Variable Bit Rate Voice Over ATM

Using Compression and Silence Removal

by

Mario A. Yearwood

Submitted to the Department of Electrical Engineering and Computer Science

in Partial Fulfillment of the Requirements for the Degrees of

Bachelor of Science in Electrical Science and Engineering

and Master of Engineering in Electrical Engineering and Computer Science

at the Massachusetts Institute of Technology

May 15,1997

Author: ____
Department of Electrical Engineering and Computer Science
May 15, 1997

Certified by: ____
Dr. Steven Finn
Thesis Supervisor

Accepted by: ____
Arthur C. Smith
Chairman, Department Committee on Graduate Theses

Variable Bit Rate Voice Over ATM With Compression and Silence Removal
by Mario A. Yearwood

Submitted to the
Department of Electrical Engineering and Computer Science

May 15, 1997

In Partial Fulfillment of the Requirements for the Degree of
Bachelor of Science in Electrical Science and Engineering
and Master of Engineering in Electrical Engineering and Computer Science

Abstract

Voice traffic is currently transported over ATM networks using a constant bit rate of 64 kilobits per second (kbps). This constitutes an inefficient use of network bandwidth since speech is highly compressible and also involves pauses for listening. A real-time Proof of Concept system was developed in software on Sun Ultra workstations at COMSAT Laboratories to demonstrate the effectiveness of transporting compressed voice over ATM with silence removal. The performance of the system indicates that voice compression and silence suppression offer substantial bandwidth savings without sacrificing speech quality.

Thesis Supervisor: Dr. Steven Finn
Title: Principal Research Scientist, Laboratory for Information and Decision Systems.

Acknowledgments

I would like to express my most sincere thanks to all those who assisted me in the preparation of my Master's Thesis. Special thanks are reserved for Faris Faris, my supervisor at COMSAT, for his patience and guidance, and for Simao Campos and Shan Lu of the Communications Technology Division at COMSAT for their time and expert input.

I would also like to express my deepest and sincerest gratitude to Dr. Steven Finn, my thesis supervisor at MIT, for his time and supervision, and to Roger Alexander of COMSAT, for his encouragement, words of wisdom, and example.

I thank you all.

Table of Contents

Table of Figures

1. Introduction to Problem

1.1. Asynchronous Transfer Mode (ATM) Networks

The introduction of the Broadband Integrated Services Digital Network (B-ISDN) is necessitated by the great demand for higher bandwidth by numerous new applications on communication networks, as well as the increased demand for communication services in general. These new, high bandwidth, applications include full motion video and large image file transfer.

ATM is the technology that will be used to implement B-ISDN because it enables a single network to seamlessly integrate all forms of traffic including video, voice, and data at very high transmission rates. The ATM Adaptation Layer, which exists above the ATM Layer and Physical Layer in the layered architecture of ATM, provides the flexibility to support the different service characteristics that video, voice, and data require. Whereas on a traditional 64kbps connection an image file that was 10^9 bits long would take 4 hours to transfer, ATM networks with access rates of 150Mbps and higher could transfer the same file in seconds. These rates are also suitable for video broadcast and video conferencing traffic [1].

All forms of traffic are transported over ATM in cells that are 53 bytes long. The 53 byte cells consist of a 5 byte ATM header and a 48 byte ATM payload. The fixed length of all ATM cells is one of the reasons that ATM networks can operate at very high speeds. Another reason is the advances in Very Large Scale Integration (VLSI) technology, which enable faster clock rates in switches and other ATM hardware, as well as advances in fiber-optic technology.

In contrast to traditional networks where a specific amount of bandwidth is reserved for a connection, ATM allows bandwidth to be allocated on an on-demand basis. Different traffic types have different transmission statistics. For example, some data traffic tends to be bursty while video may be smooth or bursty depending on its method of encoding. With pro-active management and congestion control, ATM allows all the varying traffic types to be multiplexed together with good bandwidth utilization while still meeting the Quality of Service (QoS) requirements of the individual sessions and users[2].

1.2. Problem Definition

As was previously explained, ATM has the ability to combine voice, data and video traffic onto one network thereby precluding the need for separate networks for each traffic type. However, according to Jim Hartford in an article in the LAN TIMES entitled *ATM Must Make Way for Voice* [3], "Most of the ATM Forum's attention has focused on the data and video pieces." In the process, not enough attention has been paid to utilizing the full potential of ATM for transporting voice.

This full potential needs to be tapped since the ongoing explosion of data and multimedia communications has made network bandwidth a precious and sometimes expensive commodity. This is especially true, for example, on costly satellite links where bandwidth is not as abundant as in terrestrial optical fiber networks. It is therefore important to develop ways to make more efficient use of network bandwidth.

The information content in voice which is sampled at a constant rate and fixed number of bits per sample, varies as a function of time. This is because in addition to natural pauses, conversation involves both speaking and listening. In spite of this, mainly constant bit rate (CBR) solutions for transport of voice over ATM have been considered. These CBR methods of voice transport require a constant amount of network bandwidth to be reserved for the duration of a connection, because, as the name suggests, the samples are transferred at an unchanging rate regardless of whether a person is speaking or not. CBR solutions at 64kbps are being proposed from the standpoint of expediency and ease of interworking. They are a legacy of the organization of the Public Switched Telephone Network (PSTN) into a hierarchy based on 64kbps. While ATM will always need to interwork with PSTN, the efficient carriage of voice traffic should not be discouraged by the way in which older networks were organized.

These CBR methods fail to take advantage of the bandwidth savings that can be achieved by only transmitting voice samples when someone is speaking. According to the findings of Drago et al [4], each party speaks only 40%-50% of the time during an average conversation. If a technique was employed, where samples were transmitted only during speech and no samples were sent during silence, then theoretically 50-60% of the bandwidth used by CBR voice traffic could be saved. This bandwidth could be used to multiplex other traffic or to reduce network congestion.

In addition to gains that can be achieved through voice activity detection (VAD) and silence suppression, bandwidth can also be saved by utilizing voice compression. Several speech compression techniques exist which enable toll quality voice to be transmitted at rates much less than 64 kbps. Employing such compression technologies could considerably reduce the bandwidth required by voice traffic. Currently compression is being employed in the PSTN using Digital Circuit Multiplication Equipment (DCME) [5]. This presently uses 32kbps ADPCM compression and tests are being carried out on systems using 16kbps compression. However, these circuit multiplication techniques require several voice connections in order to achieve significant gains. This thesis project is primarily concerned with gains that can be achieved over a single connection.

The bandwidth savings mentioned above can be significant and, given the amount of voice traffic in today's network, should not be ignored.

1.3. Goal of Thesis

The goal of this thesis project is to design and implement a proof of concept system which demonstrates that silence suppression and compression can be applied to voice traffic over ATM to save significant bandwidth without compromising sound quality. If successful, the project may develop into a full scale product for COMSAT.

The overall objective will be to obtain as much bandwidth reduction as possible while still maintaining toll-quality voice transfer and low implementation complexity.

1.4. Thesis Outline

In Chapter 2 a functional description of the overall system and its sub-systems is provided. Then in Chapter 3 the preliminary analysis that determined the specific implementations of the Voice Activity Detection (VAD) algorithm, the compression algorithm and the AAL best suited for voice, is discussed.

Chapter 4 provides a detailed description of how the entire system was implemented in software. Chapter 5 describes various procedures that were carried out in order to tune the system and configure it optimally. Results are described for various VAD parameter settings, silence regeneration algorithms, and lost cell replacement algorithms. Chapter 6 contains an evaluation of the speech quality and bandwidth savings obtained from the system. The conclusion and suggestions for future research are located in chapter 7.

2. System Description

The block diagram in Figure 1 depicts the logical, initial system-level architecture that was used as a starting point for the design process.

The system is separated into a Transmit side and a Receive side.

