are in use, the Central Office informs the node that a handle cannot be assigned. The selection of the oldest unused handle value for reassignment is important. This is because the number of possible handle values is chosen so that a significant amount of time will have passed before a handle faces the possibility of reassignment. This is critical because, although nodes do not preserve messages beyond the login time of a user, a node may have in temporary storage a copy of a message that is no longer stored by the Central Office and contains a handle not used elsewhere. Therefore, the reassignment of handles and reclamation of memory is based on the fact that the number of possible handle values is large enough so that a complete cycle will be significantly longer that the login time of an individual user.

The appearance of the HandleNode/DataNode structure is similar to that of the AddressNode/MessageNode queue from the previous section. The structure is shown below.
List of HandleNodes

```
\ / \\
HandleNode 1 --> HandleNode 2 --> HandleNode N --> NULL
|   |   |   |
V   V   V
DataNode 1.1 --> DataNode 2.1 --> DataNode N.1
|   |   |   |
V   V   V
DataNode 1.2 --> DataNode 2.2 --> DataNode N.2
|   |   |   |
V   V   V
|   |   |   |
V   V   V
DataNode 1.N2 --> DataNode 2.N2 --> DataNode N.N2
|   |   |   |
V   V   V
NULL NULL NULL
```

Figure 4.4 Data Storage Structure

1 The terms HandleNode and DataNode refer to Nodes of a linked list and should not be confused with systems nodes (written in small letters in this section).

2 These DataNodes are also flagged in their ID field as the last nodes.

4.2.7 Interrupt Handling/Idle Routine (SBC68K)

The nodes of the MESS system uses several interrupt driven devices. The DUART that provides serial input/output to the terminal is used in interrupt driven mode to buffer incoming keystrokes. The timer and bus error signal, both of which were used at times during system
development, use interrupts. The LANCE and DSC share an interrupt generated from the interface board.

Of the seven interrupt levels available on the 68000, the SBC68K reserves four. These are assigned as follows: the DUART uses level 4, the Parallel Interface/Timer uses level 2 for parallel I/O and level 5 for the timer, and level 7 is assigned to the software abort button. Interrupt levels 1 and 3 are not used. Interrupt level 6 may be connected to the floppy disk controller or used by an add-on device. (Jumper connections on the SBC68K are required to enable levels 2/5/6 if they are to be used.) For the MESS system, level 6 was chosen for use by the LANCE and DSC. Either device can generate an interrupt. The Interrupt Service Routine (ISR) checks both devices to determine which requested servicing.

All exceptions on the 68000 (this includes hardware interrupts, software interrupts, internal errors, and bus errors) are assigned a location in the interrupt vector table that is located at the beginning of memory. This table occupies the low 400h bytes of RAM: 100h vectors at 4 bytes per vector. Hardware interrupts are assigned an auto-vector in the table. This auto-vector is used unless the device provides an alternate vector during the interrupt/interrupt-acknowledge cycle. The DUART supplies its own interrupt vector; the Timer is programmable to operate in either auto-vector or vector driven mode; the LANCE/DSC combination uses the auto-vector. To trap an exception, the location in the interrupt vector table is set to point to an Interrupt Request handler (IRQ). This is an assembly language routine that performs several necessary functions and calls the Interrupt Service Routine (ISR) for the interrupt/exception. The MESS system has IRQ/ISR pairs for the Timer (level 5), the DUART (level 4), the LANCE/DSC (level 6) and the special Bus Error exception. The function of each IRQ is to mask interrupts, preserve the values of all of the registers used by the ISR, call the ISR, restore the registers, and issue a RTE (return from exception) command to allow the system to continue as before. The function of the ISR is specific to the exception that caused it. The interrupt vectors trapped by the MESS system are shown in the table below.

Table 4.6 Interrupt Vectors

<table>
<thead>
<tr>
<th>Vector</th>
<th>Memory Address</th>
<th>Interrupt Type</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>02h</td>
<td>000008h</td>
<td>Special</td>
<td>Bus Error</td>
</tr>
<tr>
<td>42h</td>
<td>000108h</td>
<td>Vector-driven</td>
<td>DUART-Level 4</td>
</tr>
<tr>
<td>1Dh</td>
<td>000074h</td>
<td>Auto-vector</td>
<td>Timer-Level 5</td>
</tr>
<tr>
<td>1Eh</td>
<td>000078h</td>
<td>Auto-vector</td>
<td>LANCE/DSC-Level 6</td>
</tr>
</tbody>
</table>
The Interrupt Service Routine should perform its task as quickly as possible so that other exceptions, if any, can be serviced promptly. This is important in a system where timed events such as the recording/playback of speech are being performed. For this reason, some actions initiated by an interrupt are deferred and performed outside of the ISR. This is accomplished by using a set of flags that are set by the ISR to indicate that an action is required. These flags are checked, and necessary actions performed, by the idle routine. The idle routine is called whenever the system is waiting for an event such as input from a user or a response from the Central Office to occur.

4.2.8 Terminal Interface

The terminals used for the nodes in the system are standard VT100 compatible terminals. This section describes first the interface between the SBC68K and the terminal, then the higher level routines used to control the terminal. A terminal is connected to the SBC68K Single Board Computer using a serial interface. On the SBC68K two serial input/output channels are provided by an mc68681 Dual-Universal Asynchronous Receiver-Transmitter (DUART).

The DUART is controlled by a number of registers memory mapped into the 68000's address space. The significant registers are shown below. (All are 8-bit registers mapped to odd memory locations.)

<table>
<thead>
<tr>
<th>Address</th>
<th>Read Access</th>
<th>Write Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>FF0001</td>
<td>Mode Register A</td>
<td>Mode Register A</td>
</tr>
<tr>
<td>FF0003</td>
<td>Status Register A</td>
<td>Clock-Select Register A</td>
</tr>
<tr>
<td>FF0005</td>
<td>Command Register A</td>
<td>Command Register A</td>
</tr>
<tr>
<td>FF0007</td>
<td>Receive Buffer A</td>
<td>Transmit Buffer A</td>
</tr>
<tr>
<td>FF000B</td>
<td>Interrupt Status Register</td>
<td>Interrupt Mask Register</td>
</tr>
<tr>
<td>FF0019</td>
<td>Interrupt Vector Register</td>
<td>Interrupt Vector Register</td>
</tr>
</tbody>
</table>

As mentioned above, the DUART provides two serial I/O channels. Channel A is connected to the terminal and Channel B is connected to the host system for downloading code. Channel B has registers corresponding to the Channel A registers listed above at an offset of 10h. The TUTOR monitor that runs from firmware on the SBC68K, and functions as its “operating system,” sets up both channels for 9600 bps, no parity, 8-bit characters, and 1 stop-bit communications. It uses polling mode for corresponding with the DUART.
The MESS system uses the DUART in interrupt driven mode. This is done by setting the Interrupt Vector Register, trapping the appropriate interrupt (see also Section 4.2.7 on Interrupt Handling), and unmasking (enabling) the Receive Character Ready for Channel A interrupt in the Interrupt Mask Register. Other DUART interrupts are not used and remain masked off (disabled). The Interrupt Service Routine (ISR) for the DUART is called each time an incoming character (keystroke) is available. The ISR checks the Interrupt Status Register to verify the source of the interrupt. Next it confirms that an incoming character is available. The incoming character is then read and placed into the input queue. The input queue is a special structure that consists of a read pointer, a write pointer, a status byte, and a wrap-around data buffer. The read pointer follows the write pointer. The status byte is used to indicate the state (Empty/Normal/Full) of the queue. This is important if the read and write pointers are equal. The enqueue routine, called from the ISR, stores a character, advances the write pointer, and sets the status byte. The dequeue routine is used to retrieve characters from the queue in order of arrival. It reads a character, advances the read pointer, and sets the status byte. Both routines use the ckqueue routine before performing any operation and return a failure indicator if the queue cannot accommodate a request. The two routines that provide the interface to the DUART are sio_in and sio_out. The sio_in routine simply checks the input queue (using the dequeue routine) and returns either -1 if it is empty or the first character available. The sio_out routine waits for the DUART's transmit buffer to be available then writes the desired character to it.

The DUART input/output routines are used by higher level terminal control routines to provide display functions and keyboard input. The terminals used support standard ANSI control sequences. These control sequences are special series of characters beginning with the Escape (ASCII 1Bh) character. They cause the terminal to perform special functions such as positioning the cursor, erasing sections of the screen, saving/restoring cursor position, turning on/off bold display, and setting up a scrolling region. A number of routines use these Escape sequences to provide a variety of display functions needed by the system.

The MESS system divides the screen into three regions: a title bar, a text/edit window, and a message window. (See Screen Display Format in Appendix A.) The title bar and text/edit window are written using a combination of three routines. The position_cursor routine is used to move the cursor to a location on the screen. The output routine is used to write a character to the current location. The output routine also interprets the eight-bit character code used for text. The most significant bit is used as a flag to indicate if the text is to be bold/highlighted or not. The remaining seven-bit ASCII value is transmitted to the terminal for display. The display routine combines the functions of the position_cursor and output routines. It writes a string of characters to the terminal beginning at the location specified. The message window is setup as
a scrolling window. The cursor position within the message window is preserved using the save/restore position cursor Escape sequences. The message routine is used by the system to print a string within the message window. Old messages are automatically scrolled up to make room for new messages as they arrive.

Two methods of getting input from the terminal keyboard are used. The read_key routine is used for normal input. It automatically interprets complex keystroke sequences such as those generated by function/cursor-movement keys. It also polls the Idle routine (see also Section 4.2.7 on Interrupt Handling) while waiting for input so that background functions can be taken care of. The second method of checking the keyboard is to use the sio_in routine directly. This is done when a long sequence of events—that the user may wish to abort—is being performed. The sio_in routine is periodically checked to see if the designated abort key was pressed.

4.2.9 The Node to DSC Interface

The recording, playback, and generation of audio on the nodes is done using the Am79c30A ISDN Digital Subscriber Controller (DSC). The DSC communicates with the system using several memory addressable registers. These are used for both control and passing data to/from the DSC. The DSC also has the ability to generate interrupts to signal the system of important events. The DSC is divided into a number of subsections. This section discusses the programming of the two subsections that are key to the MESS system: the MAP and MUX subsections. (See Section 4.1 for more detail on the DSC subsections.)

The setup of the DSC requires a number of different tasks be performed. The DSC is an interrupt driven device. It shares an interrupt with the LANCE. The initialization of the Interrupt Vector Table, etc. required to trap the interrupt is explained in the Interrupt Handling section. The Interrupt Service Routine (ISR) shared by the DSC and LANCE is discussed later in this section. The DSC provides several possible audio inputs and outputs. The following selections are made in initializing the Main Audio Processor (MAP) control registers: the Side Tone Gain path is set for infinite attenuation; the loud speaker output is enabled, the ear piece output is disabled; for input, audio line A is selected and setup for a peak input voltage (from the microphone) of 0.625v. The B-Channel Multiplexer (MUX) is setup so that the MAP (channel Ba) is connected to one of the microprocessor channels (channel Bb). The byte available interrupt for the microprocessor is turned on. All other DSC interrupt sources are turned off. The DSC is initially placed in IDLE mode with interrupts disabled. Output volume is controlled by setting the GER Gain Filter. This is set to 0 dB at startup.
The audio interface for the system is provided by two main routines: record_local and playback_local. A third routine, generate_tone, is used to produce a warning beep by several parts of the system. All these routines put the DSC into ACTIVE mode while they are running and return it to IDLE mode when they have completed. Both the record_local and the playback_local routines enable DSC interrupts and set the volume (using the GER Gain Filter) to the level specified by the system. These routines set several parameters that are accessed by the DSC's ISR and allow it to do the work of loading or playing the contents of the sound buffer. The interrupt driven mode allows the DSC to insure that information is provided at a constant 64 kbps data rate. It does this by interrupting every 0.125 ms to read/write the next byte of data. The record_local routine monitors bit 0 of Port C of the Parallel Interface. This is connected to a push button switch. Closing the switch (a zero) starts the recording process; opening the switch (a one) signifies the end of the recording process. If the switch is held closed longer than the capacity of the sound buffer, the recording process ends when the buffer has been filled. The ISR is used to read the output of the MAP section at 64 kbps and transfer it to the sound buffer. The playback_local routine mutes the microphone input to insure that no interference occurs. It then uses the ISR to copy the contents of the sound buffer to the MAP input at 64 kbps. The generate_tone routine uses the DSC's Secondary Tone Ringer (STR) to generate a short duration 400 Hz tone. The STR does not use either the GER Gain Filter or the B-Channel accessed by the ISR.

