Performance Evaluation of Voice Compression for VoIP

By

Mr. Erez Agmoni

Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science
in Telecommunications Science
Assumption University

February, 2003

Performance Evaluation of Voice Compression for VoIP

by

Mr. Erez Agmoni

Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Telecommunications Science Assumption University

February, 2003

The Faculty of Science and Technology

Master Thesis Approval

Thesis Title	Performance Evaluation of Voice Compression for VoIP
By	Mr. Erez Agemoni
Thesis Advisor	Asst.Prof.Dr. Chanintorn Jittawiriyanukoon
Academic Year	2/2002

The Department of Telecommunications Science, Faculty of Science and Technology of Assumption University has approved this final report of the **twelve** credits course. **TS7000 Master Thesis**, submitted in partial fulfillment of the requirements for the degree of Master of Science in Telecommunications Science.

Approval Committee:

(Asst.Prof.Dr. Chanintorn J. Nukoon)
Advisor

(Asst.Prof.Dr. Dobri Batovski)
Committee Member

(Dr. Sergey Fedosseeve)
Committee Member

(Asst.Prof.Dr. Surapong Auwatanamongkol)
Representative of Ministry of
University Affairs

Faculty Approval:

(Asst.Prof.Dr. Chanintorn J. Nukoon)
Program Director

(Asst.Prof. Supavadee Nontakao)
Dean

February / 2003

Acknowledgements

The writer desires to express his most sincere appreciation and thanks to Assistant Professor Chanintorn Jittawiriyanukoon for his encouragement and advice through the course of this study and for his help in the preparation and writing of this thesis.

Thanks go to the staff and customer of Fresh Answer co. Ltd. for helping evaluate the results and conducting the subjective tests.

Thanks are also due to Ms. Myo Han for reading and checking the format of this thesis.

Abstract

Voice over IP (VoIP), the transmission of real-time voice over an Internet Protocol (IP) network, is gaining much momentum in the industry. Transmitting voice over an IP network has many benefits over transmitting voice over conventional telephone systems. These benefits are what have motivated the industry to develop VoIP systems, but the impairments present on an IP network make implementing such a system quite a challenge.

In this thesis, Voice over IP (VoIP) will be studied with a particular emphasis on the voice compression algorithms used in such a system. The desired characteristics of a voice compression algorithm for VoIP will be analyzed. Theses characteristics will then be used to judge how appropriate are some existing standards of voice compression algorithms for VoIP. An evaluation of voice quality of some existing voice compression standards will be done and the result will be compared with subjective tests.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Motivation

Voice over IP (VoIP) is a transmission of real-time voice over an Internet Protocol (IP) network. The IP network that is used to transmit the voice can either be a private network or the public Internet. It is an alternate way to make a telephone call as compared to the Public Switched Telephone Network (PSTN). While VoIP has its origins as a cheap way to make a long distance call over the Internet, it is much more than that today. The first paper and experiments on the subject occurred in the early 1970's, but it took until the 1990's before the concept was taken more seriously [1]. Industry is now quickly recognizing the potential of VoIP and many VoIP applications have been developed in recent years. Even though VoIP is being recognized in the industry, not much detailed research has been done on the subject. This is one of the motivations for proposing this thesis.

Why would we transmit voice over an IP network if we are already able to transmit voice over the conventional telephone network? The answer is simple. Transmitting voice over an IP network has many benefits over transmitting voice over PSTN. These benefits are what have motivated the industry to develop VoIP systems and can be summarized as:

- it offers free or cheaper calls as compared to PSTN
- it permits the integration of voice and data networks while recognizing the reality that data is the dominant traffic
- it permits the use of state of the art speech compression algorithms and silence suppression to reduce the necessary bandwidth for a single call

- it permits the easier addition of additional features to telephony and a better user interface
- it permits corporations to make better use of their investments in data networks

These benefits are so great that they warrant using VoIP as an alternative or even possibly a replacement to the conventional telephone network.