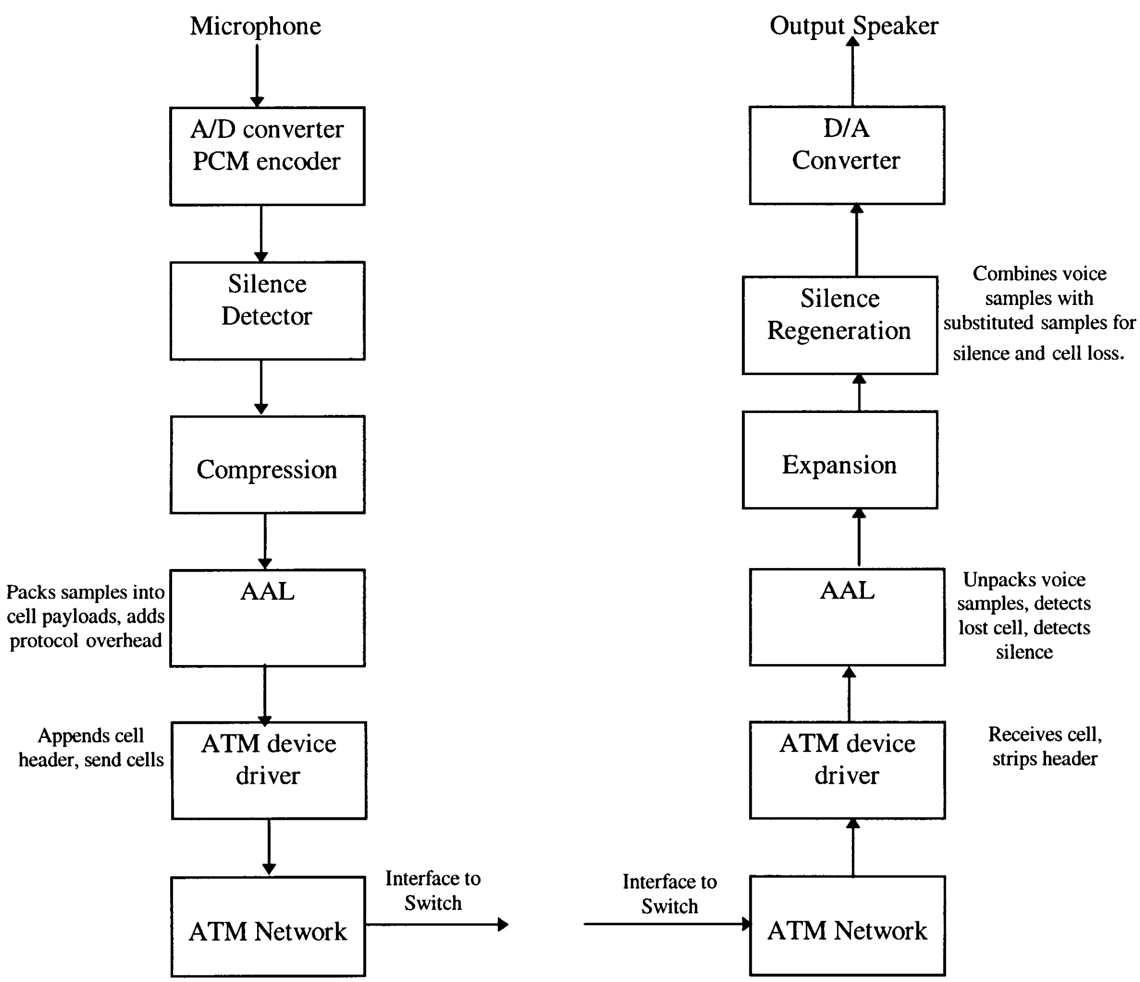


Figure 1 : Block Diagram of Proposed System

2.1. The Transmit Side

First, an Analog to Digital (A/D) converter samples the speech from a microphone and creates 16-bit, digitized samples. The speech samples are then analyzed in a silence detector by the VAD algorithm to determine if they constitute silence or voice. Only the samples characterized as voice are to be sent out over the ATM network. The voice is then compressed to further reduce the bandwidth it requires for transmission. The AAL software packs the voice sample into ATM payloads and appends any AAL overhead that may be necessary. The ATM driver then appends an ATM header to the payload before sending the ATM cell to the network.

2.2. Receive Side

The ATM driver receives the cells from the network and it strips off the ATM header and then sends the payloads to the AAL module. The AAL module is responsible for stripping off any AAL overhead and for detecting lost cells and gaps due to silence suppression. The payloads are passed on to the decompression module, where the speech samples are decompressed, and then to the silence regeneration module. In the silence regeneration module, the gaps detected by the AAL module due to silence intervals and cell losses are filled with generated payloads. The payloads generated for cell loss are different from the payloads generated for silence intervals. The exact contents of these generated payloads is determined in Chapter 5 where several alternatives are evaluated and compared.

The output stream of samples is then passed to the Digital to Analog (D/A) converter where the analog speech signal is reconstructed and passed to the loudspeaker for playback.

3. Preliminary Analysis

3.1. The VAD Algorithm

It was necessary for the VAD algorithm to be simple and easy to implement because it was hoped that any resulting product could be manufactured and installed inexpensively. Also, for reasons explained in Chapter 4, there was a time constraint on the processing time of the algorithm and the less complex the algorithm, the less the amount of time it would require. In addition to being simple, the algorithm still had to be able to distinguish between speech and silence reasonably well.

Based upon these criteria, a basic Root Mean Squared (RMS) algorithm was selected. It was supplied by the Communications Technology Division at COMSAT Labs. The basic premise upon which the algorithm operated was the fact that, in general, speech is louder than silence, and therefore a sampled voice waveform would have a larger amplitude during speech than during silence. The RMS of a group of samples could therefore be used as an index of the loudness of the signal represented by those samples.

$RMS = \sqrt{\frac{1}{N}\sum_{n=1}^{N} s_n^2}$, where N is the number of samples in the group and s_n is the n^{th} sample in the group.

The RMS of the set of samples could then be compared to a threshold value which demarcated the boundary between silence and speech. Sets with RMS values above this threshold are classified as voice while those with RMS values below the threshold are categorized as silence. It is possible to analyze the characteristics of a large example of speech in order to calculate what this threshold should be. However, in this case the threshold was found by trial and error, as is explained in Chapter 5.

In order to simplify matters significantly, it was decided that the group size on which the VAD algorithm operated would be identical to the number of 8 bit samples which could fit into an ATM payload. This number depended on which AAL was used. In this way, entire cells could be classified as either speech or silence, and complications arising from having to handle cells that were part speech and part silence, were avoided.

When the simple RMS algorithm was tested it became clear that sometimes silence was being removed from within words, causing unpleasant sounding effects. To prevent the removal of such inter-syllabic silences, a minimum wait period was introduced. The length of this wait period was configurable and a silence period would have to be at least as long as the wait period parameter before samples were removed. Chapter 5 details the analysis performed to determine the optimal value for the wait parameter.

In addition to the wait period, a delay of one group assembly time was introduced into the VAD algorithm. This delay was introduced in order to allow the categorization of the current group of samples to be based on the characterization of the group arriving after it. A group classed as silence which was immediately followed by a group classed as voice was treated as voice. This was done in order to counteract clipping of speech at the beginning of a voice burst which was especially noticeable with the compressed voice. The delay did not have to be an entire group assembly time in order to counteract clipping, but it was implemented this way for simplicity.

3.2. *The Degree of Compression*

Compression is one manner in which the bandwidth required for transmission can be reduced. When sampled linearly at 8kHz, successive speech samples are highly correlated and therefore contain a great deal of redundant information. This leads to high compressibility without any resulting degradation in speech quality.

Owing to the fixed size of ATM cells any compression of speech will incur a penalty of increased cell assembly delay. Cell assembly delay is the time it takes to obtain enough samples to fill an ATM cell payload. With 64kbps speech coding it would take 6ms worth of samples to fill an entire 48 byte payload. With 32kbps speech coding this would require 12ms. Since our major interest is high bandwidth utilization this delay is unavoidable as we must wait until the cell is full before it can be sent. However, if low delay was more important than efficiency, then partially filled cells could be used to reduce the cell assembly time.

Although voice traffic has been successfully compressed to bit rates as low as 8kbps without loss of toll quality and to 4.8kbps with near toll quality, the question arose as to whether or not the delay incurred by reducing bit rates to such low levels was tolerable within the context of a voice conversation. According to the International Telecommunication Union (ITU) end-to-end delays of up to about 25ms are tolerable if no echo-cancellation is employed [6]. With echo-cancellation, end-to-end delays of up to 150ms allow top quality, while up to 400ms is tolerable but may adversely affect some applications [7].

At 16kbps there would be 24ms of assembly delay if the entire payload was used for voice samples. In addition to this, delay will also be added by buffering required at playout to compensate for Cell Delay Variation (CDV) and buffering in the VAD algorithm to minimize clipping of speech. These additional delays may make echo cancellation a necessity for 16kbps voice transmission over ATM. With only 12ms of assembly delay at 32kbps, it is unlikely the additional delays for buffering at playout and in the VAD algorithm would make echo cancellation a necessity in a commercial implementation. In our implementation, however, the total cell assembly and buffering delay would be approximately 30ms for 32kbps transmission and 54ms for 16kbps transmission.

Compression to 32kbps compression was chosen over compression to 16kbps because in addition to the higher cell assembly delay, the 16kbps algorithm would also have a longer processing time due to its higher complexity. Also, the effects of cell loss are magnified at higher compression rates and the possibility of loss of as much as 24ms of speech at 16kbps was cause for concern. Another advantage of choosing to use 32kps was that a 32kps Adaptive Differential Pulse Code Modulation (ADPCM) algorithm was readily available from the Communications Technology Division at COMSAT which met the processing time constraints of the software implementation. These constraints are explained in chapter 4.