The ISR for the DSC is shared with the LANCE. The ISR checks the DSC Interrupt Register to determine if the DSC caused the interrupt. If so, the ISR either copies a byte of data from channel Bb into the sound buffer or from the sound buffer to channel Bb. This in effect directs the data to/from the MAP since during initialization the B-Channel Multiplexer was setup to connect the MAP (channel Ba) to the microprocessor interface (channel Bb). The choice of action performed and the number of times it is be repeated before the ISR signals the main routine that it has completed its task are controlled by parameters that record_local and playback_local set before putting the DSC into ACTIVE mode. If the ISR is called and the DSC Interrupt Register indicates that the DSC did not request service, the ISR precedes to check/process LANCE interrupts.

4.2.10 The Node to LANCE Interface

The Am7990 Local Area Network Controller for Ethernet (LANCE) is used to provide the nodes an interface to the ethernet. The LANCE communicates with the system for data/control in three ways: memory addressable registers, interrupts, and direct memory access (DMA). The LANCE uses two 16-bit directly addressable registers to provide indirect access to its four
control/status registers. The LANCE uses interrupts to signal several different important conditions. It directly access memory to read its initialization block, to poll receive and transmit control structures, and to read/write packets of data. This section describes the programming/interaction with the LANCE using all of these means. (Additional information about the LANCE can be found in the Hardware section.)

The LANCE starts out in STOPPED mode. In this mode, it does not access memory, cause interrupts or respond to incoming packets from the ethernet. In order to initialize the LANCE, it is necessary to construct an initialization block in memory. This consists of twelve consecutive 16-bit words. The initialization block contains a mode register (1 word), address register (3 words), logical address filter (4 words), a receive control structure pointer (2 words), and a transmit control structure pointer (2 words). The mode register allows the LANCE to be put into a variety of "test" modes. The nodes use the LANCE in "normal" mode. (Note that the Central Office uses the LANCE in a special "test" mode.) The address register for a node is set based on the userid of the of the person logged in. This enables the Central Office to selectively send messages to users whichever node they may be using. All addresses used are designated as local following the international addresses standard (see Ethernet Address Conventions in Appendix D). The logical address filter is not used in this system. The receive control structure pointer is set to point to a four buffer receive control structure. The transmit control structure pointer is set to point to a two buffer transmit control structure. (These control structures are not initialized at this point. They are discussed again below.) Once the initialization block is complete, the LANCE is instructed to initialize itself. This is done by writing the address of the initialization block to LANCE control registers, setting up bus interface control registers, and triggering the INIT flag in the main LANCE control register. The LANCE responds by taking control of the system bus, reading the block, and generating an Initialization-Done (IDON) interrupt.

After the LANCE has been initialized, it waits to be told to START before it begins polling its control structures in memory or processing packets from the ethernet. First the LANCE control structures and data buffers must be prepared. The control structures consist of a number elements (always a power of 2) located sequentially in memory. Each element contains four consecutive 16-bit words that hold information about and a pointer to a data buffer. The control structures must be aligned on a 32-bit boundary in memory. Four receive buffers are used in this system. The receive buffers are allocated dynamically from free memory (See Section 4.2.4 for memory allocation details) but remain constant sized throughout execution. The receive control structures are initialized by setting the ownership bit to LANCE ownership, writing the buffer size into the appropriate location, and setting the pointer to the corresponding receive buffer location. Two transmit buffers are used in this system. They are allocated in the same
manner as the receive buffers. The transmit control structures are initialized by setting the ownership bit to USER ownership, and setting the pointer to the corresponding transmit buffer location. The receive and transmit buffers are made the maximum ethernet packet size: 1518 bytes for receive buffers and 1514 bytes for transmit buffers. The 4 byte difference is due to the CRC which is generated by the LANCE during transmission and added to the packet.

Once the transmit/receive control structures are setup, the LANCE is started by triggering the START flag in the main LANCE control register. The LANCE then will poll the control structures every 1.6 ms and monitor the ethernet for packets addressed to it. If the LANCE finds that the current transmit buffer is set to LANCE ownership, it will process and transmit that data. If the LANCE receives a packet from the ethernet, it will write that packet to the first receive buffer that it owns; however, it the LANCE does not own any receive buffers the packet will be lost. While the LANCE is running, there are six different conditions that can cause an interrupt. Three are errors that should not occur under normal operation: BABBLE indicates that the transmitter has been sending out data for longer than the time required to transmit a maximum length packet; MISS indicates that a packet was lost because the LANCE could not locate an available receive buffer to store it in; MERR indicates that the LANCE was unable to communicate with memory during a DMA operation. The interrupts all signal that an event has occurred: IDON indicates that initialization of the LANCE is complete; TINT indicates transmission of a packet is either complete or failed; RINT indicates that a packet was received. The action taken as a result of the last two interrupts should be to check the control structure for a current buffer to determine exactly what happen to cause the interrupt.

The basic packet transmission routine for the system is xmit_data. It is called by the higher-level SendPacket routine that is responsible for implementing the command protocol. The task of xmit_data is to build a complete packet within the current transmit buffer and signal the LANCE to transmit it. The LANCE expects to see four fields in an ethernet packet destined for transmission. These are a 6-byte destination address, a 6-byte source address, a 2-byte length field, and a 50-1500 byte data field. During transmission the LANCE computes and appends a 4-byte CRC field. In the MESS system, the first data byte is always the command identification byte. The xmit_data routine must be flexible enough to allow use in a variety of different scenarios. It sets the address fields using the command byte to determine what type of packet is being sent (a message destined for another node is treated differently than a standard request of the Central Office). It uses packet sequencing specified by the protocol and maintained by SendPacket. It automatically pads packets that are shorter than the minimum length. It even allows the calling routine to directly write data to the transmit buffer if that data cannot be easily merged and assigned a pointer. In addition to creating a packet, xmit_data must first check that it (the USER) owns the current transmit buffer and must setup the transmit

60
control structure so that the LANCE will transmit the packet. Instructing the LANCE to transmit a packet is done by writing the size of the packet to the control structure, setting several flags within the control structure, and finally setting the ownership flag for the transmit buffer to LANCE ownership. The LANCE will discover this and transmit the packet as soon as it polls the control structures. (If the delay of up to 1.6 ms is too long for a certain situation, it is possible to trigger the LANCE to immediately look at the transmit control structure using a flag in one of the control registers.)

The routine that handles incoming packets is ProcessIncomingPacket. This routine is called by the idle routine. (See discussion of ISR below.) It has three main tasks: to verify the incoming packet, to provide processing of certain packet types, and to inform the routine that solicited a response from the Central Office what the result is. Verification of the packet includes checking its length, that it is properly addressed, that its CRC matches with one computed for the packet, and that it contains a valid packet id. A check is also made to insure that the packet was, in fact, requested by the node. (The protocol allows one exception where an unsolicited packet is accepted. See Section 4.2.2.) Processing jobs such as extracting the current message count, extracting a handle value, or storing a message to the local message queue are also performed by the ProcessIncomingPacket routine. For several types of packets, information of interest to the initiating routine is recorded. In all cases, a result code is set for use by SendPacket in maintaining the protocol.

The Interrupt Service Routine (ISR) for the LANCE is shared by the DSC. The ISR first checks to see if the DSC caused the interrupt. If not, it checks each of the six possible sources of a LANCE interrupt. The three fatal error conditions (BABBLE, MISS, MERR) are simply reported. The IDON interrupt sets a flag that indicates to the LANCE initialization routine that it may proceed. The RINT/TINT interrupt likewise set flags. These flags are used in the idle routine. The ISR resets the interrupt causing condition before it gives up control. The RINT flag is used to tell the idle routine to call ProcessIncomingPacket. The TINT flag is simple reset by the idle routine as the xmit_data routine uses the ownership bit in the control structure to tell if a packet has been sent.

The LANCE is a critical part of the MESS system. Its low-level interface to the rest of the system has been described here. Additional details on interfacing the LANCE may be found in the Hardware section. Additional information on the Node/Central Office Protocol and the interrupt mechanism can be found in separate parts of this section.
4.2.11 The Central Office Ethernet Interface (Including DMA and Interrupt Mechanisms)

The Central Office System uses an Allied Telesis AT1500 commercial ethernet card. This card uses the Am79c960 Single Chip Ethernet Controller for ISA (Ethernet Controller) which is a superset of the Am7990 LANCE used by the nodes. The setup and uses of the Ethernet Controller are similar to that of the LANCE. This section describes the programming of the Ethernet Controller. Similarities and differences versus programming the nodes ethernet interface--due to the PC environment, the Ethernet Controller vs. the LANCE, and the tasks of the Central Office--are pointed out.

The Ethernet Controller interacts with the system in three ways: it uses a 24 byte block of I/O space for conventional I/O register access; it provides interrupts to signal certain conditions, and it uses direct memory access (DMA) to read/write data directly from memory. The PC, unlike the SBC68K, uses separate memory and I/O address spaces. The AT1500 maps its 24 byte block of I/O space to one of four starting addresses: 300h, 320h, 340h, 360h. [The Central Office currently uses 320-338h.] The 24 byte block is divided up into 16 bytes reserved for an address PROM to store a permanent ethernet address for the board, a 2-byte reset location, a 2-byte vendor specific word (not used by the AT1500), and three 2-byte (16-bit) ports for accessing internal registers. Two of these directly addressable ports provide indirect access to internal registers in the same manner as on the LANCE. The third port provides indirect access to internal ISA-bus configuration registers. The Ethernet Controller makes available the same four control/status registers as the LANCE but also defines some 90 additional internal registers.

The interrupt mechanism provided by the Ethernet Controller functions in the same manner as that of the LANCE. It has the same six interrupt causes as the LANCE: BABBLE, MISS, MERR, IDON, RINT, and TINT. (These are defined in the previous section.) A problem with the AT1500 and certain Programmable Interrupt Controllers (PICs) forced a different solution to be used by the Central Office than that used by the nodes. The Ethernet Controller interrupt mechanism is disabled and a timer interrupt, that repeats 18.2 times a second, is used to monitor each of the interrupt causing flags in the main status register. The timer interrupt, generated by the PC's real time clock, is trapped by using the DOS get_vector and set_vector functions to assign its interrupt vector to the Ethernet Controller Interrupt Service Routine (ISR).

As does the LANCE, the Ethernet Controller accesses memory to read its initialization block, to poll receive and transmit control structures, and to read/write packets of data. On the PC, however, the Ethernet Controller must go through a DMA controller that moderates requests from all devices. The PC uses two i8237A (or compatible) DMA controllers cascaded together. This provides a total of seven DMA channels: four from each i8237A minus the one reserved to
perform the cascade function. The channels are numbered from 0-7 with channel 2 reserved for the floppy disk drive and channel 4 reserved for the cascade function. The two i8237As are addressed into I/O space as shown in the Section 4.1 (Hardware). The low controller is setup for 8-bit operation and therefore requires half as much I/O space as the high controller which is setup for 16-bit operation. The PC's BIOS initializes the DMA controllers; therefore, the Central Office system needs only to unmask the DMA channel used by the Ethernet Controller and set that channel to use cascade mode. This is accomplished by writing to the Mode and Mask registers of the controller for the correct channel. Macros are used to find the correct addresses and channel values from an absolute channel value (0-7). [The AT1500's DMA channel usage is programmable. Currently, it uses channel 5.]

The initialization procedure for the Ethernet Controller is nearly the same as that described for the LANCE in the previous section. The differences are as follows. The Central Office uses the Ethernet Controller in a special mode, called Promiscuous Mode. This is set in the mode register of the initialization block. This mode allows the Central Office to receive all packets regardless of destination address. It then determines which packets should be processed and which should be discarded. The receive/transmit control structures are the same as for the LANCE. One difference in initialization is required because of the PC's segmented architecture. A routine called far_to_absolute is used to convert addresses from Segment:Offset in normal PC operations to absolute 24-bit addresses for the Ethernet Controller to use in DMA operations. Once running the Ethernet Controller will poll its transmit/receive control structures every 1.6 ms and will monitor the ethernet for incoming packets.

The Central Office system has an xmit_data routine that is similar to the one used by the nodes. The Central Office does not have a higher-level SendPacket-like routine since the nodes are responsible for maintain the protocol and the Central Office simply responds to the current request. (The Central Office is required to maintain packet sequencing information, etc. associated with the protocol.) The Central Office's xmit_data routine builds an ethernet packet in the same manner as the node based routine does. The major difference is what is used for source and destination addresses.

The Central Office system also has a ProcessIncomingPacket routine that (like the nodes' routine) handles all incoming ethernet traffic. This routine is called from the Central Office's main wait loop whenever a flag variable is set by the ISR to indicate that a packet has been received. The function of the routine is to verify the packet then process it and provide a response to the requesting node. The first steps in verifying a packet are to check its size, addressing, and CRC value. If a packet is a valid size, it is addressed to the Central Office, and its CRC matches with a computed one, the packet is accepted. Next, the packet sequencing and command field are checked. From the sequencing, the Central Office determines if the request
is a repeated one. (This can be caused by the Central Office’s response being lost/corrupted during transmission.) The decision on how to process the packet is made based on the command field and whether the request is a repeated one. Every valid command packet will cause the Central Office to send a response packet. (The commands/responses are discussed in the Node/Central Office Protocol section.)