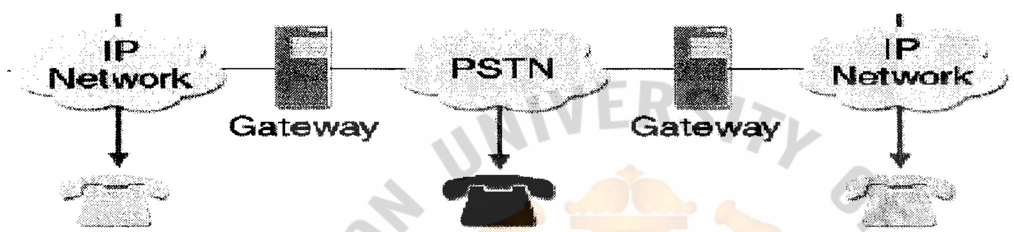

Figure 1.1: Sample implementation for VoIP

There is a great interest in supporting voice applications over both the public Internet and private intra-nets, i.e., Voice over IP (VoIP). Several popular Internet implementations are the Video Audio Tool (VAT) [2] and the Robust Audio Tool (RAT) [3], as well as a host of ITU-T H.323 implementations. An important aspect of VoIP is developing a performance monitoring capability to track the quality of the voice transport. The main impairments that exist on an IP network that are disturbing to VoIP application are delay, jitter and packet loss. Examining the techniques to deal with these impairments and trying to improve on them is also a main motivation for proposing this thesis.

Voice compression is an essential part of any VoIP system. It determines how we encode the speech for transmission and has an enormous effect on the speech quality observed. One of the main motivations to propose this thesis was to examine and evaluate the speech compression algorithms and determine which type of speech compression algorithms are better suited for VoIP.

1.2 Objectives

The objectives of this thesis are the following:

- Provide a complete literature review of VoIP and related subjects (speech compression and IP networks). This is necessary to properly understand the challenges that exist in designing a VoIP system.
- Provide a comparative study on speech compression algorithms for VoIP. It is desirable to identify clearly what are the most important characteristics for a speech compression algorithm to be used in VoIP. Then we can use these characteristics to compare how appropriate the different standards for speech compression are for VoIP.
- Investigate, examine and evaluate different speech compression algorithms and determine which types of speech compression algorithms are better suited for VoIP; this represents an important contribution for this thesis.
- Propose a method of monitoring the performance of speech compression in VoIP

1.3 Thesis Layout

The five remaining chapters will be divided as follows:

Chapter 2: Voice over IP – Review and description of typical VoIP system.

Chapter 3: Voice Compression – Different methods of performing speech compression: waveform-based, perceptual-based and model-based.

Chapter 4: Voice Compression for VoIP– Speech compression algorithms in the context of VoIP.

Chapter 5: Performance Evaluation of Voice Compression for VoIP– Evaluation of the different type of voice compression.

Chapter 6: Discussion and Conclusions.

Chapter 2: Voice over IP

2.1 Background of VoIP

The Internet telephony systems are composed of these elements [4]:

- end devices: these may be either traditional telephones (analog/GSM/ISDN/...), audio-equipped personal computers, or single use appliances
- gateways: if a traditional telephone is used at either calling side the call (i.e. its transmission format, signaling procedures, audio codecs) has to be translated to/from the format for transport over the Internet; this is the task of the gateways
- Gatekeepers/proxies: the gatekeepers/proxies provide centralized call management functions; they may provide call admission control, bandwidth management, address translation, authentication, user location, and so on.
- multipoint conference units: these manage multiparty conferences

The components may be implemented as hardware or software and may be integrated into single units optionally.

They communicate with each other over signaling and voice-transporting **protocols**. To ensure interoperability between products of different vendors, standardization bodies have elaborated standards for both classes of protocols. Between them are The Internet Engineering Task Force (IETF), The International Telecommunication Union (ITU), and the European Telecommunications Standards Institute (ETSI).