3.3. The Transmission Service and AAL

ATM technology is the medium of choice for the future because of its speed, and versatility. Its speed can be attributed to, among other things, the fixed length of the ATM cell. The versatility is in part a result of the ATM Adaptation Layer (AAL). The AAL, as the name suggests, adapts the services provided by the ATM network to the services required by the applications using it. The objective was to determine which particular combination of transmission service and AAL is best suited for the transfer of compressed voice with silence removal over ATM.

The variety of different transmission services supported by ATM are: Constant Bit Rate (CBR), Variable Bit Rate (VBR), Available Bit Rate (ABR) and Undetermined Bit Rate (UBR). With CBR service a certain amount of bandwidth is guaranteed to the connection for its duration. VBR service requires that a connection observe a certain maximum rate and also specifies a maximum sustained rate which the average rate of the connection should not exceed. Also, a certain minimum amount of bandwidth is also guaranteed to the user. ABR and UBR services are much more complicated but since there is no minimum amount of bandwidth that is guaranteed, these services are unsuitable for voice traffic. It is possible for either of these service to have zero bandwidth available to the user at any time, and if this happens to be when someone is speaking then this would be disastrous.

Since in general CBR service is more expensive than VBR service, it was preferable for our purposes to use a VBR service. Service providers are able to save money on VBR traffic by using the extra unused bandwidth for ABR and UBR service. However, with CBR service, bandwidth cannot be allocated elsewhere during silent periods because it must be used at all times for the connection. Since with voice traffic with silence removed there will be gaps when bandwidth is not used, VBR service is by far the better option.

VBR service offers two options, a real-time service and a non real-time service. The real-time service satisfies more stringent requirements on end-to-end timing of a connection. These include lower jitter and lower cell delay variation (CDV). Video and voice streams require real-time service because their quality is highly dependent upon timing. In these kinds of streaming applications a late cell is as bad as a lost cell because there is no use for a cell after it is late. Therefore, real-time VBR was chosen as the service over which compressed voices with silence removal would be sent.

The AAL's considered were AAL0, AAL1 and AAL5. AAL2 was not yet standardized and therefore could not be evaluated. For the purposes of the thesis, AAL3/4 was considered to be very similar to AAL5 in the details which mattered, with less bandwidth efficiency. Therefore, there was no need to evaluate AAL3/4.

The most important characteristic upon which the AAL's were compared was bandwidth efficiency. However, some consideration was given to how widely installed each was, as well as each AAL's functionality which might be useful for VBR voice. Timing recovery is one such function which is important for voice traffic, and cell loss detection is another. Some AAL's provide their own overhead bytes within the ATM payload for timing recovery and cell loss detection, while others do not and must rely on other methods

The applicability to compressed VBR voice of AAL0, AAL1 and AAL5 is discussed below.

3.3.1. AAL0

AAL0 provides no special functionality. The entire ATM payload of 48 bytes is used to carry user data and there is no additional AAL overhead. For the purposes of this thesis project this is considered the base case: 48 bytes per ATM cell are used for voice

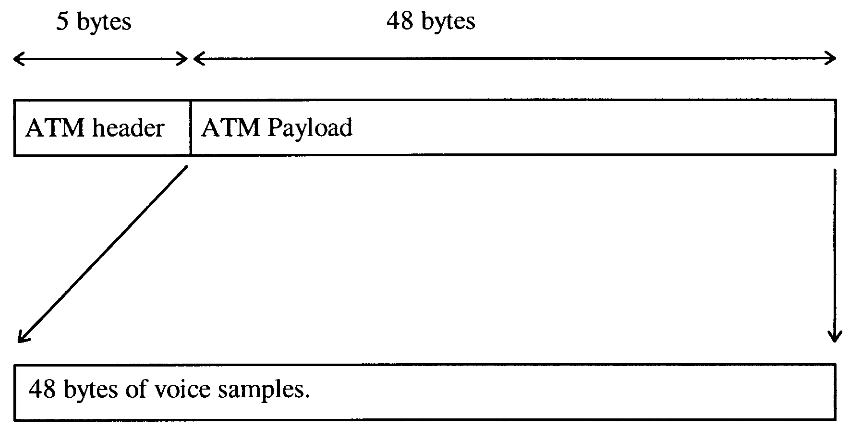


Figure 2 : AAL0 ATM Cell Structure

samples, and nominally this is considered 100% bandwidth efficiency. A simple diagram of the AAL0 ATM cell structure is shown in Figure 2.

At 64kbps it would take 6 ms of speech samples to assemble the 48 bytes of an ATM payload and likewise, loss of a single ATM cell results in 6 ms of lost speech. At 32 kbps the cell assembly time is 12 ms and 12ms of speech is lost when an ATM cell is lost..

AAL0 provides no overhead for timing conveyance therefore some other method of timing recovery must be used.

3.3.2. AAL1

AAL1 is specified in ITU-T document I.363 [8]. It has been standardized for several years and was designed to provide service for CBR, connection-oriented, time sensitive traffic. It is therefore used for circuit emulation services and is currently the AAL of choice for 64kbps voice traffic [9].

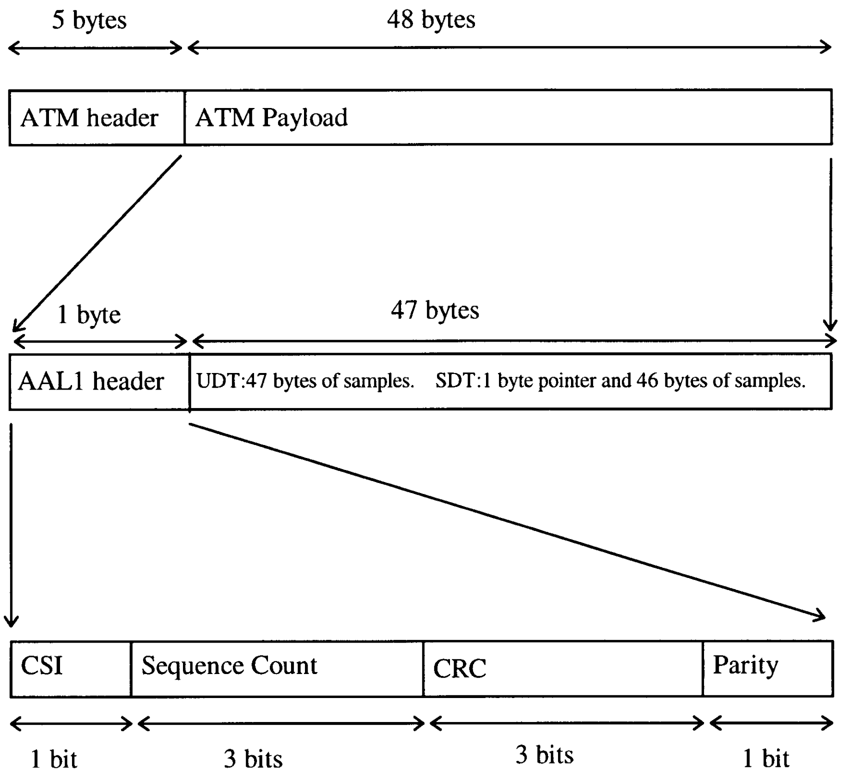
AAL1 provides two different modes of operation, Structured Data Transfer (SDT) and Unstructured Data Transfer (UDT). SDT is used when the user data is structured such that it consists of fixed length blocks longer than 1 byte. UDT is employed when the data has no such block structure. Independent of the mode of operation, the first byte of each ATM cell payload is reserved for an AAL1 header, and in UDT mode the remaining 47 bytes of the ATM payload are used for user data. In SDT mode there is a 1 byte pointer field in the second byte of the ATM payload to delineate the structure. The pointer is only allowed at a maximum in every other cell, so only 46 bytes are used for user data in those cells and 47 bytes are used in the other cells.

The AAL1 header also provides capability to transmit timing through use of a Synchronous Residual Time Stamp (SRTS) method. Alternatively, the time stamp can be ignored and an adaptive method such as buffer level monitoring can be used [8].

The structure of the AAL1 ATM cell is depicted in Figure 3. The AAL1 header consists of four fields. A 1 bit Convergence Sublayer Indication, a 3 bit ATM Sequence Count (SC), a 3 bit Cyclic Redundancy Check (CRC), and 1 bit for even parity.

In both SDT and UDT mode, in ATM cells with odd AAL1 sequence counts, the CSI bit is used to transmit the SRTS. In SDT the CSI bit in ATM cells with even sequence numbers is used to indicate the presence of the pointer field in the second byte of the ATM payload. This is the reason that the pointer can only be present at most every other cell. The function of the CSI bit in UDT in ATM cells with even sequence counts is for future study.