The ISR for the Ethernet Controller checks each of the interrupt causing flags in the main status register. (These are set by the Ethernet Controller regardless of the fact that its interrupt mechanism is disabled and a timer interrupt is used to trigger the ISR.) It sets a flag variable to be used in the Central Office’s main wait loop if any "interrupts" are pending. It then clears the condition in the control/status register. The three error conditions (BABBLE, MISS, MERR) are simply reported when their flag variable is set. The RINT flag variable is used as a signal to call ProcessIncomingPacket. The TINT flag variable is not used. The IDON flag variable is used as a signal to the initialization routine that it may proceed.

The Am79c960 Ethernet Controller is programmed in much the same way as the LANCE. Some differences, due primarily to environment differences, do exist. Additional details on the AT1500, the host for the Ethernet Controller, may be found in Section 4.1 (Hardware).
Summary and Conclusions

Chapter 5 contains results from the testing of the e-Mail with Embedded Speech System (MESS) that was developed for this thesis. The testing methods for system functions such as the protocol and speech compression are explained. Alternate methods considered--and in some cases experimented with--for different aspects of the system are discussed. Difficulties encountered during the development of the project are also described.

5.1 Communications System Testing

The communication system between the nodes and Central Office system are based on ethernet controllers in both the nodes and the Central Office system. These are controlled by low level interface software and a protocol designed to handle transmission errors and other problems with the ethernet connection. Testing of these was divided into verification of the hardware design, and testing of the protocol.

5.1.1 Ethernet Hardware Verification

The Local Area Network Controller for Ethernet (LANCE) used by the Nodes and the Ethernet Controller on the Allied Telesis 1500 board used by the Central Office system both provide several levels of loopback testing. These test features were used to verify the operation of the ethernet hardware on each of the systems.

5.1.1.1 Ethernet Hardware – Nodes

The initial phase of the development of the ethernet system on the nodes was to interface the LANCE chip to the 68000 bus. The LANCE was designed primarily to interface with a bus that uses multiplexed address/data lines. Its design is flexible enough, however, that it can be adapted to work with many different bus configurations. Latches are used to demultiplex the LANCE bus for compatibility with the 68000 bus. In addition, a PAL (Programmable Array Logic) device and external flip flops are used to resolve differences between the LANCE's Bus
Mastering interface and that expected by the 68000. Initially, the bus slave mode interface was tested to insure that the control/status registers of the LANCE could be accessed. Next, the bus mastering mode of the LANCE was tested by instructing the LANCE to initialize itself -- which it does by reading an initialization block in memory. The final stage of interfacing the LANCE was to use its internal loopback test to send known packets through the LANCE. The "received" packet was then compared to the original to verify that the LANCE interface was operating correctly. The next stage of the design was to add the Serial Interface Adapter (SIA) chip to the node interface board. This device provides the interface between the LANCE and the Transceiver that connects to the ethernet itself. The Serial Interface Adapter was tested using the LANCE's external loopback test. This was done by connecting the Tx (Transmit) and Rx (Receive) pairs together (after they had passed through an external pulse transformer that provides isolation of the SIA/LANCE from the Transceiver). This test allowed known packets to be transmitted from the LANCE through the SIA, back into the SIA, and then received by the LANCE. The Cl (Collision) pair, the third wire pair that makes up the SIA to Transceiver interface, was left unconnected during this test. During normal operations, the Transceiver provides a "Heartbeat" signal, on this pair, to the SIA/LANCE to verify that it is working properly. The absence of this "Heartbeat" signal is not a fatal error; therefore, it was possible to conduct the test while ignoring the warning flag that signaled a possible Transceiver problem. The final hardware design phase for the Nodes' ethernet interface was the design of the Transceiver. This is an external circuit, isolated from the SIA/LANCE by a set of pulse transformers, that requires its own -9v dc power supply. The main component in the design of the transceiver was an Am7996 device. This comes with an application note, from Advanced Micro Devices, that specifies a variety of parameters that must be met within tight tolerances for the Transceiver circuit to operate properly. The Transceiver circuit was first tested using an external signal source and an oscilloscope to verify that it properly received and attenuated a signal. Once this was confirmed, the Transceiver was tested using a loopback connector on its bnc ethernet connector. The Transceiver failed the external loopback test in this configuration. (See Section 5.3 for further discussion of the Am7996-based Transceiver.) The Am7996-based Transceiver was replaced with a commercial Transceiver, the AT&T ST-500. This Transceiver enabled the system to pass the complete--to the ethernet--external loopback test. The AT&T ST-500 was then used for tests in which known packets were sent to/from the Central Office System. This provided the final verification that the communication system was operating as expected.
5.1.1.2 Ethernet Hardware – CO System

The Ethernet Controller used for the Central Office System is a commercial product: the Allied Telesis 1500. Therefore, no hardware design was required for this interface. The board was designed to be used with device drivers provided by Allied Telesis for various commercial LAN products. For this project, low level interface code was developed that directly controlled the Ethernet Controller on the Allied Telesis board. The operation of the board was initially verified using the Internal Loopback test to send known packets through the Ethernet Controller. This verified the DMA controller was setup correctly and the board's interface to the 80486 PC was operating properly. (A problem with the board and/or the Programmable Interrupt Controller(s) in the 80486 PC was encountered. This is discussed in Section 5.3.) To verify the operation of the board's interface to the ethernet, a loopback connector was constructed and attached to the board's bnc connector. This allowed the use of the external loopback test to send known packets through the transmit side of the Ethernet Controller, onto the ethernet, and back through Receive side of the Ethernet Controller to the 80486 PC. The final verification of the Allied Telesis board was done by sending packets to and receiving packets from one of the system's nodes.

5.1.2 Protocol Testing

The purpose of the Node/Central Office protocol is to provide both a command interface for the nodes to request service of the Central Office System and a method of handling transmission errors and other problems associated with the ethernet. In the system designed for this project, the ethernet is not heavily loaded and the transmission distances are short. Transmission errors are thus too infrequent to test the protocols ability to handle them. In order to fully test the protocol, the CRC check routine was given the ability to intentionally report errors to the protocol. Initially, the protocol was tested to insure that under error-free conditions it provided the functionally expected. Next, its ability to time-out, when the Central Office System is unavailable, was tested. This was done by disabling the Central Office System but continuing to make requests of it from the nodes. The protocol successfully handled this situation and resumed operating with the Central Office System as soon as it was brought back on-line. The next test was to generate errors using the modified CRC check routine. First errors were generated at the Central Office System. The Central Office System communicated to the node that a flawed packet was received. As specified in the protocol (See Section 4.2.2), the system handled single and double errors by repeating the request until it was successful, and sequences of errors caused the node to give up and report to the user that the Central Office System was
experiencing a problem. When errors were generated at the nodes, or at both systems at once, the Central Office System was able to correctly identify which packets contained requests it had already processed and needed only to repeat its response for the node. Again, after multiple failures the node will give up and report a Central Office System problem to the user. During testing, the protocol handled all of the simulated problems as expected. Because of the scope of this project it was not possible to test the protocol under a heavily loaded system. There is one known possible weakness in the protocol under such conditions. It is explained, and a solution suggested, in Section 4.2.2.

5.2 Speech Coding

The speech compression algorithms used by the Central Office System were tested to determine their effectiveness. This section contains the results of comparative testing of the final algorithms incorporated into the system. This testing was done using speech samples generated by the nodes and sent to the Central Office System for storage. Following the comparative testing results, individual discussions of each of the speech compression methods are presented.

Four things were taken into consideration when judging the algorithms: the encoding/decoding time, the amount of compression achieved, the subjective quality of the output speech, and the intelligibility (again judged subjectively) of the output speech. The ratings used for speech quality and intelligibility are shown below.

<table>
<thead>
<tr>
<th>Rating</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Excellent</td>
<td>Equal to Original</td>
</tr>
<tr>
<td>Good</td>
<td>Easily Understandable</td>
</tr>
<tr>
<td>Fair</td>
<td>Understandable</td>
</tr>
<tr>
<td>Mediocre</td>
<td>Barely Understandable</td>
</tr>
<tr>
<td>Poor</td>
<td>Unintelligible</td>
</tr>
</tbody>
</table>
Table 5.2 Quality Ratings

<table>
<thead>
<tr>
<th>Rating</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Excellent</td>
<td>Indistinguishable from the Original</td>
</tr>
<tr>
<td>Good</td>
<td>Slightly Different from Original</td>
</tr>
<tr>
<td>Fair</td>
<td>Distinctly Different from Original</td>
</tr>
<tr>
<td>Mediocre</td>
<td>Distinctly Different and Noisy</td>
</tr>
<tr>
<td>Poor</td>
<td>Noisy and Distorted</td>
</tr>
</tbody>
</table>

To provide well-rounded testing of the algorithms, a number of different types of utterances were used and the results for each compression method compared. The utterances used are shown below. Each labeled utterance was spoken once and stored in a local buffer on one of the nodes. It was then sent to the Central Office for storage once for each method under test. In between storage requests the Central Office was setup to use a different compression method. The stored speech signals were then retrieved and played back in order to judge the quality and intelligibility of the speech resulting from each method. The data storage requirement for the encoded speech shown below is that reported by the Central Office System. It should be noted that this figure contains a small amount of overhead (non-speech); therefore, the compression ratios are not exact.

Table 5.3 Utterances

<table>
<thead>
<tr>
<th>Label</th>
<th>Utterances</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>&quot;Obey&quot;</td>
<td>Spoken Normally</td>
</tr>
<tr>
<td>B</td>
<td>&quot;The quick brown fox&quot;</td>
<td>Spoken Normally</td>
</tr>
<tr>
<td>C-1</td>
<td>&quot;It is raining&quot;</td>
<td>Spoken Normally</td>
</tr>
<tr>
<td>C-2</td>
<td>&quot;It is raining&quot;</td>
<td>Spoken Normally</td>
</tr>
<tr>
<td>C-3</td>
<td>&quot;It is raining&quot;</td>
<td>Spoken Loudly</td>
</tr>
<tr>
<td>C-4</td>
<td>&quot;It is raining&quot;</td>
<td>Spoken Softly</td>
</tr>
</tbody>
</table>

The first test conducted compares the results of compressing a single word versus compressing a phrase. This test was intended to give an indication of how much each compression method can take advantage of the silence between words. (Although not shown here, the AVQ and LPC algorithms can achieve high compression ratios on signals with large periods of silence. No silence was allowed in the test utterances—except normal breaks in
speech—so as to not skew the results.) Interestingly, the LPC algorithm was able to compress the single word utterance better than the phrase utterance. The mostly likely explanation for this is that the phrase utterance contains several fricatives ('quick' and 'fox') that may be more difficult to compress since they are generated by forced air that is essentially equivalent to white noise, rather than by the vocal cords. The single word utterance does not contain any fricatives. The third utterance used also contains two fricatives ('it' and 'is') which may explain why the LPC algorithm achieves the best compression on the single word utterance. The second test was designed to compare the results of the compression algorithms on the same utterance spoken two different times. The compression ratios were similar, as expected. There was a slight difference in quality which is most likely due to the first utterance being spoken slightly less clearly and the difference being magnified by the compression process. The third test was designed to see how the algorithms responded to different input volumes. The same utterance was spoken three times: once normally, once loudly, and once at a whisper. As expected, the AVQ and LPC algorithms were better able to compress the lower resolution (whispered) signal. The quantization noise from the compression process also was more evident in the whispered sample. For the ADM algorithm, the speech came through the noise better for the whispered signal than for any of the other test utterances. This indicates that the ADM algorithm can better track a lower volume signal which is consistent with its design.