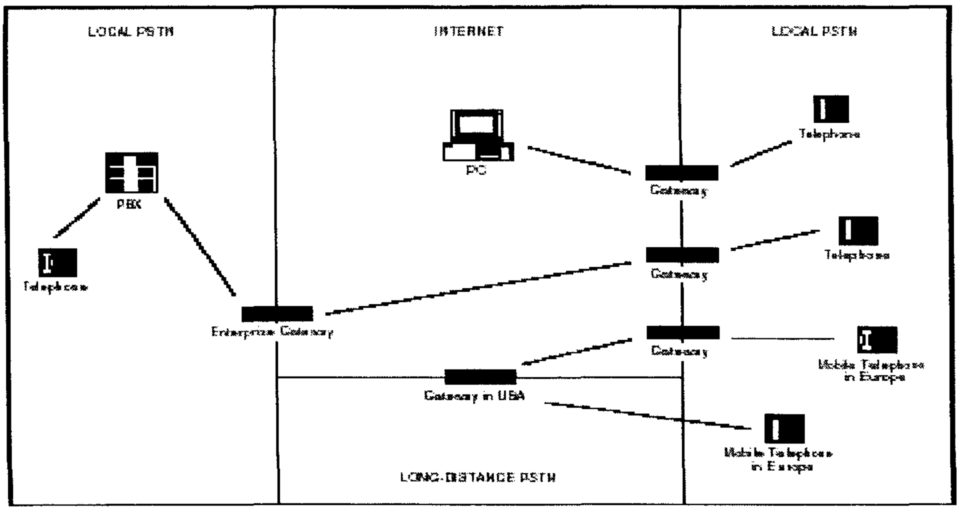

Figure 2.1: illustration of VoIP components

As I mentioned before, the best delivery service offered by the Internet results in highly variable packet delays, loss, and jitter [6, 7]. The packet loss probabilities and packet delays are often beyond what is considered acceptable for good speech quality. The International Telecommunications Union (ITU) has recommended one-way delays no greater than 150 ms for most applications [8], with a limit of 400 ms for acceptable voice communication. Tolerable loss rates depend heavily on the speech codec in use [9], and can range from 0 to 10% [10]. The implication is that the best effort Internet will not always be sufficient for high quality voice.

There are two approaches that can be taken to combat this problem. The first is to provide improved network layer performance. The Internet Engineering Task Force (IETF) has proposed the Integrated Services architecture [11, 12] as one approach to the problem. *Intserv*, and its companion signaling protocol, the Resource Reservation Protocol (RSVP) [13, 14], allows hosts to request end-to-end QoS. Using the guaranteed service model, they can request a bounded delay with zero loss. The controlled load model allows hosts to request service identical to an unloaded network, without specific numerical guarantees. However, scaling concerns (among other difficulties) have led the IETF to consider a more lightweight approach to network QoS, called differentiated services [15, 16, 17, 18].

The second approach for reducing loss and delay is through end-to-end adaptive mechanisms.

In this case, end systems measure the service being delivered by the network (using Real Time Control Protocol - RTCP [5]), and send additional information, and/or run additional algorithms, to improve voice quality.

These mechanisms do not rely on explicit support from the network beyond normal packet transport. It is for this reason they are considered end-to-end mechanisms.

Ideally, a well engineered, QoS-aware network would obviate the need for end-to-end adaptation. However, the heterogeneous nature of the Internet leads us to conclude that it is unlikely for any solution to be ubiquitously deployed any time soon. As such, end-to-end adaptive mechanisms are, and will remain, critical for high quality voice.

2.2 IP Networks

2.2.1 Protocols for VoIP

The layering of the IP protocols for VoIP system commonly called a protocol stack for VoIP is illustrated in Figure 2.2.

The first and the most basic protocol on the protocol stack is the *physical layer protocol*. This protocol is responsible for interacting with the physical medium. The protocol defines what electrical signal must be used to represent each bit on the wire and at what physical rate transmission should be accomplished. Since such a protocol differs from every different physical medium (fiber optics, twisted copper pairs, etc.)

it can conveniently be changed without affecting any of the other higher level protocol.