The SC field is a modulo 8 sequence counter. Using the sequence count enables the receiver to detect cell loss and cell misinsertion. Current specifications on AAL1 for voice suggest that the SC field be ignored unless it can be used without causing processing delay. Instead other methods for detecting loss such as buffer fill-level monitoring are proposed [9].



CSI : Convergence Sublayer Indication; used to transfer Synchronous Residual Time Stamp for Source Clock Recovery, and for Indication of Structured Data Transfer.
Sequence Count : Modulo 8 sequence counter use in detection of cell loss and misinsertion.
CRC : Cyclic Redundancy Check, used for error detection and correction of single bit errors in the AAL1 header.
Parity : Single bit to maintain even parity and used for error detection in the AAL1 header.

Figure 3 : AAL1 ATM Cell Structure

The combination of the Parity bit and the CRC is used for error detection and correction in the AAL1 header. All single bit errors can be corrected and multi-bit errors can be detected. As is the case with AAL0, the payload is not protected.

The Pointer byte for SDT mode contains two fields. The first bit is reserved for future standardization. The next seven bits make up the pointer field and contain the offset in bytes to the next start of a structure block in the ATM cell containing the pointer or the cell immediately following. The pointer can only occur in ATM cells with even

AAL1 sequence numbers. Exactly how often the pointer is employed determines how robust structure recovery is, and has not been fully standardized as yet. SDT structure blocks can have lengths up to 93 bytes.

The AAL1 header means that only 46 or 47 bytes which translates to 95.83% or 97.92% of the ATM cell payload respectively, can be used for samples. This represents a slight decrease over AAL0. The advantage of AAL1 is the functionality that the time stamp and modulo 8 sequence numbering in the 1 byte AAL1 header provide.

One possible disadvantage noted in a contribution to the ATM Forum [10] is that 47 bytes per cell is a cumbersome number that may not be easily supported by existing hardware buses, for example those in desktop computers.

It takes 5.875 ms at 64kbps to completely fill 47 bytes of an AAL1 cell and 11.75ms to fill them at 32kbps. Similarly, a loss of one cell corresponds to the loss of 5.875 ms of speech at 64kbps and to the loss of 11.75ms of speech at 32kbps.

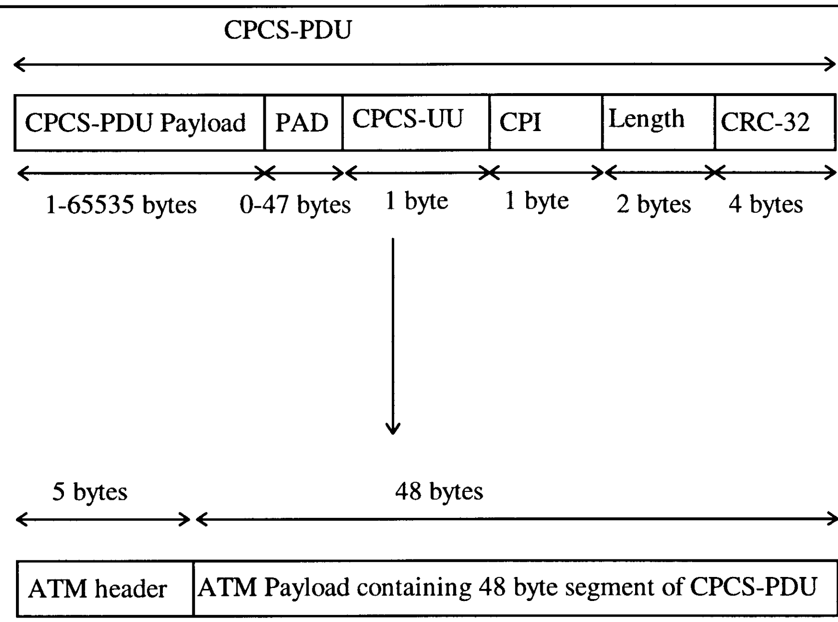
At present UDT is the mode used to transport 64kbps voice traffic. However, SDT is being investigated for possible use with multiplexing of traffic from multiple sources in a regular block structure into the same ATM cell [11].

3.3.3. AAL5

AAL5 is specified in ITU-T document I.363 [8]. It is meant to carry VBR traffic that requires no timing relationship between the transmitter and the receiver. It contains two sublayers: the upper sublayer is the Convergence Sublayer (CS), and the lower sublayer is the Segmentation and Reassembly (SAR) sublayer. The CS is further subdivided into two sublayers: the upper being the Service Specific Convergence Sublayer

(SSCS), and the lower being the Common Part Convergence Sublayer. At present only the CPCS has been defined.

The unit of data produced by the CPCS is called the Common Part Convergence Sublayer Protocol Data Unit (CPCS-PDU). It consists of three fields: a payload, a padding field, and a trailer. See Figure 4 for an illustration of the various fields. The CPCS-PDU payload contains the actual user data and can have a length from 1 byte to 65535 bytes. The padding field consists of dummy bytes inserted between the CPCS-PDU payload and trailer, to make the length of the entire CPCS-PDU an exact multiple of 48 bytes.



PAD : Padding bytes to make the CPCS-PDU length a multiple of 48 bytes.
CPCS-UU : User-to-User Information.
CPI : Common Part Indication. Currently it makes the CPCS trailer 64 bytes long. Other uses are for future study.
Length : Contains the length of the CPCS-PDU payload.
CRC-32 : Cyclic Redundancy Check providing 32 bits of error detection across the entire CPCS-PDU.

Figure 4 : AAL5 ATM Cell Structure

The CPCS-PDU trailer is eight bytes long and has four fields: a one byte CPCS User-to User field (CPCS-UU), a one byte Common Part Indicator (CPI), a sixteen bit Length field, and a 4 byte Cyclic Redundancy Check (CRC-32).

The 1 byte CPCS-UU field is used to transparently transfer User-to-User information between users. The 1 byte Common Part Indicator (CPI) as yet has no other function than to make the trailer 64 bytes long. The Length Field indicates the length of the CPCS-PDU payload in bytes. The field is 16 bits long and therefore the payload can be 1 to 65,535 bytes long.

The 32 bit CRC provides error detection for the entire CPCS-PDU. The ITU-T Recommendation I.363 does not list error correction as an option for AAL5. However, it does list an option to either completely discard CPCS-PDU's with errors or to allow delivery of such CPCS-PDU's to the higher layers with some error indication. The exact details of the latter option have not been finalized as yet and therefore it is not considered.

The CPCS-PDU, which is always a multiple of 48 bytes due to padding, is split into segments of 48 bytes by the SAR sublayer at the transmitter and each 48 byte segment becomes an ATM cell payload. The SAR sublayer at the receiver reassembles the ATM payloads to form the CPCS-PDU's. The ATM User-to-User indication in the ATM cell header is used to indicated which cells contain the end of a CPCS-PDU. If only one ATM cell is lost then the entire CPCS-PDU to which its contents belonged will be discarded. This is known as cell loss multiplication [12].

AAL5 has been considered as an option for CBR voice. Its major drawback is that the CPCS trailer is so large that it severely affects bandwidth efficiency. If one trailer is included in every ATM cell then only 40 bytes per ATM cell are left for voice samples. If the trailer is not included in every cell then in addition to possible cell loss multiplication, due to one lost cell causing the discard of the entire CPCS-PDU, there is additional delay at the receiver where the cell containing the trailer must be awaited before any of the samples in the CPCS-PDU can be released. This delay is prohibitive. Also, unlike AAL1, AAL5 provides no timing transmission features. Some companies have already implemented voice over AAL5 with their own proprietary method for timing recovery [11].

There is currently a proposal before the ATM-Forum [13], which suggests the use of AAL5 for voice over ATM. In this proposal there is one CPCS-PDU trailer and 40 bytes of voice samples per ATM cell. 40 out of 48 bytes represents only 83.3% bandwidth efficiency, which is significantly less than AAL0 and AAL1. Since AAL5 does not provide any timing transmission mechanisms the proposal uses an adaptive method. The proposal cites the fact that AAL5 is already supported in much of the ATM equipment on the market and its applicability to other services, as its major advantages.

3.3.4. The Chosen AAL

AAL1 with UDT was chosen to implement voice over ATM with compression and silence removal. It provided not only high bandwidth efficiency, but also useful timing and sequence number functionality which was not present in AAL0. SDT was found to be unnecessary for the purpose of a single voice connection.

AAL5 was ruled out mainly because for the purposes of speech traffic its extensive error detection scheme was unnecessary. Voice is highly resilient to bit errors and so discarding an entire cell or multiple cells due to the occurrence of bit errors was senseless and definitely not worth all the bandwidth it required.

In addition to the AAL1 header, another one byte header, referred to as a voice header, was introduced by COMSAT. It was situated in the second byte of the ATM payload and its introduction could be viewed as introducing a new AAL sublayer just above the AAL1 layer. The voice header contained a one bit indication of the end of a speech burst. It was set to zero unless the cell containing it was the last cell in a speech burst in which case it was set to one. The rest of the voice header was reserved for future extension by COMSAT.