Table 5.4 Phrase/Single-Word Compression Results

<table>
<thead>
<tr>
<th>Utterance</th>
<th>Normal</th>
<th>ADM</th>
<th>AVQ</th>
<th>LPC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Size</td>
<td>Size/Ratio</td>
<td>Intelligibility</td>
<td>Quality</td>
</tr>
<tr>
<td>A</td>
<td>10395</td>
<td>1330/(8:1)</td>
<td>Mediocre</td>
<td>Poor</td>
</tr>
<tr>
<td>B</td>
<td>16335</td>
<td>2090/(8:1)</td>
<td>Mediocre</td>
<td>Poor</td>
</tr>
</tbody>
</table>

Table 5.5 Repeated Utterance Compression Results

<table>
<thead>
<tr>
<th>Utterance</th>
<th>Normal</th>
<th>ADM</th>
<th>AVQ</th>
<th>LPC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Size</td>
<td>Size/Ratio</td>
<td>Intelligibility</td>
<td>Quality</td>
</tr>
<tr>
<td>C-1</td>
<td>10395</td>
<td>1330/(8:1)</td>
<td>Mediocre</td>
<td>Poor</td>
</tr>
<tr>
<td>C-2</td>
<td>10395</td>
<td>2090/(8:1)</td>
<td>Mediocre</td>
<td>Poor</td>
</tr>
</tbody>
</table>
Table 5.6 Volume Variation Compression Results

<table>
<thead>
<tr>
<th>Utterance</th>
<th>Normal Size</th>
<th>ADM Size/Ratio</th>
<th>Intelligibility</th>
<th>Quality</th>
<th>AVQ Size/Ratio</th>
<th>Intelligibility</th>
<th>Quality</th>
<th>LPC Size/Ratio</th>
<th>Intelligibility</th>
<th>Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-2</td>
<td>10395</td>
<td>1330/(8:1)</td>
<td>Mediocre</td>
<td>Poor</td>
<td>6130/(1.7:1)</td>
<td>Excellent</td>
<td>Excellent</td>
<td>4788/(2.1:1)</td>
<td>Excellent</td>
<td>Excellent</td>
</tr>
<tr>
<td>C-3</td>
<td>10395</td>
<td>1330/(8:1)</td>
<td>Mediocre</td>
<td>Poor</td>
<td>5882/(1.8:1)</td>
<td>Excellent</td>
<td>Excellent</td>
<td>4756/(2.2:1)</td>
<td>Excellent</td>
<td>Excellent</td>
</tr>
<tr>
<td>C-4</td>
<td>10395</td>
<td>1330/(8:1)</td>
<td>Fair</td>
<td>Poor</td>
<td>2762/(3.8:1)</td>
<td>Good</td>
<td>Fair</td>
<td>3145/(3.3:1)</td>
<td>Excellent</td>
<td>Good</td>
</tr>
</tbody>
</table>

5.2.1 Adaptive Delta Modulation

The Adaptive Delta Modulation (ADM) algorithm is intended to be a quick, simple compression scheme. It provides the best compression ratio and the shortest execution time of the algorithms. This is made up for by poor quality output speech. The ADM algorithm yields a constant 8:1 compression ratio. In tests, the output speech was both noisy and distorted. It was intelligible through the noise in all cases. With some improvements to reduce the noise and distortion, a usable ADM algorithm is possible; however, this method is not acceptable for a system that requires good quality speech reproduction.

5.2.2 Adaptive Vector Quantization

The Adaptive Vector Quantization (AVQ) algorithm is a general compression scheme, not unique to speech compression. It has a programmable tolerance that can be changed to alter the compression versus quality trade-off. For this system the tolerance was set low enough that little distortion of the signal was expected. In tests, the AVQ algorithm produced good quality speech. In almost all cases the results matched the original utterance closely; however, in a few cases the AVQ algorithm slightly distorted fricatives. (Hissing sounds such as made by f and s.)

The compression ratios yielded by the AVQ algorithm varied from as low as 1.6-to-1 up to 3.8-to-1. This was slightly less than expected based on tests of the AVQ algorithm using a PC based sound card. The reason for the drop is most likely an increase in background noise resulting from a less isolated recording method. The AVQ algorithm provides one major benefit over the LPC algorithm. The AVQ algorithm does most of its work in the encoding process, thus its compression time is similar to that of the LPC algorithm. In decompressing the signal for playback, the AVQ algorithm is the equal of the ADM algorithm. The LPC algorithm takes significantly longer. Quick playback is considered a significant advantage since playback is likely to be more frequently used than record and expected--by the user--to occur without delay.
5.2.3 Linear Predictive Coding

The Linear Predictive Coding (LPC) algorithm is specific to speech compression. It is based on many of the same principles that drive low-bit rate coders used in industry. In tests, the LPC algorithm produced good quality speech. In some cases, slight changes in volume were noticed; however, the overall quality and intelligibility remained good. The LPC algorithm took longer to encode and decode signals than the other algorithms. This difference was not that great, however. The compression ratios yielded by the LPC algorithm were more consistent and generally better than those of the AVQ algorithm. They ranged from 2.1-to-1 up to 3.8-to-1. As with the AVQ algorithm, these results were slightly less than those produced during the development of the algorithm using a PC based sound card. This is attributed to higher background noise generated during the recording process on the nodes. In the case of the LPC algorithm, unlike with the AVQ algorithm which starts with an empty codebook each time, some difference could result from the quality of the VQ codebook computed for the different data sets. The compression ratios of the LPC algorithm could be improved significantly by using more advanced coefficient quantization and residual compression techniques.

Some testing of different methods for use with the LPC algorithm was done during the project. The results of different methods of quantizing the prediction coefficients were compared. As a basis for comparison, in one test the prediction coefficients were preserved as is and used for decoding. The only quantization method that noticeably degraded the signal was direct quantization of the prediction coefficients. Both the linear and non-linear methods of quantizing the reflection coefficients produced output signals indistinguishable from that produced without any quantization. It is expected that the advantage of the non-linear approach would become noticeable as the compression ratio was increased. This was not verified, however. Several different methods of compressing the residual signal were tested to determine their effect on the output produced. The first set of tests was designed to test the criteria that the peaks of the residual are its most critical part. Zeroing all values, other than the peak itself, produced an intelligible, but low quality result. Zeroing all values, other than the five element vector that included the peak, produced an intelligible and decent quality result. A second set of tests involved finding the optimal center-clipping threshold. It was found that--without using any other techniques--approximately 25% of the residual could be removed with significant loss of quality. Intelligibility of fricatives in the signal was reduced slightly by the center-clipping. A third test was done using a simple voice encoding technique. The residual was removed entirely and replaced with a sound generating function created at the decoder. The only information passed to the decoder was whether the signal was voiced or unvoiced in a specific region. This was
used to create a residual signal that contained a constant pitch pulse in the voiced regions and a randomly generated noise signal in the unvoiced regions. This method produced an intelligible output, but one that did not sound like the original speaker. For this test, the window for which prediction coefficients are calculated had to be reduced to 10 ms and the thresholds for the voiced/unvoiced decision had to be occasioned tuned to a particular signal. This method if improved upon--by adding a pitch detector and improved voiced/unvoiced logic--would be a legitimate LPC Vocoder type of compression scheme.

5.3 Difficulties

A number of difficulties were encountered during this project. Those that are considered major--that caused an alternate approach to be employed or resulted in significant delay to the completion of the project--are discussed in this section. In addition some difficulties that were less major, but have the potential to cause major problem if overlooked are mentioned.

The design of the node hardware was the source of many difficulties. The interface of the Am79c30a Digital Subscriber Controller (DSC) required the construction of a simple state machine to match the 68000 bus to the signals/timing expected by the DSC. Meeting the requirements of the 68000 bus proved difficult while designing this interface for the first of the two nodes. For the second node, the established design was ported to a Programmable Array Logic (PAL) device. Few difficulties were encountered in this process. The design of a microphone to meet the specifications of the DSC audio-in port also caused a problem. The initial solution was an electret microphone element based circuit. The circuit provided external filtering and significant gain. This microphone circuit was used for testing during much of the project; however, it provided too noisy a signal to be used for the final system. The problem was that power/ground line noise was amplified, along with the signal, producing significant noise at the input to the DSC. For the final system, a telephone handset interface was constructed that allowed the handset microphone to be used for recording audio. The power for the telephone was provided by a 9v battery, and an audio isolation transformer was used to insure that power/ground line noise from the rest of the system was isolated from the audio input. This solution produced a much clearer signal than the previous setup.

As mentioned in Section 5.1.1.2, the development of a Transceiver circuit based on the Am7996 was discontinued and a commercial AT&T ST-500 was used in its place. The design of this Transceiver circuit caused numerous difficulties. The application note that accompanied the Am7996 specified a number of parameters that had to be tuned for the complete circuit to work correctly. Some of these--involving the parasitic capacitance of certain wire paths--proved difficult to accomplish on a wire-wrap prototype. The external attenuation of the circuit was
measured using a known signal and an oscilloscope. This was found to be within the specified tolerance. Initially, the Transceiver was powered by a 9v dc wall transformer. Using this power source, the Transceiver failed all tests. It was suspected that this wall transformer did not provide proper isolation from the power used by LANCE/SIA. Using a 9v dc battery to power the Transceiver resulted in some improvements, but still not correct operation. Using the LANCE initiated external loopback test, the Transceiver reported collisions during every transmission attempt. The transmission levels were measured and appeared to be correct; however, this problem could not be eliminated. In testing, with the Transceiver connected to a functioning Ethernet, it was able to correctly receive packets. Transmission tests, on the other hand, continued to fail as in earlier tests. Eventually, it was concluded that this design was not well suited to a wire-wrap prototype system, and the AT&T ST-500 Transceiver was substituted.

The interfacing and programming of the Am7990 LANCE itself caused several difficulties. Problems with this circuit were the most difficult to deal with because of the LANCE's use of Bus Mastering to perform Direct Memory Access. Errors often caused the LANCE to read/write random areas of memory causing a wide variety of problems not easily traceable back to the LANCE. The most common software problem related to the LANCE was incorrect manipulation of the address of LANCE control structures in memory. An example of such a problem occurred when the address of the highest LANCE receive buffer first was moved into memory above the FFFFh memory boundary by the addition of unrelated code. The control register that contained the upper 4 bits of the 20-bit memory address stored by the LANCE had not been programmed correctly; therefore, the LANCE used only the lower 16-bits as the address in which to store received packets. As long as the desired address was below the FFFFh boundary no problems occurred; however, when the desired address crossed this boundary, the LANCE began writing received packets into the middle of the interrupt vector table. The Bus Master interface of the LANCE caused a difficult to find problem as well. The pull-up resistor pack initially used on the node interface board turn out to have resistors values that were higher than that specified by the LANCE data sheet. Under normal operations, this did not cause any problems. Under certain circumstances, the timing of the transition of the line that the LANCE uses to request the bus was altered enough that the LANCE would release the bus, but the external logic that communicated that to the 68000 bus would not be reset. This resulted in a bus with no master and the 68000 board ceasing to operate until power was cycled. Replacement of the resister pack corrected this problem.

The Allied Telesis 1500 Ethernet board (AT1500) used in the Central Office System was responsible for one major difficulty. When multiple interrupts occur in succession on the AT1500, the Programmable Interrupt Controller (PIC) in the 80486 PC does not appear to function correctly. When the board is first initialized an IDON interrupt is correctly generated;
however, when a TINT or RINT interrupt is generated the Interrupt Service Routine (ISR) is entered repeatedly until the stack overflows. A number of things were attempted to prevent this from occurring. Interrupts were disabled on the AT1500 immediately upon entering the ISR. The AT1500 Interrupt on the PIC was also masked off. The Interrupt Flag for the 80486 was even cleared. None of these, or other attempted solutions, prevented the repeated entering of ISR. It is evident that this occurs before the ISR actually gets the opportunity to perform any commands. Allied Telesis has acknowledged that this problem has been observed under certain (rare?) conditions. In order to avoid this problem, a timer interrupt is used to trigger a polling routine that controls the Ethernet board in place of the IRQ driven method. No problems were encountered using this method. (See also Section 4.2 - Software.)

A problem was encountered due to the nature of the indirect register access provided by both the LANCE and the DSC. This type of problem could occur with any interrupt driven device that uses indirect register accesses. To access indirect registers, the address of a specific register is written to the directly accessible Address Register. The contents of the indirect register are then mapped to the directly addressable Data Register. A problem can occur when an interrupt occurs in between the writing of the Address Register and the access of the indirect register through the Data Register. This is caused when an Interrupt Service Routine (ISR) uses the Address Register/Data Register to access information—stored in an indirect register—that it needs. Upon returning from the interrupt, the Address Register may no longer be pointing to the indirect register to which it was assigned in the previous command. This can cause major problems if arbitrary data is written to control registers. This problem was rectified by masking interrupts during accesses to indirect registers. Another solution would be to save and restore any address registers accessed in an ISR in the same manner as the microprocessor's registers.

5.4 Conclusions

The MESS system designed for this thesis serves as a reasonable prototype on which to base the design of a phone terminal. During the development of the system, a number of design choices had to be made. This section contains a discussion of which design choices proved to be good and which should be reconsidered in future phone terminal systems.

In a final system, one speech compression algorithm would have to be selected as the standard or default choice. As they stand now, the Adaptive Vector Quantization (AVQ) algorithm merits serious consideration. Its achieves compression ratios slightly worse than that of the Linear Predictive Coding (LPC) algorithm; however, it equals the LPC algorithm in quality and intelligibility of the output, and it performs the playback function (which is more critical to users than the record function) more quickly. If the algorithms were to continue to be developed,
the choice would have to be the LPC algorithm. The ADM algorithm could be improved to give decent quality results, but will never match those of the other algorithms. The AVQ could be improved somewhat by fine-tuning the vector-size and tolerance. The LPC algorithm has a great deal more room to grow. The LPC coefficients are currently compressed 4:1. It is possible to improve that ratio to at least 6:1 without losing quality. The compression of the residual can be increased to as much as 10:1 with the application of more advanced techniques. This means that without resorting to voice encoding (where the residual is removed entirely) the compression ratio of the LPC algorithm could be at least doubled. A trade-off between compression time and the compression ratio would have to be made, but a more advanced LPC algorithm would easily beat the AVQ algorithm.