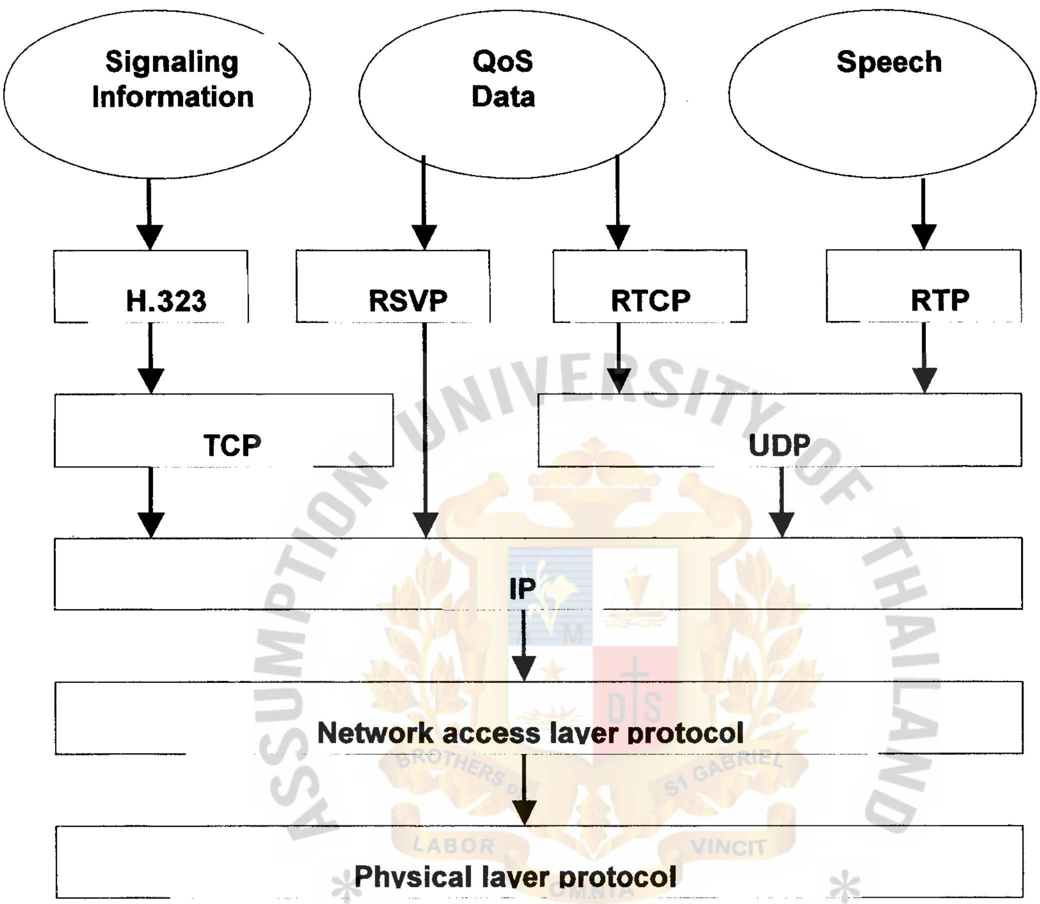

Figure 2.2: Typical VoIP Protocol Stack

The next protocol is the *network access layer protocol*, which is also called Medium Access Control (MAC). It is necessary to ensure a proper exchange of information within one homogeneous network. This protocol varies with each network technology. This protocol requires adding additional information to every packet that is sent on the network in the form of a header. Information included in this header usually includes the destination of the message and requests to use network facilities such as priority [19].

The next higher level protocol is the *Internet Protocol* (IP). This level of protocol is required to provide internetworking i.e. the communication between different networks. In order to provide the routers enough information to direct the packets properly to their destination, information must be added to the user data in the form of an IP header. The content of the IP header has a minimum size of 20 bytes [20].

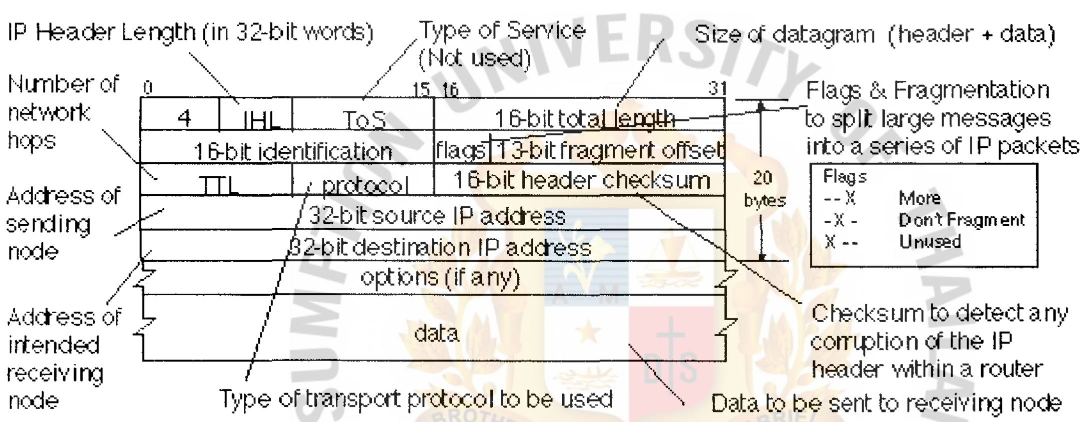

Figure 2.3: IP header fields.