With the one byte of AAL1 header and the one byte of voice header in the ATM cell payload there are 46 bytes remaining for voice data. This represents a bandwidth efficiency of 95.8%.

4. Implementation

Once the VAD algorithm, the compression algorithm, the transmission service, and AAL had been determined, the next step was to actually implement the system and demonstrate its effectiveness.

The required functionality of the implementation included being able to make a connection and compare performance with and without compression and silence removal. The system needed to be flexible enough to be altered easily if the need arose. This led to the decision that the proof of concept system should be implemented in software rather than in hardware.

The COMSAT proprietary real-time operating system , COSNIX, provided the ideal tool for such a system. It simulates a real-time environment by scheduling simultaneous events sequentially in such a way that behavior is no different from that in a hardware system. In order to properly recreate real-time behavior, certain time constraints had to be observed. In particular, the sum of all the processing times of all the modules, needed to be less than one cell assembly time, 11.5ms, if the system was to work in full duplex. If this constraint could not be observed then the sequential execution of the COSNIX operation system could not create the illusion of simultaneous hardware execution.

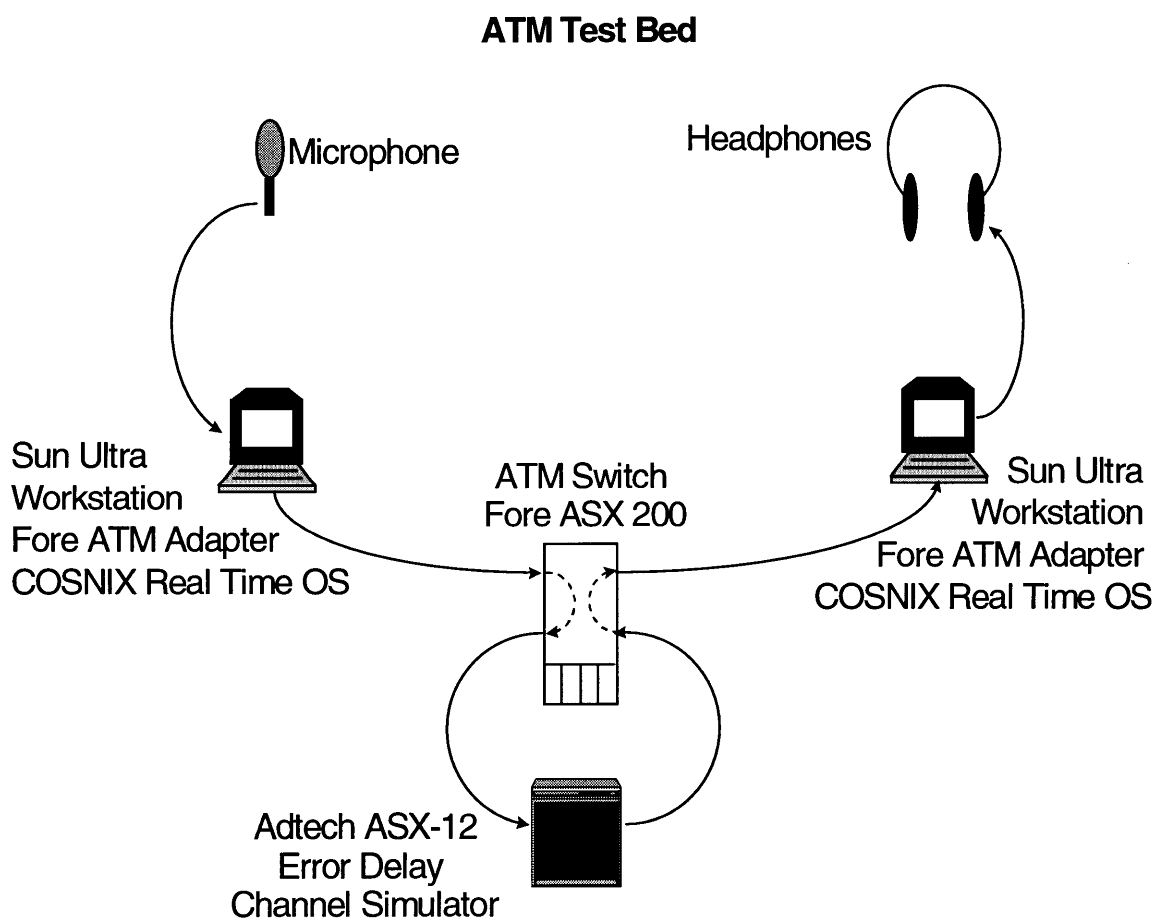


Figure 5 : The ATM Test Bed

The system was implemented on two Sun Ultra workstations equipped with headphones, a microphone, and a FORE Systems ATM Card. The Sun workstations were connected to each other through their respective ATM Cards via the ATM test bed. It consisted of a looped back FORE ATM Switch connected through an ADTECH SX/12

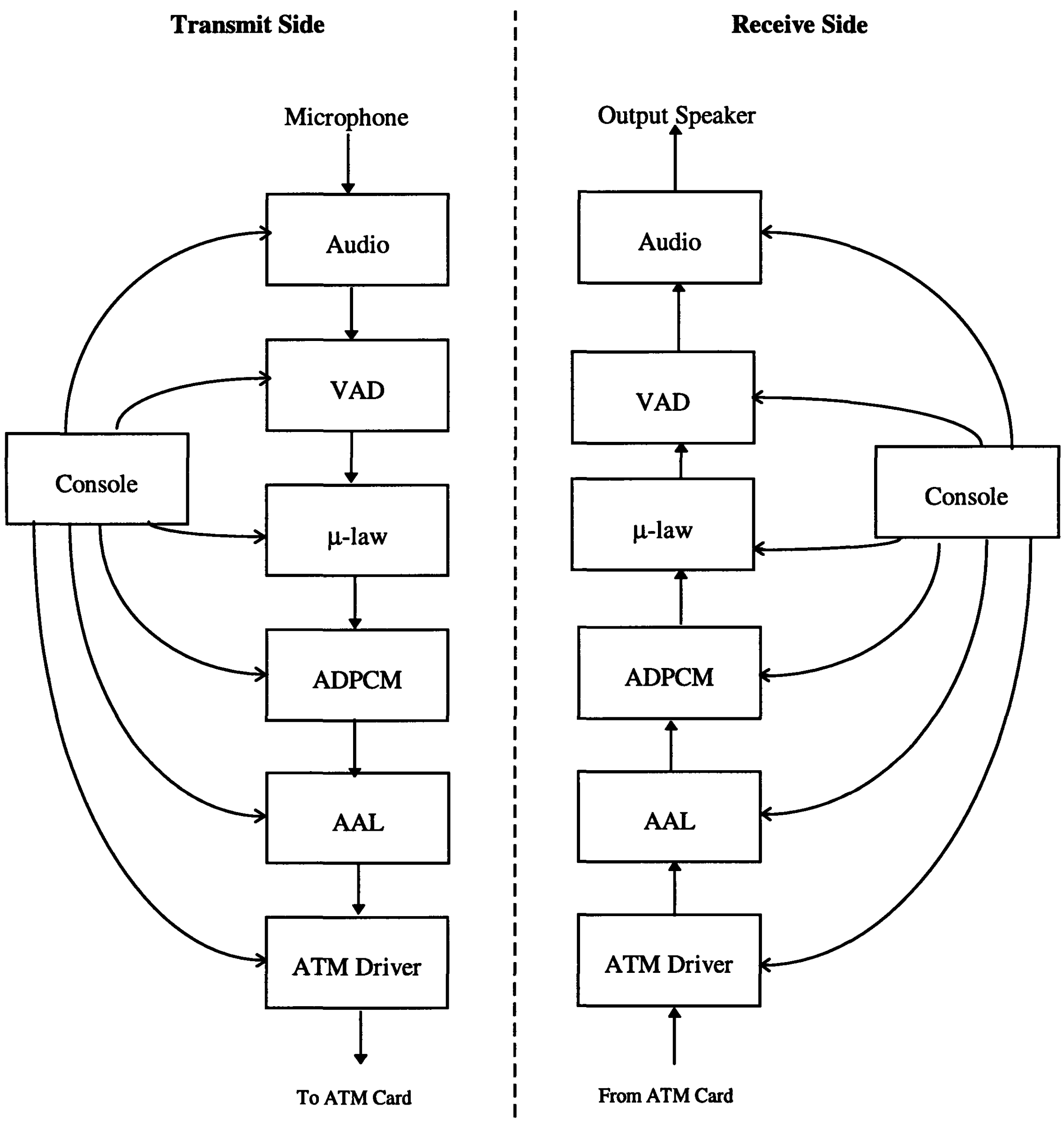


Figure 6 : The Software System Modules

Data Channel Simulator. Figure 5 contains a diagram of the ATM test bed.