The choice of the hardware used for a future system also should be considered. The choice of the Am7990 LANCE and Am7992 SIA to provide the ethernet interface proved to be a good one. The incorporation of the LANCE into the design of a single microcontroller board, as opposed to being added via an external bus interface, would simplify the bus master interface. The choice of the 68000 as the controller for phone terminal should be reconsidered. The Motorola 68302 Multiprotocol Processor provides a number of items on-chip that require additional components in the current system. The DUART, used for interfacing to the terminal, has been incorporated into the processor itself. In addition, the ISDN S-bus interface can be provided by the 68302. This would allow the Am79c30a DSC to be removed. In its place, an audio interface capable of performing Analog-to-Digital conversion and Digital-to-Analog conversion would be required. The 68302 provides on-chip RAM, but not a sufficient amount to eliminate the need for external RAM for this application. [The Motorola 68302 was not available at the beginning of this project. Now a single board computer based on the 68302 is even available from Arnewsh, the maker of the SBC68K.] A future phone terminal based on the 68302 could be significantly smaller than one based on the 68000, and would provide a more uniform programming interface.

The MESS system certainly does not represent a state of the art commercial product. There are a number of improvements that could make it smaller and better. It does serve as a prototype for a phone terminal that demonstrates a useful application for ISDN and broadband data services that follow ISDN.
Appendices

Appendix-A  MESS System Screen Display .............................................................. 78
Appendix-B  SBC68K System Memory Map - RAM..................................................... 79
Appendix-C  The Information Super Highway............................................................ 80
Appendix-D  Ethernet Addressing Conventions....................................................... 88
Appendix-E  Aztec C Compiler Modifications ........................................................... 89
Appendix-F  Audio Data Formats .............................................................................. 92
Appendix-G  Central Office System Hardware Addresses ......................................... 98
Appendix-H  SBC68K Hardware Addresses ............................................................... 99
Appendix-I  Quick Reference Guide for MESS System ............................................... 100
Appendix-J  Source Code Listings ............................................................................. 103
Appendix-K  Node Interface Board Schematics ......................................................... 105
Screen Display

Figure A.1
System Memory Map (RAM)

000000
Interrupt Vector Table

000400
Data Area for TUTOR Monitor

001000
Code Segment

Initialized Data Segment

Uninitialized Data Segment

User Stack

Supervisor Stack

System Heap

Message Buffer (Editor)

Receive Buffers

Transmit Buffers

Wrap-Around Mail Buffer

Note: The Heap is aligned on a 32-bit boundary by adding 0-7 to Storg.. The message buffer starts at Storg..+HEAPSIZE=00 to insure a safe margin. [See rom68.asm]

STKSZ = 2000h
Storg...
HEAPSIZE = 800h

Compiled

Dynamic

[500:20 = 640h]

[500:24 = 780h]

[500:26 = 850h]

[SBC68kTotalRAM]

Figure A.2
Appendix-C The Information Super Highway

As we have seen, ISDN technology is here and moving towards better standardization and more widespread availability. It is slowly gaining acceptance and expanding in the corporate world. Even as this process takes place, newer, higher data rate technologies are beginning to emerge. This section addresses current directions in what has become a race to build a new telecommunications infrastructure in the United States generally referred to as the "Information Super Highway." First, a brief description of the forces involved in shaping telecommunications in the United States is presented. This is followed by a discussion of the directions the major players are taking.

The two main groups that influence telecommunications in the United States are the government and the telecommunications industry. The government's role includes court rulings, state and federal regulation, and participation in Research and Development (R&D) and standardization efforts. The telecommunication industry includes equipment manufacturers, local service providers, long distance carriers, and information service providers. (Many companies cross the boundaries of these categories.)

The modern era in telecommunications began on 1 January 1984 with the Modified Final Judgment (MFJ), a Consent Decree between the Department of Justice and the Bell System (AT&T) that specified the breakup of its monopoly over telephone service. The MFJ divided AT&T into seven Regional Bell Holding Companies (RHCs) and a greatly reduced AT&T. Prohibitions on business areas were placed on the RHCs (which maintained virtual monopoly control over local telephone service in their areas), and a Consent Decree Court was setup to monitor the agreement and review requests for waivers from RHCs to enter new lines of business. The United States was logically divided up into Local Access and Transport Areas (LATA) for determining range of service limitations. Large states typically consist of a number of LATA; however, smaller states may be covered by one or two LATA. Three ranges of services were then defined: local, non-toll calls; intraLATA, short-haul toll calls, and interLATA, long-haul toll calls. Service providers classified as Interexchange carriers (IXCs) may provide interLATA (and in some cases intraLATA) services throughout the country. IXCs include the big three that have their own nationwide networks: AT&T, MCI, and Sprint. There are also a number of other regional long distance companies that lease capability where they do not have their own. These include LDDS Communications, US Long Distance, and RCI Communications (a subsidiary of Rochester Telephone). The seven RHCs are Local Exchange Carriers (LECs). The RHCs and their region of influence are shown below.
Table A.1 Regional Bell Holding Companies Area of Influence

<table>
<thead>
<tr>
<th>Name</th>
<th>Region of Influence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ameritech</td>
<td>17.0M Access Lines in IL,IA,MI,OH,WI</td>
</tr>
<tr>
<td>Bell Atlantic</td>
<td>18.5M Access Lines in PE,NJ,DE,MD,VA,WV,WDC</td>
</tr>
<tr>
<td>Bellsouth</td>
<td>19.2M Access Lines in AL,FL,GA,KY,LO,MS,NC,SC,TC</td>
</tr>
<tr>
<td>NYNEX</td>
<td>16.0M Access Lines in New York and New England</td>
</tr>
<tr>
<td>Pacific Telesis</td>
<td>14.5M Access Lines in CA/NV</td>
</tr>
<tr>
<td>Southwestern Bell</td>
<td>13.2M Access Lines in AK,KA,MO,OK,TC</td>
</tr>
<tr>
<td>US West</td>
<td>13.7M Access Lines in AZ,CO,ID,IO,MN,MT,NE,NM,ND,SD,UT,WA,WA,WN</td>
</tr>
</tbody>
</table>

In addition to the RHCs, a number of Independents operate LECs throughout the country. The largest of these and their areas of operation are shown below.

Table A.2 Independents Telco's Service Regions

<table>
<thead>
<tr>
<th>Name</th>
<th>Service Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>GTE</td>
<td>21.4M Access Lines in 40 states, Canada, South America</td>
</tr>
<tr>
<td>Southern NE Telephone</td>
<td>1.9M Access Lines in Connecticut</td>
</tr>
<tr>
<td>Rochester Telephone</td>
<td>895k Access Lines in 15 states (primarily NY, PE, MI, IA)</td>
</tr>
<tr>
<td>Century Telephone Enterprises</td>
<td>426k Access Lines in 15 states in the South/Midwest</td>
</tr>
<tr>
<td>Telephone and Data Systems</td>
<td>350k Access Lines in 29 states (rural telcos)</td>
</tr>
</tbody>
</table>

All LECs are required to provide basic service elements, equal in "type, quality, and price" to all long distance carriers. [This requirement has hindered the progress of ISDN by restricting LECs from working with an IXC to develop ISDN solutions.] The RHCs were also banned from providing IXC services, providing Information Services, and manufacturing telecommunications equipment. Since 1984, a number of waivers of the MFJ have been granted and the prohibition on providing Information Services was lifted entirely on 31 October 1991. The basic monopoly nature of LECs, and the RHCs prohibition on providing long distance service have remained. This will not be the case for long, however, as LECs and government elements have begun to realize that competition will be beneficial and is inevitable.

In addition to the Consent Decree Court which with the US Department of Justice oversees the MFJ and the operation of the RHCs, the telecommunications industry is subject to regulation at both the state and federal level. State regulation is administered by Public Service
Commissions and governs intrastate communications only. Federal regulation, from the Federal Communications Commission (FCC), governs both intrastate and interstate communications. FCC regulations typically preempt state regulations.

The movement towards competition in local services has already begun. The Department of Justice has concluded that the local exchange is no longer a "natural monopoly" as new services that fall between Community Access Television (CATV) and telephone service are planned and alternate technologies such as cellular telephony continues to spread. A recent court ruling involving Bell Atlantic found that the prohibition on cross-ownership of Telcos/CATV to be unconstitutional. An FCC ruling that allows LECs to provide video services to the home, followed closely by the merger of Bell Atlantic with a large CATV firm (Tele-Communications, Inc.), caused the broadband revolution to take off this past year. The first of the LECs to address the issue of competition in local service was Rochester Telephone. In February of 1993, it filed a proposal with the New York State Public Service Commission (NY-PSC) that would allow competition in the Rochester area. The way that the plan would work is that the LEC portion of Rochester Telephone would be divided into two companies referred to as R-Com and R-Net. R-Net would own the network infrastructure and be responsible for its operation. It would sell basic network services to competing companies at set prices in much the same way as LECs sell basic services to IXCs today. R-Com would be one of the retail telecommunications firms providing competitive local services. Other Rochester Telephone assets such as Rochester Telephone Mobile (which provides cellular service) and RCI Communications (which provides long distance service) would be combined under a new holding company. The NY-PSC has yet to rule on the proposal. Updated information was provided by Rochester Telephone in September of 1993.

The most active RHC in planning for competition is Ameritech. It has proposed an experiment in Illinois beginning in 1995 that would allow competition. Ameritech would open its network to competitors in exchange for permission to provide interLATA service in the state. It has also filed a plan it calls "Customer First" with the FCC that would eventually bring about competition in its five state region.

The RHCs as a group are pushing Congress (the Senate Communications Subcommittee) to remove the ban on their entering the IXC business. The RHCs believe that the ban is outdated in today's highly competitive marketplace, arguing that cellular service already provides competition in the local exchange. IXCs (led by AT&T and MCI) oppose allowing the RHCs into long distance on an unrestricted basis. MCI argues that LECs have too tight control over local markets and would have an unfair advantage in long distance competition. (At the same time MCI is planning to enter the LEC business itself, see below.) AT&T disputes the RHCs cellular argument contending that 99% of cellular calls are routed
through the LEC. Their assertion is that the ban should remain until the local exchange is at least 75% competitive.

In the mid-1980's, the LEC community had the grand idea that they would run fiber-optic cable to every home. This was to provide the bandwidth needed for ISDN and video services. In 1989, when the cost of such a project became apparent (about $3000 per subscriber), these plans suddenly died. Since that time, LECs and IXCs have been developing ATM and SONET to provide high-speed voice/video/data services for businesses. Fiber-optic solutions have already been used for backbone networks by many carriers. (ATM and SONET are both standards for high-speed data transmission. They are discussed in the section on B-ISDN.)

All of the big three IXCs are now developing SONET-based fiber-optic networks for high data rate business needs and for their own use. AT&T initiated interLATA SONET service (called Accunet T155 for its 155.250 Mbps data rate) between 200 cities in December of 1993. Sprint plans to have an all SONET network by the end of 1994 and introduce interLATA SONET services thereafter. MCI's network is 50% SONET capable but they do not currently provide any interLATA SONET services. MCI has announced plans to build an $18 billion dollar SONET network, called networkMCI, that will serve both domestic and international customers. They also have plans for a $2 billion dollar fiber-optic network, called metroMCI, that will enable them to provide local service in 20 of the largest US markets. MCI owns the necessary right-of-ways and conduit in these areas as the result of a deal with Western Union in 1990.

As the broadband revolution gets underway, a number of solutions for making the local connection to homes are available to telecommunications companies for consideration. The first, and most popular, is fiber-optics and combinations of fiber-optics with other conventional techniques. Coaxial cable (co-ax) networks are a second option. The Asynchronous Digital Subscriber Line (ADSL), a copper-pair based solution, promises to make major advances that will put it in contention for some applications. Lastly, wireless solutions are being looked at to overcome some problems.

The original idea of fiber-to-the-home (FTTH) has returned as an option. A less expensive solution is fiber-to-the-curb (FTTC). This idea uses fiber-optics to a point central to a group of customers then provides the individual drops to homes using conventional co-ax and twisted-pair wiring. A third concept is a hybrid fiber-optic/co-ax network. The biggest problem to overcome with fiber-optic solutions is the delivery of power for emergency operation. Since fiber-optic cable cannot carry power, most companies run separate lines for power distribution. In a FTTC system, separate power is run to the central location and then delivered to the home on co-ax or twisted-pair along with the data.