The software system consisted of seven modules: the Audio module, the Voice Activity Detection (VAD) module, the μ-Law Companding module, the ADPCM Compression module, the AAL module, ATM Driver module, and the Console module. Figure 6 contains a block diagram of the software modules.

4.1. The Console Module

The Console module provides a user interface to all of the other modules. Through this module the user can specify command line arguments which start and stop the system, change various parameters of operation, and display the operation statistics.

The parameters that can be changed through this module are: the VAD threshold, the length of the wait period in cells, the percentage of cells lost, the compression toggle, the source of the input samples, and the destination of the output samples. Possible sources include a microphone or a voice file, while destinations include headphones and voice files.

The statistics which can be displayed include the total number of frames declared as silence or voice by the VAD module, as well as the number of cells sent and the number of cells discarded by the transmitter AAL module.

4.2. The Transmit Side

4.2.1. The Audio Module

The Audio module performs the sampling of speech from the microphone or from an input voice file. This sampling occurs at a rate of 8kHz and the samples are 16 bits wide. Groups of 46 consecutive samples are assembled into frames. Each frame is then passed to the VAD module.

4.2.2. The Voice Activity Detection (VAD) Module

The VAD module measures the Root Mean Square (RMS) of all of the samples in each frame received from the Audio module. Based upon comparison of the RMS value with a preset threshold, each frame of samples is labeled as either silence or voice. The label consists of one bit that is added temporarily to the frame.

$RMS = \sqrt{\frac{1}{46}\sum_{n=1}^{46} s_n^2}$, where s_n is the n^{th} sample in the frame. If the RMS is less than the threshold then the frame's label is set to zero signifying silence. If the RMS is greater than or equal to the threshold then the label is set to one, signifying voice.

Each frame is delayed in this module by one frame assembly time, 5.75ms, because the label assigned to a particular frame can depend on that of the next frame. For example, if a frame has been labeled as silence and then the next frame is labeled as voice, the first frame's label is changed to voice. Each Frame is then passed on to the μ-Law module along with its label.

The reasons that the one frame buffer was introduced were two-fold. Firstly, the very simple VAD algorithm was incapable of properly detecting the exact start of a speech burst. This was because if the burst commenced toward the end of a frame the averaging done by the RMS computation might cause the increased activity towards the end of a frame to go unnoticed. Secondly, the nature of the ADPCM algorithm is such that it has memory. This means that it uses past samples to generate estimates of current samples when performing decompression. Therefore whenever a gap due to silence occurred at the receiver, the decompression algorithm would need time to reconverge before it could properly decompress the samples at the start of a new speech burst. By sending the last frame classified as silence before a speech burst, the algorithm at the receiver would have some samples to regain convergence on before the actual speech burst started.

In reality, the buffer does not have to be an entire 46-byte frame in order to perform these functions. A much smaller number of samples could have been buffered thereby reducing the delay introduced, but it was implemented this way for simplicity.

4.2.3. The μ-Law Companding Module

This module was necessary because the ADPCM algorithm need μ-Law companded samples while the VAD algorithm operates on linear samples.

In the transmit direction, this module merely takes each frame received from the VAD module and compands each 16 bit sample within the frame, via the μ-Law rule, into an 8 bit sample. It then passes each frame along to the ADPCM compression module along with its VAD label.

4.2.4. The ADPCM Compression Module

If compression has been activated by the console module, then each sample within each frame received from the μ-Law module is compressed from 8 bits down to 4 bits using the ADPCM compression algorithm. If compression has not been activated then no compression is performed. Either way the frame is then passed on to the AAL module along with its VAD label.

4.2.5. The AAL Module

This module assembles the ATM cell payload. If compression is enabled, then every two frames received from the ADPCM module are combined to form 46 bytes of voice samples. If either of the two frames is labeled voice then the resulting payload is labeled voice, otherwise it is considered silence. If compression is disabled then only one frame of 46 samples of 8 bits each is required for a payload. One byte of AAL1 overhead is added followed by the one byte voice header to form the 48 byte ATM payload.

All ATM payloads that were labeled as voice are sent to the ATM driver module. Whenever a silence payload is observed after a voice payload, a number of consecutive silence payloads equal to the wait parameter are sent on to the ATM driver module. All other silence payloads are not sent, but instead are discarded. The wait parameter is set by the user via the console module. The one bit End of Burst indicator is set to one in the voice header of the last silence payload of the wait period.

During long periods of silence an update payload is sent every 44 cells. This was added because some of the schemes for background noise generation required occasional updates. At 32kbps this gives an update approximately every 0.5s. The frequency of this update was chosen arbitrarily for this study. Its necessity and frequency are left as a matter for future investigation. The update cells contain the End-of-Burst bit and they are not played out at the receiver.

4.2.6. The ATM Driver Module

In this module ATM headers are added to the 48 byte ATM payloads received from the AAL module. The ATM cells are then transmitted to the ATM switch through the ATM card in the Sun Ultra workstation.

In order to aid in the testing of the cell loss compensation, this module is able to randomly drop a certain percentage of cells as specified by the user to the console module. This was necessary because the ADTECH channel simulator was only able to generate bit errors and burst errors but could not be specifically instructed to drop a cell. In this way cell losses could be simulated and isolated from other types of errors.

4.3. The Receive Side

4.3.1. The ATM Driver Module

This ATM driver module receives ATM cells from the network and strips off the 5 byte ATM header before passing the payloads to the AAL module.

4.3.2. The AAL Module

In the receive direction, the AAL module strips off the two AAL header bytes. It then extracts the End-of-Burst bit from the voice header. This module contains a window algorithm which is used to detect lost cells and silence gaps. Based on the Maximum Cell Delay Variation (CDV) specified in the Quality of Service (QoS) specifications or the service contract, the algorithm is able to determine a specific window of time in which the next cell is expected to arrive. If a cell does arrive within the window then its 46 byte voice payload is sent on to the ADPCM module. If no cell arrives within this window then it is considered lost and a dummy payload is sent to the ADPCM algorithm on the receive side indicating that a cell is missing and the gap needs to be filled with generated samples. Using the presence or absence of the End-of-Burst bit in the payload preceding the loss, the module is able to determine whether the loss is due to silence removal or cell loss. A different dummy payload is used for each case so they are distinguishable from each other and from payloads which actually contain samples.

The window algorithm operates as follows: take t_0 to be the time of a arrival of a cell. Then the next cell is expected to arrive between t_0 and t_1 , where t_1 is t_0 plus one maximum CDV plus one cell assembly delay. If no cell arrives before this time then the expected cell is considered lost or removed and a dummy payload signifying cell loss or removal is sent to the ADPCM module. The next cell is then awaited until t_1 plus one cell assembly delay time. If still no cell arrives then another cell assembly delay time is added to the end of the window and a dummy payload signifying cell removal is sent to the ADPCM module. This is continued until a cell does arrive. When a cell does arrive, its arrival time is taken as the new t_0 and the algorithm restarts.

This algorithm is not very robust and is not practical for commercial use. In order for this algorithm to work correctly the CDV must always be less than one half of the cell assembly delay, otherwise windows will overlap and cells arriving late will be indistinguishable from cells arriving early. If the maximum CDV is exceeded the algorithm will fail catastrophically. A better implementation might make use of the sequence numbering and time stamp provided by AAL1. Provided the maximum CDV satisfies the constraint mentioned above and is never exceeded, then the algorithm will always be able to link a particular cell to a particular window, whether it or its predecessor was late or early.

4.3.3. The ADPCM Compression Module

The payloads are received from the AAL module and the samples within the payloads are decompressed if they had been compressed. If the payload is a dummy payload then no processing is performed on its contents. However, if compression is active, then the dummy payloads are doubled in size. The payloads of 8 bit samples and the dummy payloads are then passed to the μ-Law module.

4.3.4. The μ-Law Companding Module

The μ-law module simply expands each payload of μ-law companded samples into linear 16 bit samples, and then passes the payloads on to the VAD module. The dummy payloads are simply doubled in size but no processing is performed on their contents.

4.3.5. The Voice Activity Detection (VAD) Module

The VAD module replaces the dummy payloads in the stream of payloads received from the μ-law module, with payloads of generated samples. Nothing is done to payloads that contain voice samples. The payloads from update cells are treated as dummy payloads but their contents may be used for silence regeneration.

If the dummy payload is filling a gap due to silence suppression at the transmit side then some scheme for reproduction of background noise is used to replace the dummy payload with a payload of comfort noise samples. If the gap is due to cell loss then other methods are employed to fill this gap with the least amount of quality degradation.

Various schemes were tried for both the silence gaps and cell loss gaps and these are described in the chapter 5.

After the gaps are filled with payloads of generated samples, the stream of payloads is passed on to the Audio module for playback.