Some cable TV companies plan to try to use existing co-ax networks (although they must be modified for bi-directional transmission) to provide telephone and other services. Three
companies that have developed systems that support telephony over existing CATV networks are Scientific Atlanta, ADC Telecommunications, and Antec Corporation. Downstream telephony is carried in the high part of the spectrum above the TV channels, and the upstream telephony is carried in the reserved 5-30 MHz return channel. These systems can support up to 480 non-blocking voice channels.

The original ADSL systems, mentioned in the section on the Evolution of ISDN, were designed to provide unidirectional 1.5 Mbps. This is too little bandwidth for high quality video. It uses a Quadrature Amplitude Modulation (QAM) scheme similar to that used by modern modems. The latest proposed ADSL standard is 6 Mbps with the following specifications. It will operate over 12000 feet of 24 gauge twisted-pair or lesser distances for 26 gauge or mixed gauge wiring. It will provide four unidirectional A channels each 1.5 Mbps and a bi-directional channel up to 384 kbps. (Either an ISDN H0 channel at 384 kbps or an ISDN BRA line at 144 kbps could be selected.) Analog phone service on the twisted-pair will not be disturbed. The capability to transmit over 6 Mbps of data on a copper twisted-pair has been demonstrated using a technique called Discrete Multitone Technology (DMT). The basic principal of DMT is to divide the available bandwidth into 256 subchannels. The characteristics of each channel are measured and used to assign a certain quantity of bits (0-11) to the channel. On good lines, DMT's performance has exceeded 6 Mbps. On poor quality lines, DMT's performance is similar to that of QAM. For the ADSL, 6 Mbps standard, the bandwidth is allocated as follows: 0-10 kHz is reserved for the analog telephone signal, 10-50 kHz is used for the 384 kbps upstream signal, and 50 kHz-1.1 MHz is used for the four 1.5 Mbps and one 384 kbps downstream signals.

There are two wireless techniques currently being studied for delivery of services to the home. (Mobile solutions such as Cellular, Digital Cellular, and Personal Communications Systems are not included.) The first is a low data rate NYNEX system designed to replace the copper drop wires from telephone poles to homes. This system uses the 1850 MHz-1990 MHz band. Using a 32 kbps ADPCM coding scheme, ten full-duplex voice channels can be provided per 1 MHz of bandwidth. In a test being conducted with the cooperation of 160 families and small businesses in Brooklyn, New York, the quality has been rated subjectively as equal to that of a normal copper wire system. The primary purpose of this system is to reduce installation costs, but it could serve along with a co-ax drop for video/power in a FTTC solution. The second wireless solution under study is a broadband, 28 GHz band system. This type of system offers the possibility of broadband without the high cost of installing fiber-optic cable. The concept is called Local Multipoint Distribution Service. At present its primary user/advocate is CellularVision. They use a 1 GHz bandwidth signal within 2-3 mile cells to broadcast 50 television channels. A five inch square receive antenna is all that is required at the destination. This type of system holds promise for bi-directional, broadband service in the future.
(CellularVision claims that they could support 400,000 telephone lines per cell); however, it is early in the study phase at present.

A number of LECs have begun to deploy fiber-optics in the local exchange. Announced plans include the following: Ameritech plans to have 2.5 million fiber-optic lines deployed by 1995. They plan to use FTTC and some hybrid fiber-optic/co-ax solutions for businesses and neighborhoods. ADSL will be used for isolated subscribers. NYNEX plans to deploy FTTC for some 400,000 lines by line 1995. BellSouth has plans to deploy 150,000 fiber-optic lines. Pacific Bell, a subsidiary of Pacific Telesis, has announced that it plans to begin providing basic service over a hybrid fiber-optic/co-ax platform. Construction will begin in 1994, reach 1.5 million homes by 1996, and cover California sometime after the year 2000. Many LECs are conducting real-world trials of broadband systems. US West and Southern New England Telephone are launching hybrid fiber-optic/co-ax trials using neighborhood centralized nodes. Bell Atlantic is currently conducting ADSL trials in Virginia that began in May of 1993. The initial test was conducted using 1.5 Mbps ADSL (with a 16 kbps back channel) over up to 18000 feet of unconditioned copper-pair to 70 employees' homes. Future tests will use 3 Mbps ADSL at distances up to 12000 feet and 6 Mbps ADSL at distances up to 6000 feet. Both employees and later customers will be involved. Rochester Telephone plans to conduct a video-on-demand trial for 6 months in Rochester beginning in the April of 1994. (FCC approval had delayed the trial, but was granted on 30 March.) The trial participants will include 75-90 high density residential area customers served by a fiber-optic/co-ax network in an FTTC arrangement. (Fiber-optic cable will run to a centralized optical node that will then be connected to individual homes by co-ax.) An additional 10-25 single family homes will be connected by 6 Mbps ADSL. The video server and programming will be provided by Rochester Telephone's partner in the trial, USA Video. Bell Atlantic is the first LEC to have announced specific plans to provide video services commercially. They plan to deliver basic cable and video-on-demand to 1.2 million homes VA, MD, NJ, PE by 1995 and full interactive services to 9 million homes by 2000.

For all of the testing and planning for the "Information Super Highway" that companies are doing, two key factors that will shape its future are out of their control. First is what services the public will want and how much they will be willing to pay for them. Second is how the federal government chooses to shape the expanding telecommunications industry.

The White House has chosen to take a leading role in the creation of the "Information Super Highway." The document, "The National Information Infrastructure: Agenda for Action," released on 15 September 1993 officially began the Administration's initiative to shape the future of telecommunications in the United States. Since that time, Vice President Al Gore has provided more specifics on the policies the White House would like to see implemented.
In a 21 December 1993 speech to the National Press Club, the Vice President outlined the competitive "information marketplace" of the future. This "information marketplace" would be made up of four components: the owners of the information highways; the makers of information appliances like televisions, telephones, computers, and future products that combine all three; the information providers including broadcasters, digital libraries, and information service providers; and information customers whose demands for privacy, affordability, and choice need to be met. He sees a future (in 10-15 years) of free and open competition where the lines between cable television, telephones, and cellular telephones will have disappeared. While being careful to say that government management of the transitional period is critical, he announced that "the Administration will support removal, over time, under appropriate conditions, of judicial and legislative restrictions on all types of telecommunications companies: cable, telephone, utilities, television, and satellite." The principles that White House sponsored legislation will be based on are encouraging private investment, promoting and protecting competition, providing open access to the network, and ensuring that everyone can benefit no matter their economic status.

In an 11 January 1994 speech to the Academy of Television Arts and Sciences, Vice President Gore outlined the Administration's proposals for creating the National Information Infrastructure (NII). He also pledged support for the current efforts in Congress to modify some of the provisions of the MFJ and give the FCC the power to allow RHCs into long distance when certain conditions are met. The NII White Paper released at the same time spells out the proposals in detail. The White Paper cites the goal that "by the year 2000, all of the classrooms, libraries, hospitals, and clinics in the United States will be connected to the NII." One proposal is for legislation that will establish a federal standard that permits entry to local telephone markets (called the "on and off ramps' of the NII") by all competitors. Regulations insuring that LECs provide equal service to all competitors would be enacted. The FCC would be given the authority to wave certain regulations for small LECs and LECs serving rural areas. The proposal also has provisions to help insure that universal service is available to all people. The limitations on cross-ownership of different types of telecommunications services would be relaxed under the proposal. Companies would be required to maintain separate affiliates for each type of service and follow certain rules to insure that one service is not used to subsidize another. In his speech, the Vice President pledged to work with Congress to reconcile the Administration's proposal with the three pending Congressional bills and quickly pass this legislation.

The competition proposals from Rochester Telephone and Ameritech, the moves by MCI to bypass LECs, and the White House push to create a National Information Infrastructure are setting the stage for rapid changes in the telecommunications industry. The current technologies being developed for broadband service by many companies go well beyond what is needed for
ISDN. It now appears that the slow development of the ISDN Information Highway may be swept along in the race to develop a broadband Information Super Highway.

Appendix-D Ethernet Addressing Conventions

Ethernet frames (or packets) are defined as part of the ISO 8802-3 standard. The Ethernet Controllers used in this project accept partial Ethernet packets and add the remaining fields. They can be setup to respond to all Ethernet packets (promiscuous mode), to a certain group of Ethernet addresses, or only to packets that match their assigned Ethernet address. Ethernet addresses are defined as 6-byte fields. A source and destination address are included in each Ethernet packet. In the ISO 8802-3 standard certain Ethernet Addresses are designated for certain uses. Although not necessary for this project, these rules were followed in assigning address used by the system. The selection of Ethernet Address as governed by the ISO 8802-3 standard and their use in the MESS system is described in this section.

The two most significant bits of the 48-bit Ethernet Address field are reserved under the ISO 8802-3 standard. \( b_{47} \) is used as a group address indicator. If set, the address is treated as a group address. If clear, the address is treated as an individual address. \( b_{46} \) is used as a global/local indicator. If set, the address is assumed to be a locally administered address. If clear, the address is assumed to be a globally administered address. In the MESS system, the nodes are assigned addresses based on the UserID given when a person begins using the system. The first byte of a node address is always 40h (\( b_{46} \) is set to indicate that it is a locally administered address) and the remaining bytes, two through six, are the five byte UserID. A node will respond only to packets addressed to this Ethernet Address. The Central Office System is setup in Promiscuous Mode. It accepts all incoming packets from the Ethernet. The Central Office System will retain and process only packets whose address contains the expected first byte. Bytes two through six are ignored for packet validation and are used only if the incoming packet is a message designated to another user. In this case, the destination UserID is stored in bytes two through six of the destination address. There are two valid first address bytes for the Central Office System: C0h and CFh. (Note that \( b_{47} \) and \( b_{46} \) are both set to indicate that a locally administered, group address is being used.) Each node is required (by the Node/Central Office Protocol -- see Section 4.2.2) to alternate between these two valid bytes.
The cross-compiler used to produce code for the SBC68K systems during the course of this project was the Aztec C68k/ROM by Manx Software Systems, Inc. This compiler is primarily intended to produce code that will be run from Read Only Memory (ROM). The startup routine incorporated into compiled programs was modified to correct several problems that the assumption of ROM caused. In particular, the initialization of stack and heap pointers was changed and the unnecessary copying of initialized data segment from ROM into RAM was removed. The modifications to the startup routine, which is contained in the file rom68.a68 are shown below.

; Copyright (C) 1986 by Manx Software Systems
; Start-up routine for 68k rom based development.
; These masks enable and disable interrupts when running in supervisor mode. The $2000 part of the mask indicates supervisor mode.

; Modified for RAM system — *aam07/25/91

Maskints    equ $2700 ; mask to disable interrupts
Enableints  equ $2000 ; mask to enable interrupts

; DATA SEGMENT

dseg
    public _mbot,_mtop,_mcur ;*jd 15 Dec 86
    public _stkbase ;*jd 8/20/87
    public _errno

HEAPSIZE     equ $800 ; size of heap space ;*jd 15 Dec 86
__mbot ds.l 1 ; ptr to bottom of heap ;*jd 15 Dec 86
__mtop ds.l 1 ; ptr to top of heap ;*jd 15 Dec 86
__mcur ds.l 1 ; ptr to top of allocated heap space ;*jd 15 Dec 86
__stkbase ds.l 1 ; ptr to bottom of stack area ;*jd 8/20/87
__errno ds.w 1 ; place to store error numbers

    public _H1_org,_H1_end ;*jd 27 Oct 86
    public _H2_org,_H2_end ;*jd 27 Oct 86
    public _H0_org,_H0_end ;*jd 27 Oct 86

; CODE SEGMENT

; When the 68k processor is reset or powered up, it fetches the values at offsets 0 and 4, which contain the address of the first instruction to execute and the initial stack pointer value respectively.
; The space up to $400 is reserved for the interrupt vector. This vector is not initialized!

cseg

    public HEAPSIZE ;*aam 7 jan 94
Program is to be loaded into RAM on an SBC68K.
therefore, the entry point should be the
address specified in the In68 command.

begin:
move.w #MaskInts,sr ; mask interrupts (supervisor mode)

; Assume program uses small code &/or small data
; and that it uses A5 to access jmptabs and data
; If program uses large code and large data,
; and you want to use A5 for other purposes (eg, register variable),
; delete the next line of code
; If it uses a different address register, change the next line
move.l #_H1_org+32766,a5 ;"jd 17 Nov 86

; Calculate length of initialized data segment
move.l #_H1_end,d0 ;"jd 27 Oct 86
sub.l #_H1_org,d0 ;"jd 27 Oct 86
beq .1end ; skip if no initialized data
lsl.l #1,d0 ; divide length by 2 and copy words

; Get addresses of initialized data segment and destination
move.l #_H0_end,a0 ; data is at end of code segment
move.l #_H1_org,a1 ; data segment destination address

; Copy initialized data segment from _Cend_ to _Dorg_
 .1
move.w (a0)+,(a1)+
 dbra d0,1
 .1end

; Calculate length of BSS segment and initialize address register
move.l #_H2_end,d0 ;"jd 27 Oct 86
sub.l #_H2_org,d0 ;"jd 27 Oct 86
beq .2end ; skip if no BSS segment
lsl.l #1,d0 ; divide length by 2 and clear words
move.l #_H2_org,a0 ; BSS base address

; Initialize BSS segment to 0
 .2
clr.w (a0)+
 dbra d0,2
 .2end

* aam 7 Jan 94

Replaced addition of 3 with 7 to insure that aligning the Heap
does not allow it to interfere with the stack. The Heap may now
exceed _Storg_+HEAPSIZE by 0-7 and must be taken into when accessing
the "free" memory at the end of RAM. The results of the Heap varying
by -4 to +3 during alignment were not clear, but seemed to be
a problem.