4.3.6. The Audio Module

The receive audio module takes a payload of samples from the VAD module and conveys it as a stream of samples to the audio hardware output device, where the sound is played out, or to an output voice file. There was a playout buffer in this module which was primed with one payload of samples. The buffer was necessary in order to compensate for cell delay variation. This buffer introduced an amount of delay equal to the cell assembly delay, which was 5.75 ms at 64kbps and 11.5ms at 32kbps. It was not necessary for the buffer to introduce a full cell assembly delay but it was implemented this way for increased simplicity. In theory the buffer only needed to contain enough samples as necessary to compensate for the cell delay variation.

5. Parameter Tuning and Configuration Optimization

The flexibility and modularity of the software system enabled various alternative configurations of the system to be tried in order to determine the optimal one. For the analysis of tuning and configuration described below, the same voice file was used for uniformity. It consisted of a male voice saying several sentences with a slight pause in between each. It was recorded in a laboratory environment at COMSAT and so contained actual background noise. These are the sentences:

1) It seems simple to my mind.

2) She says that she adores men.

3) You will have to be very quiet.

4) There was nothing to be seen.

5) I want a minute with the inspector.

6) The juice of lemons makes fine punch.

7) He punched viciously at the ball.

8) It's easy to tell the depth of a well.

5.1. VAD Tuning

First the VAD threshold was raised continuously from zero while the same voice file was played repeatedly until the point at which none of the background noise was transmitted. This was done manually by replacing any gaps at the receiver due to silence removal with a very loud, disturbing noise before playback. The lowest threshold was found where all the periods between sentences were filled with the objectionable noise. At this chosen threshold, it was observed that the disturbing noise was occurring not only throughout periods of silence but also in the middle of some words. This indicated that inter-syllabic silences were being removed as well. Since this effect was not desired, it was necessary to implement a mechanism to ensure that only silences beyond a certain minimum length were removed.

At the threshold determined previously, the same voice file was played over and over and the wait parameter was increased from zero cells until the disturbing noise was no longer heard in the middle of words. The minimum number of cells that had to be waited before this occurred was chosen as the ideal wait period for the rest of the analysis.

The initial determination of these variables was performed with compression deactivated. The resulting values were a threshold of 400 and a wait period of 26 cells. The threshold value has no units and is dependent on the Analog to Digital converter performing the sampling. It would also have to be reset every time the ambient noise level changed. Much emphasis was not placed on in depth VAD threshold analysis since it was not within the scope of the thesis. The wait period of 26 cells corresponds to the findings of other researchers. It has been demonstrated that 99.56% of continuous speech segments have periods of silence smaller than 150ms [14]. Twenty-six cells at 64kbps contain 149.5ms of samples.

When 32kbps ADPCM compression was added into the system, additional adjustments had to be made. Not surprisingly, the ideal number of cells to wait was successfully halved and became 13. However, the VAD threshold had to be lowered to 150 and it became necessary to label the VAD before the start of a speech burst as voice even though the threshold classified it as silence. This was necessary to remove the click that was audible at the receiver at the start of every speech burst during the time that the ADPCM algorithm was trying to converge after a gap of silence. The silence frame that was sent first afforded the algorithm an opportunity to converge before actual speech started to arrive at the receiver.

Figure 7 contains a histogram showing the distribution of the RMS values of the 46-byte frames in the file of sentences.

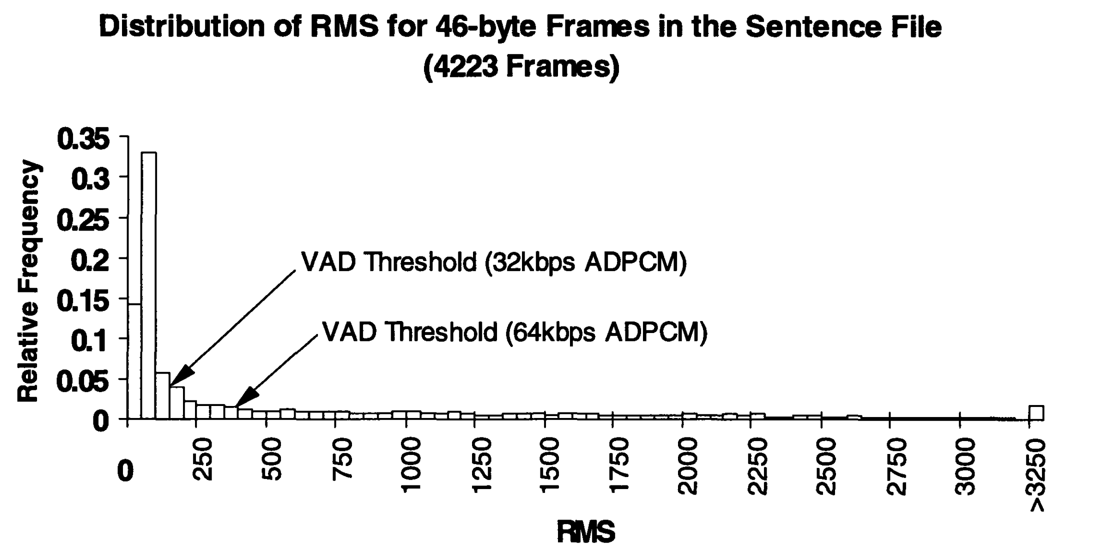


Figure 7 : Distribution of RMS of 46-byte Frames in the Sentence File.

5.2. Comfort Noise Generation

5.2.1. Procedure:

After the optimal VAD threshold and wait period had been determined, the best way to produce comfort noise at the receiver during silent gaps was addressed. Three different methods were compared.

First, nothing at all was played during the silence gaps and the resulting effects were evaluated. A second method was to utilize the cells from during the wait period before the silence gap, and play them back repeatedly throughout the gap itself. Another method tried involved using the RMS value of the silence that was sent during the wait period to estimate the level of the background noise, and then generating white, Gaussian noise around that level.

5.2.2. Results

The simplest way to handle periods of removed silence at the receiver was to play nothing out. However, this sounded completely unnatural. In a normal situation the listener would be tempted to say ‘hello’ intermittently because it appears that the line has gone dead.

The second method in which the silence sent during the wait period was repeated led to an unpleasant periodic effect no matter how many of the cells were used.

The most successful method was found to be using the RMS value of the silence that was sent during the wait period to estimate the level of the background noise, and then generating white Gaussian noise around that level. Although there was some difference from the quality of the actual background noise, since their levels were quite similar the effects were not unpleasant. It was similar to the effects one might notice currently in some overseas calls. The RMS value of the samples in the update cells sent during silence periods were used to update the estimate of the level of the background noise.

5.3. Cell Loss Compensation

5.3.1. Procedure

Two different methods of cell loss compensation, extrapolation and interpolation, were tested. The sound quality obtained with compensation was compared to that obtained when no compensation was performed.

The first method that was tried to minimize the effects of cell loss was extrapolation. Here, when a cell was lost, the last cell was repeated with some smoothing at both its ends in order to fill the gap and reduce any discontinuities. This smoothing involved creating a more gradual progression of samples between the generated cell and the cell before it using a linear combination of samples from each.

The second method tried was interpolation. Here both the cell after the loss and the cell before the loss were used to generate an approximation of the lost cell. Here smoothing was employed also to reduce the effects of discontinuities. In order for the cell after the loss to be used in the interpolation process an additional cell time worth of delay had to be introduced into the system.

This part of the analysis assumed that loss of multiple consecutive cells occurred with negligible frequency.

5.3.2. Results

When no compensation was performed a loud click was heard when a cell was lost. The click was especially conspicuous when ADPCM compression was employed because each cell accounted for more samples and the nature of the ADPCM algorithm is such that it needs time to converge when a gap occurred. Although cell loss is not a very frequent occurrence on ATM networks, especially when users observe the requirements of their respective service contracts, it is still preferable to minimize the detrimental effects of cell loss.

Both extrapolation and interpolation succeeded in substantially muffling the loud clicks caused by cell loss. There was no apparent improvement in using interpolation over extrapolation and therefore it was determined that extrapolation would suffice and that interpolation was not worth the extra delay.

6. Evaluation of Speech Quality and Bandwidth Efficiency

6.1. Speech Quality

The same file with the sequence of sentences used for the tuning and configuration analysis was used for the evaluation of speech quality. The speech quality in a particular configuration was evaluated by comparison with how the same speech sounded when played at 64kbps with no silence removal. The evaluation was done by having several people to listen to the different instances of the same speech file and getting feedback from them as to whether there were noticeable differences in quality.

It was quite apparent there was absolutely no degradation in quality as a result of the ADPCM compression. When 64kbps speech was compared with 32kbps speech with no silence removal, they were indistinguishable. When silence removal was introduced then as long as a sufficiently long wait parameter was used, there was no difference in quality for the periods of speech because all the speech itself was played. The only differences occurred in the silence periods, where there was a need for adequate silence regeneration in order to satisfy the listeners. The Gaussian white noise generation algorithm provided satisfactory results.

6.2. Bandwidth Savings

For the evaluation of bandwidth savings a two-way conversation was used with the parameter settings obtained in the previous chapter. The conversation was a simulated discussion between a motel receptionist and a client making a reservation. A transcript of the simulated conversation follows:

Receptionist : Good morning, South East Motel. How may I help you?

Client : Yes, I'm calling to confirm a room reservation I made for this weekend.

Receptionist : May I have your name sir?

Client : Dover, Ben Dover.

Receptionist : Just a moment let me see … (short pause) … I'm sorry, I'm not seeing a reservation under that name.

Client : But that can't be. I'm certain I made one two days ago.

Receptionist : Well I'm sorry sir I double checked. There doesn't seem to be one in the system. Would you like to go ahead and make a reservation now?

Client : Yes, please.

Figure 8 contains a histogram of the distribution of RMS values of the 46-byte frames in the two-way conversation.

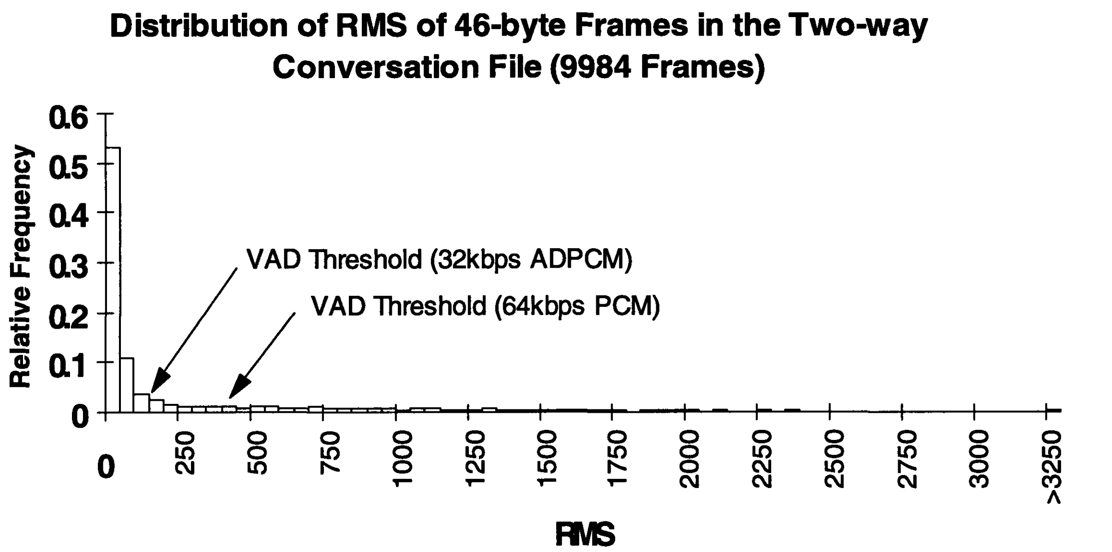


Figure 8 : Distribution of RMS of 46-byte Frames in the Two-way Conversation File

As should be completely obvious, employment of 32kbps ADPCM with no silence suppression resulted in a 50% bandwidth savings over 64kbps voice transmission. When the ADPCM compression and the silence suppression were combined on the two-way conversation described above, using the final optimal configuration for the best speech quality, the unidirectional bandwidth savings obtained were very encouraging. The receptionist side of the conversation was reduced to 30% of the original bandwidth it occupied when transmitted at 64kbps. This represented 70% bandwidth savings. The client side was reduced to 20% of the original bandwidth resulting in 80% bandwidth savings.

Bandwidth savings is defined as:

$(1-\frac{A}{B})\times 100\%$, where A is the number of cells sent at 32kbps with silence removed and B is the number of cells sent at 64kbps with no silence removal.

The results obtained in a real telephone application clearly will depend heavily on the type of conversation the system is used on. However, even on the one-sided sequence of sentences described in section 5.1 which clearly is more dense with voice activity than a two-way conversation, 62% bandwidth savings was attained.

7. Conclusions and Suggestions for Future

Based on the results obtained, it can be concluded that VBR, compressed voice does provide a viable opportunity for bandwidth savings on ATM networks. These savings can been obtained without noticeable degradation in speech quality. If such basic algorithms as were used in this project provided such remarkable results, then more in-depth study and sophisticated techniques could provide even more substantial savings.

If in the future a more sophisticated VAD algorithm was employed the bandwidth savings could be increased. Similarly, more sophisticated interpolation and extrapolation techniques already exist for speech and if these were implemented in the system to compensate for cell loss, the muffled clicks may be reduced even further.

The ITU-T has recommended a scheme for using a 4-bit background noise level indicator to aid in silence regeneration when silence suppression is employed [15]. This 4-bit indicator could be included in COMSAT's one-byte voice header along with some error detection, and would provide a better estimate of the background noise level than the scheme used here, because the transmitter has access to all of the silence samples, while the receiver only has access to the few that are sent.

8. References

[1] D. Bertsekas, R. Gallager "Data Networks," 2nd edition, pp.129.

[2] S. Annukka Piironen, "Multimedia Traffic Management and Congestion Control in Satellite ATM Networks," MIT Master's Thesis, May 1994.

[3] J. Hartford, "ATM Must Make Way for Voice," LAN Times, July 3, 1995, p.33.

[4] P.G. Drago, A.M. Molinari, and F.C Vagliani, "Digital Dynamic Speech Detectors," IEEE Transactions on Communications, Vol. 26, No. 1, pp. 140-145, January 1978.

[5] "Digital Circuit Multiplication Equipment Using ADPCM and Digital Speech Interpolation," ITU-T Recommendation G.763.

[6] "Control of Talker Echo," ITU-T Recommendation G.131.

[7] "One-way Transmission Time," ITU-T Recommendation G.114.

[8] "B-ISDN ATM Adaptation Layer (AAL) Specifications," ITU-T Recommendation I.363.

[9] "Voice and Telephony Over ATM to the Desktop," ATM Forum Doc. #95-0917, August 1996.

[10] L.G. Roberts, "The Usefulness of AAL1 for Voice to the Desktop," ATM Forum Doc. #96-0612, April 1996.

[11] D.J. Wright, "voice Over ATM: An Evaluation of Implementation Alternatives," IEEE Communications Magazine, May 1996, pp. 72-80.

[12] J.H. Baldwin et al., "Error and Loss Resilience of Proposed AAL-CUs," ATM Forum Doc. #96-0843, June 1996.

[13] S. Kumar, "AAL5 Voice for Private Networks," ATM Forum Doc. #96-0950, August 1996.

[14] J.F. Lynch Jr., L.G. Josenhans, and R.E. Crochiere, "Speech/Silence Segmentation for Real-Time Coding via Rule Based Adaptive Endpoint Detection, IEEE International Conference Acoustics, Speech, and Signal Processing (ICASSP-87), Dallas, TX, USA, April 6-9, 1987, pp. 1348-1351.

[15] "Voice Packetization - Packetized Voice Protocols," ITU-T Recommendation G.764.

9. Appendix

Results from Sentence File:

Wait Period With no Silence Regeneration

Cells	Transmitted	Received	Utilization	Threshold	Rate	CD (%)	Wait
4223	4223	0	100	0	64	0	N/A
4223	2358	2358	55	150	64	0	1
4223	2897	2897	68	150	64	0	13
4223	3162	3162	74	150	64	0	26

Wait Period With Gaussian Silence Regeneration:

Cells	Transmitted	Received	Utilization	Threshold	Rate	CD (%)	Wait
4223	2358	2358	55	150	64	0	1
4223	2897	2897	68	150	64	0	13
4223	3162	3162	74	150	64	0	26

Silence Regeneration

Cells	Transmitted	Received	Utilization	Threshold	Rate	Regen.	Wait
4223	3162	3162	74	150	64	Repetition	26
4223	3162	3162	74	150	64	Gaussian	26
4223	3162	3162	74	150	64	None	26

Cell Loss Compensation

Cells	Transmitted	Received	Utilization	Threshold	Rate	Compen.	CD (%)
4223	2111	2070	50	0	32	Repetition	2
4223	2111	2076	50	0	32	Gaussian	2
4223	2111	2066	50	0	32	None	2

Results from Two-way Conversation at 32kbps ADPCM with silence removal:

Side	Total Cells	Transmitted	Received	Threshold	Wait	Utilization
Client	2495	1002	1002	150	13	20
Receptionist	2495	1497	998	150	13	30

Notes:
CD is Cell Drop Percentage.
Utilization is $100 \times \frac{A}{B}$, where A is the number of cells transmitted and B is number of cells required for 64kbps PCM tranmission with no silence removal.