*aam 21 June 91

Program is to be loaded into RAM on an SBC68K.
 therefore, the entry point should be the
address specified in the In68 command.

dc.l _Storg_ ; initial stack pointer value
dc.l .begin ; address of first instruction
ds.b $3f8 ; reserve space for interrupt vectors

begin:
move.w #MaskInts,sr ; mask interrupts (supervisor mode)
; Initialize heap pointers
move.l #_Storg_d0, d0 ; align heap on 32-bit boundary *jd 15 Dec 86
add.l #3, d0
add.l #7, d0 ; *aam 7 Jan 94 - buffer to separate Heap/Stack
and.l #&-8, d0 ; *jd 15 Dec 86
move.l d0, __mbot ; *jd 15 Dec 86
move.l d0, __mcur ; *jd 15 Dec 86
add.l #HEAPSIZE, d0 ; *jd 15 Dec 86
move.l d0, __ntop ; *jd 15 Dec 86

; Initialize stack pointers
move.l #_Storg_d0, d0
sub.l #STKSIZ, d0 ; *jd 10/27/86
move.l d0, __stkbase ; bottom of stack
move.l d0, a0 ; for stack overflow checking *jd 8/14/87
move.l #$4d414e58, a0 ; "MANX" at base of stack area
add.l #STKSIZ/2, a0 ; *aam 07/25/91 initialize user stack ptr
move.l a0, usp ; *aam 07/25/91 initialize system stack ptr
move.l a0, sp

; Enable interrupts and branch to main routine
move.w #EnableInts, sr ; enable interrupts (supervisor mode)
jmp _main ; *jd 10/28/86
end

Figure A.3 Modified rom68.a68 Startup Routine
Appendix-F Audio Data Formats

In this project, two different sources for audio input/output were used. One was the ISDN Digital Subscriber Controller (DSC) interfaced to the SBC68K. The second was a PC-based sound card using a Microsoft Windows 3.1 compatible driver. In this section, a description of the formats used by each source, as well as a method for conversion between the two, will be presented.

The DSC can be programmed to provide raw audio data in several formats. The constants are an 8 kHz sampling rate and 8-bits per sample. One programmable selection is between using u-Law coding or A-Law coding to convert the data from its internal 12-bit sample size to an 8-bit sample size for output. A second option is bit-reversal. This determines how data passed to/from the microprocessor from/to the DSC is evaluated. If bit-reversal is disabled, the highest bit of the word from the microprocessor is treated as the most significant by the DSC, the second highest as the second most significant, and so on. If bit-reversal is enabled, the word from the microprocessor is treated in the opposite manner with the highest bit being considered least significant, and so on. For this project u-Law coding (the US standard) is used and bit-reversal is disabled.

The u-Law coding used by the DSC has the following format. An eight-bit word consisting of seven data bits preceded by an independent sign bit is used. The individual bits are inverted: a zero is used to represent a high value, and a one is used to represent a low value. The u-Law encoding itself is a logarithmic one whose function and effect is shown below:
Input Resolution: $R_{in} = 11$ bits
Output Resolution: $R_{out} = 7$ bits

$u = 127$

$x \in \left(0, 2^{R_{in}} - 1\right)$

$x' = x / 2^{R_{in}}$

$y' = \frac{\ln(1 + ux')}{\ln(1 + u)}$

$y = y(2^{R_{out}})$

Figure A.4  u-Law Scheme used by DSC

![Graph of u-Law Scheme used by DSC](image)

Figure A.5  Graph of u-Law Scheme used by DSC

Audio Data under Microsoft Windows 3.1 is stored in a standard multimedia file format known as the Resource Interchange File Format (RIFF). Standard functions are provided for operating on data stored in this form. Two utility programs (Media Player and Sound Recorder) are provided that allow simple recording, editing and playback of standard audio data. A file type of .WAV is typically used for these audio data files.

A RIFF file is composed of chunks (varying length blocks) of data each identified by an ASCII tag. This format allows simple programs to work on data produced by more complex ones by using only those basic data chunks that it understands and skipping others. A chunk is stored as follows:
The TAG is a four byte ASCII value. The LENGTH is a four byte field representing the number of bytes in the DATA field. The DATA field contains an even number of bytes of information. Two special chunk types are reserved: RIFF and LIST. The RIFF chunk is used for the entire file as shown below:

```
R I F F Total Length Form Sub-Chunk-1 Sub-Chunk-2 ... Sub-Chunk-n
```

Figure A.7 Special RIFF Chunk Structure

The Form field is used to indicate the nature of the RIFF file. A LIST chunk is similar to the RIFF chunk shown above, but is used within the file and with one of a number of predefined LIST identifier tags.

A standard waveform audio file uses a RIFF Form type of WAVE, and has two sub-chunks named "fmt " and "data" that are mandatory. (Note that additional sub-chunks that provide more information about the file are allowed and would be passed over by programs that do not recognize them.) The "fmt " sub-chunk consists of 16 bytes that specify the sample size, sampling rate, etc. (See table below.) The "data" sub-chunk consists of the actual waveform data coded as specified in the "fmt " sub-chunk.

Table A.3 Components of RIFF "fmt " Structure

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wFormatTag</td>
<td>2 bytes</td>
<td>1 = PCM (Other values currently undefined)</td>
</tr>
<tr>
<td>nChannels</td>
<td>2 bytes</td>
<td>1 = Mono; 2 = Stereo</td>
</tr>
<tr>
<td>nSamplesPerSec</td>
<td>4 bytes</td>
<td>Sampling Rates of 11025/22050/44100 are standard</td>
</tr>
<tr>
<td>nAvgBytesPerSec</td>
<td>4 bytes</td>
<td>(Sampling Rate)x(Sample Size in Bytes)x(Number of Channels)</td>
</tr>
<tr>
<td>nBlockAlign</td>
<td>2 bytes</td>
<td>(Sample Size in Bytes)x(Number of Channels)</td>
</tr>
<tr>
<td>nBitsPerSample</td>
<td>2 bytes</td>
<td>1-16 (Above 8 is signed data; 8 or below is unsigned data)</td>
</tr>
</tbody>
</table>
The sound board used (during development) for this project—coupled with the Microsoft Windows 3.1 utilities—records audio in Mono using an 8-bit unsigned value and a sampling rate of 11,025 Hz. Playback of 11,025 and 22,050 Hz signals is supported.

From the descriptions above, it can be seen that two obstacles must be overcome to convert from PC-based audio to DSC compatible audio. First the u-Law coding must performed (or reversed in the opposite case). Second the difference between the 8 kHz sampling rate and the 11,025 or 22,050 Hz sampling rates must be reconciled. The procedures uLaw() and uLawInvert() shown at the end of this section are used for the first obstacle. Note that conversions between Sign/Magnitude, Two's Complement, and Unsigned data formats are also required. The Macros used for these conversions are shown at the end of this section. Also, notice that in both cases the conversion is scaled so that it is from 8-bit data to 8-bit data unlike the 12bit-to-8bit and 8bit-to-12bit conversions performed by the DSC. This is because of the limitations of the PC-based audio system. For the uLaw() procedure, the function implemented is the one shown earlier in this section. The reverse of this function can be derived as shown below. This reverse function is implemented by the uLawInvert() procedure.

\[
y' = \frac{\ln(1 + ux')}{\ln(1 + u)} \\
y' \ln(1 + u) = \ln(1 + ux') \\
\left( \ln \left( (1+u)^{y'} \right) \right) = \ln(1 + ux') \\
(1 + u)^{y'} = 1 + ux' \\
x' = \left( \left( 1 + u \right)^{y'} - 1 \right) / u
\]

Figure A.8 Inverse u-Law Equation

The conversion from one sampling rate to another is performed by the procedures fcvt() and sL_approx(). The conversion uses a straight line approximation technique. The concept is to divide up the input waveform into divisions of 1 ms. From the n samples of the input waveform during a period, the straight line approximation is used to create an n x m sample (where m is the number of samples of the output waveform in 1 ms.) virtual waveform. The m output samples are then selected from this virtual waveform. This technique works for converting either upwards (from 8 kHz to 11/22 kHz) or downwards (from 11/22 kHz to 8 kHz). The only cause for concern is the case in which a component of a frequency higher than that of the system being converted to is present in the original signal. This problem can be eliminated by using a low-
pass filter prior to the conversion so that aliasing of the high frequency signal into lower frequencies will not occur.

```
#define TWOS_TO_UNSIGNED(x) ((uchar)((x)+0x80))
#define TWOS_FR_UNSIGNED(x) (((int)(x))-0x80)
#define TWOS_TO_SIGNMAG(x) ((uchar)((((((x)<0)?(-(x)-1):x)&0x7F)|(((x)<0)?0x80:0x00)))
#define TWOS_FR_SIGNMAG(x) (((int)(((x)&0x7F)*(((x)&0x80)?(-1):(+1))+(((x)&0x80)?(-1):0))))

Figure A.9 Macros for Format Conversion
```

```
int uLaw(int X)
{
    int y;
    double sign;
    double u;
    double x;
    if (X>=128||X<-128) {
        printf("Problem with conversion\n");
        return (0);
    }
    u = 127;
    sign = ((X<0)?(-1.0):(1.0));
    x = (sign*(((double)x)/128));
    y = (int)((sign*(log(1+(u*x))/log(1+u)))*128);
    return(y);
}

Figure A.10 uLaw() Routine
```

```
int uLawinvert(char Y)
{
    int x;
    double sign;
    double u;
    double y;
    u = 127;
    sign = ((Y<0)?(-1.0):(1.0));
    y = (sign*(((double)y)/128));
    x = (int)((sign*((pow((1+u),y)-1)/u))*128);
    return(x);
}

Figure A.11 uLawInvert() Routine
```
```
#define MAXSAMPLESPERMS 44
int A[MAXSAMPLESPERMS];
int B[MAXSAMPLESPERMS];
int ASamplesPerMS;
int BSamplesPerMS;

Figure A.12 Globals used by fcvt() and sl_approx() Routines

int fcvt(char *Input, char *Output, int Length, int I, int O)
{
    int i, j, k;
    ASamplesPerMS = I;
    BSamplesPerMS = O;
    j = 0;
    A[0] = 0;
    for (k = 0; k < Length; ++k) {
        A[(k % ASamplesPerMS) + 1] = Input[k];
        if ((k % ASamplesPerMS) == ASamplesPerMS - 1) {
            for (i = 0; i < BSamplesPerMS; ++i) {
                B[i] = sl_approx(i * ASamplesPerMS);
                Output[j++] = (char)B[i];
            }
            A[0] = A[ASamplesPerMS];
        }
    }
    return (j);
}

Figure A.13 fcvt() Routine

int sl_approx(int x)
{
    float y1, y2, slope;
    int y;
    y1 = A[(x / BSamplesPerMS)];
    y2 = A[(x / BSamplesPerMS) + 1];
    slope = (y1 - y2) / ((float)BSamplesPerMS);
    y = (int)((slope * (x % BSamplesPerMS)) + y1);
    return (y);
}

Figure A.14 sl_approx() Routine
```
Appendix-G  Central Office System Hardware Addresses

Table A.4 Programmable Interrupt Controllers

<table>
<thead>
<tr>
<th>Master PIC</th>
<th>Slave PIC</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>20h</td>
<td>A0h</td>
<td>Mask Register</td>
</tr>
<tr>
<td>21h</td>
<td>A1h</td>
<td>End of Interrupt Register</td>
</tr>
</tbody>
</table>

Table A.5 Direct Memory Access Controllers

<table>
<thead>
<tr>
<th>Low DMA Controller</th>
<th>High DMA Controller</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>00h</td>
<td>C0h</td>
<td>Base Address Register</td>
</tr>
<tr>
<td>08h</td>
<td>D0h</td>
<td>Command/Status Register</td>
</tr>
<tr>
<td>09h</td>
<td>D2h</td>
<td>Request Register</td>
</tr>
<tr>
<td>0Ah</td>
<td>D4h</td>
<td>Single Channel Mask Register</td>
</tr>
<tr>
<td>0Bh</td>
<td>D6h</td>
<td>Mode Register</td>
</tr>
<tr>
<td>0Eh</td>
<td>DCh</td>
<td>Mask Clear Register</td>
</tr>
<tr>
<td>0Fh</td>
<td>DEh</td>
<td>Mask Set Register</td>
</tr>
</tbody>
</table>

Table A.6 Allied Telesis 1500 (Am79c960)*

<table>
<thead>
<tr>
<th>Address</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>320h</td>
<td>Address PROM</td>
</tr>
<tr>
<td>330h</td>
<td>Register Address Port</td>
</tr>
<tr>
<td>332h</td>
<td>Register Data Port</td>
</tr>
<tr>
<td>334h</td>
<td>Reset Register</td>
</tr>
<tr>
<td>336h</td>
<td>ISA-bus Control Data Port</td>
</tr>
<tr>
<td>338h</td>
<td>Vendor Specific Word</td>
</tr>
</tbody>
</table>

* The base address of the AT1500 is programmable to 300h/320h/340h/360h.
## Table A.7  DSC Register

<table>
<thead>
<tr>
<th>Address</th>
<th>Read Access</th>
<th>Write Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>7FFFF1</td>
<td>Interrupt Register</td>
<td>Command Register</td>
</tr>
<tr>
<td>7FFFF3</td>
<td>Data Register</td>
<td>Data Register</td>
</tr>
<tr>
<td>7FFFF5</td>
<td>D-Channel Status #1</td>
<td>n/a</td>
</tr>
<tr>
<td>7FFFF7</td>
<td>D-Channel Error</td>
<td>n/a</td>
</tr>
<tr>
<td>7FFFF9</td>
<td>D-Channel Receive</td>
<td>D-Channel Transmit</td>
</tr>
<tr>
<td>7FFFFB</td>
<td>B-Channel #1</td>
<td>B-Channel #1</td>
</tr>
<tr>
<td>7FFFFD</td>
<td>B-Channel #2</td>
<td>B-Channel #2</td>
</tr>
<tr>
<td>7FFFFF</td>
<td>D-Channel Status #2</td>
<td>n/a</td>
</tr>
</tbody>
</table>

## Table A.8  LANCE Registers

<table>
<thead>
<tr>
<th>Address</th>
<th>Read Access</th>
<th>Write Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>7FFFE0</td>
<td>Data Register</td>
<td>Data Register</td>
</tr>
<tr>
<td>7FFFE8</td>
<td>Address Register</td>
<td>Address Register</td>
</tr>
</tbody>
</table>
The MESS system has three different modes that a user can be in: the main menu, text editing, and mail message processing. Each of the modes also has a help screen associated with it. All of the screens within the system consist of a title bar, a text window, and a message window. The title bar tells the user what mode he/she is in. The text window is used to display help/information and mail message text. The message window is a scroll box where all messages from the system are displayed.

The user begins by "logging in" to the system by entering a 5-character id. This id is used by others who wish to send messages to the user. The user then enters the main menu from which he can access either of the other modes. A guide to each of the three modes is given below.

Incoming messages are received and stored by the Central Office system. A user may retrieve messages addressed to him/her from the Central Office. These messages as well as ones created using the editor are stored in a local message queue. The user may read, modify, send, or forward messages on the local message queue while he/she is logged on. The local message queue is not preserved once the user logs off the node. Self-addressed messages may be sent to the Central Office for storage if desired. The Central Office will automatically broadcast a message to the addressee each time it receives an incoming message. When a user first logs on, the node requests a count of messages pending from the Central Office. It then maintain this count by monitoring messages from the Central Office. In this manner, the count of messages pending for a user is kept current.

**Main Menu**

The MESS Project main menu provides the user with the a number of options. Each is selected by entering a keystroke command. The user may display the number of messages in the local message queue and held by the Central Office. He/She may also retrieve messages held by the Central Office: two commands are provided one which gets and removes a message from the Central Office, and one which simply copies a message held by the Central Office. Messages in the local message queue may then be read and/or modified using the mail message system. New messages may be created by invoking the editor. The main menu commands are listed below.
(D)isplay pending messages
(G)et messages held by the Central Office
(L)ook at messages held by the Central Office
(R)ead pending messages
(E)dit a new message
(V)ersion of MESS System
(H)elp for MESS System
(Q)uit MESS System

Editor

The MESS Editor is designed for the composition and modification of text messages. The editor provides normal cursor movement using the arrow keys; however, only overtype mode is supported. Both backspace and delete are supported. There is primitive support for word wrapping. The word wrapping mode is toggled using ^W. Word wrapping is useful only when entering a message. It functions only when text is being added to the bottom of a document. When the end of a line is reached and you are in the middle of a word, it is moved to the next line and typing continues uninterrupted. If the user chooses to abort the during the creation of a new message, the user is returned to the main menu and nothing is saved. If the user chooses to save the message, he/she is placed into the mail message system in order to be able to add speech to and/or send the message.

Control-A Abort, Return to Main Menu
Control-Z Save Message and Proceed to Message System
Control-R Refresh the Screen
Control-W Toggle Word-Wrapping Mode
Control-B Display Help Screen

Mail Message System

The MESS Mail Message System is designed for the review and modification of voice/text messages. It is entered either following the creation of a new message using the editor, or to read existing messages. The user has the option to quit--preserving the message, to discard the message, to send/forward the message, or to return to the editor to make modifications to the message. The system also provides an interface for manipulating voice messages. The user may attach a voice message(s), or playback existing voice message(s).
Voice messages are marked by highlighted sections of text. To attach a voice message, the user should place the cursor at the desired point and enter the attach command. To playback a voice message, the user should move the cursor to a highlighted section of text and enter the playback command. Voice messages may be removed one at a time by positioning the cursor on the desired highlighted text and entering the remove command. Entering the editor will clear all old voice messages. There is a limit of one voice message per line, and a maximum of four per message. The last voice message recorded or played is saved in a local sound buffer. It can be attached to multiple points in a document, if desired.

(Q)uit to main menu and leave message in local message queue
(D)iscard message and return to main menu
(S)end/Forward message
(E)dit the message
(A)ttach a new voice message
   (Z) re-attach current voice message
(P)layback a voice message
   (X) re-play current voice message
(R)emove a voice message

Control-R       Refresh the Screen
Control-B       Display Help Screen
Appendix-J  Source Code Listings

[Note: Due to the size of the Source Code Listings they are contained in a separate volume.]

Node Source Code

<table>
<thead>
<tr>
<th>File</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>mess.c</td>
<td>3</td>
</tr>
<tr>
<td>audio.c</td>
<td>4</td>
</tr>
<tr>
<td>duart.c</td>
<td>7</td>
</tr>
<tr>
<td>edit.c</td>
<td>11</td>
</tr>
<tr>
<td>globals.c</td>
<td>14</td>
</tr>
<tr>
<td>lance.c</td>
<td>27</td>
</tr>
<tr>
<td>mail.c</td>
<td>30</td>
</tr>
<tr>
<td>menu.c</td>
<td>37</td>
</tr>
<tr>
<td>packet.c</td>
<td>45</td>
</tr>
<tr>
<td>que.c</td>
<td>49</td>
</tr>
<tr>
<td>term.c</td>
<td>56</td>
</tr>
<tr>
<td>userlvl6.c</td>
<td>58</td>
</tr>
<tr>
<td>consts.h</td>
<td>64</td>
</tr>
<tr>
<td>globals.h</td>
<td>67</td>
</tr>
<tr>
<td>macros.h</td>
<td>72</td>
</tr>
<tr>
<td>procs.h</td>
<td>74</td>
</tr>
<tr>
<td>types.h</td>
<td>76</td>
</tr>
<tr>
<td>mon.a68</td>
<td>78</td>
</tr>
<tr>
<td>makefile</td>
<td>79</td>
</tr>
</tbody>
</table>

Central Office System Source Code

<table>
<thead>
<tr>
<th>File</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>co.c</td>
<td>82</td>
</tr>
<tr>
<td>dsp.c</td>
<td>83</td>
</tr>
<tr>
<td>globals.c</td>
<td>86</td>
</tr>
<tr>
<td>hw.c</td>
<td>98</td>
</tr>
<tr>
<td>lance.c</td>
<td>100</td>
</tr>
<tr>
<td>lpc.c</td>
<td>103</td>
</tr>
<tr>
<td>packet.c</td>
<td>110</td>
</tr>
<tr>
<td>riff.c</td>
<td>120</td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>const.c</td>
<td>134</td>
</tr>
</tbody>
</table>
Appendix-K  Node Interface Board Schematics

Node Interface Board Schematics .................................................................................................................................................. 106
High Level Schematic .................................................................................................................................................................. 106
Chip Select and DSC Interface ..................................................................................................................................................... 107
  DSC Interface Circuit (Nodes 1/2) ........................................................................................................................................... 107
  Components (Nodes 1/2) ......................................................................................................................................................... 108
mc68000 bus R/W Timing Diagram ........................................................................................................................................ 109
Combinatorial Logic (Node 1) ....................................................................................................................................................... 110
Chip Select/Dtack Logic (Node 1) ............................................................................................................................................... 111
Combinatorial Logic (Node 2) ....................................................................................................................................................... 112
Chip Select/Dtack Logic (Node 2) ............................................................................................................................................... 113
LANCE Interface .......................................................................................................................................................................... 114
  LANCE Interface Circuit .......................................................................................................................................................... 114
  Components ................................................................................................................................................................................ 115
LANCE Interface Considerations ................................................................................................................................................ 116
Serial Interface Adapter Circuit .................................................................................................................................................... 117
Transceiver Circuit ......................................................................................................................................................................... 118
Microphone Circuit ........................................................................................................................................................................ 119
Telephone Handset Interface Circuit ........................................................................................................................................... 120
Op-Amp Hybrid ............................................................................................................................................................................. 121
  Op-Amp Hybrid Circuit ........................................................................................................................................................... 121
  Components ................................................................................................................................................................................ 122
DSC Interface Circuit
Components

**Components**

---

---
0800 Read/Write Timing
<table>
<thead>
<tr>
<th>State</th>
<th>Input</th>
<th>Output</th>
<th>Next State</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>A</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ \overline{RD} = \overline{ABC} \]
\[ \overline{WR} = \overline{ABC} \]
\[ A = \overline{ABC} \]
\[ B = \overline{ABC} + \overline{ABC} \]
\[ C = \overline{ABC} + \overline{ABC} \]

![Combinatorial Logic Diagram]

![Components Diagram]
STACK LOGIC (WITH WAIT STATES)

DATA DIRECTION LOGIC

CHIP SELECT LOGIC

Chip select / Stack Logic
\[
\begin{array}{c|c|c|c|c|c|c|c}
\text{State} & \text{Input} & \text{Output} & \text{Next State} \\
\hline
A & B & C & x_1/11 & x_2/11 & x_3/11 & x_4/11 & x_5/11 \\
\hline
0 & 0 & 0 & X & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & X & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & X & 1 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & X & 1 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & X & 1 & 1 & 0 & 0 & 0 \\
\end{array}
\]

\[
\overline{RD} := \overline{ABC} \\
\overline{WR} := \overline{ABC} \overline{RW} \overline{RA} \\
A := \overline{ABC} \\
B := \overline{ABC} \overline{RW} \overline{RA} + \overline{ABC} \\
C := \overline{ABC} \overline{RW} \overline{RA} + \overline{ABC}
\]

\[
\overline{READ} = \overline{RW} \overline{CS1} \\
\overline{WRITE} = \overline{RW} \overline{CS1}
\]

\[
\overline{STACK.CS} = (\overline{HSLK} + \overline{CS1}) - (\overline{TSLK} + \overline{CS1})
\]

Combinatorial Logic
Components

Components
Label Interface Considerations

They drive M5-A4 only when the 74194 is in bus-master mode.

Address lines M5-A4 are multiplexed with data lines D5-D0.

Address lines A23-A16 are output-only and should be buffered (using a 74137).

Contact directly to 6800 bus.

Some control signals are optional for bus-master mode.

Important consideration in 6800 - 7400 Interface.
Transceiver Circuit
MICROPHONE CIRCUIT:

Stage 1:
- Microphone Biasing

Stage 2:
- Amplifier

Stage 3:
- High Pass Filter
  - Fc = 200 Hz, Q = 3.6
- Low Pass Filter
  - Fc = 5 kHz, Q = 7.7

Components:

MA  MB  MC  MD  ME

Microphone Circuit
Audio Header/Node 1

Audio Connector/Node 2

1 - +12v
2 - Ans
3 - 6nd
4 - AREF
5 - -12v
Audio Interface Headers:

Components:

Components:

Components:

Components:

122
References