The header fields are discussed below:

- **Version** (always set to the value 4, which is the current version of IP)
- **IP Header Length** (number of 32 -bit words forming the header, usually five)
- **Type of Service**, now known as **Differentiated Services Code Point (DSCP)** (usually set to 0, but may indicate particular Quality of Service needs from the network, the DSCP defines one of a set of class of service)
- **Size of Datagram** (in bytes, this is the combined length of the header and the data)

- **Identification** (16-bit number which together with the source address uniquely identifies this packet - used during reassembly of fragmented datagrams)
- **Flags** (a sequence of three flags (one of the 4 bits is unused) used to control whether routers are allowed to fragment a packet (i.e. the Don't Fragment, DF, flag), and to indicate the parts of a packet to the receiver)
- **Fragmentation Offset** (a byte count from the start of the original sent packet, set by any router which performs IP router fragmentation)
- **Time To Live** (Number of hops /links which the packet may be routed over, decremented by most routers - used to prevent accidental routing loops)
- **Protocol** (Service Access Point (SAP) which indicates the type of transport packet being carried (e.g. 1 = ICMP; 2= IGMP; 6 = TCP; 17= UDP).
- **Header Checksum** (A 2's complement checksum inserted by the sender and updated whenever the packet header is modified by a router - Used to detect processing errors introduced into the packet inside a router or bridge where the packet is not protected by a link layer cyclic redundancy check. Packets with an invalid checksum are discarded by all nodes in an IP network)
- **Source Address** (the IP address of the original sender of the packet)
- **Destination Address** (the IP address of the final destination of the packet)
- **Options** (not normally used, but when used the IP header length will be > 5 32-bit words to indicate the size of the options field)

The next higher level in the protocol stack is the *transport layer*. This layer provides end-to-end data service making sure that the data is received by the proper application in the destination computer. Two different versions of such a protocol are used for VoIP: User Datagram Protocol (UDP) and Transmission Control Protocol (TCP).

The service provided by UDP is an unreliable service which provides no guarantees for delivery and no protection from duplication (if this arises due to software errors within an Intermediate System (IS)). The simplicity of UDP reduces the overhead from using the protocol and the services may be adequate in many cases.

A computer may send UDP packets without first establishing a connection to the recipient. The computer completes the appropriate fields in the UDP header (PCI) and forwards the data together with the header for transmission by the IP network layer.

As opposed to UDP, The Transmission Control Protocol (TCP) is a connection-oriented reliable protocol. It provides a reliable transport service between pairs of processes executing on End Systems (ES) using the network layer service provided by the IP protocol. TCP is stream oriented, that is, TCP users exchange streams of data. The data are placed in buffers and transmitted by TCP in transport Protocol Data Units (sometimes known as "segments"). TCP is much more complex than UDP.

Four different higher level protocols are used in VoIP: *h.323, RSVP, RTCP and RTP.* The Resource Reservation Protocol (RSVP) is a protocol that implemented directly over IP providing a standardized method for time sensitive applications to reserve bandwidth through request to the router on the IP networks. A host uses RSVP to request a specific *Quality of Service* (QoS) from the network, on behalf of an application data stream. RSVP carries the request through the network, visiting each node the network uses to carry the stream. At each node, RSVP attempts to make a resource reservation for the stream [21].

A primary feature of RSVP is its scalability. RSVP scales to very large multicast groups because it uses receiver-oriented reservation requests that merge as they progress up the multicast tree. The reservation for a single receiver does not need to travel to the source of a multicast tree; rather it travels only until it reaches a reserved branch of the tree. While the RSVP protocol is designed specifically for multicast applications, it may also make unicast reservations.

Among H.323, RTCP and RTP, only H.323 is implemented over TCP while the two others work over UDP. The Real Time Protocol (RTP) is used to transmit voice over UDP. Voice does not tolerate high delay; this is the reason for selecting UDP over TCP. RTP is used as an added layer instead of using UDP directly because of a real-time nature of voice transmission. RTP adds a time stamp and sequence number to the voice data in order to permit the proper ordering of the voice data before playback. As well, RTP permits multiple users to participate in the same VoIP call. In order to provide such service, RTP needs to add a header of minimum size of 12 bytes to the speech data.

A protocol that is associated with RTP is the Real Time Control Protocol (RTCP). It is essentially a control protocol for RTP [22]:

- Permits receiving feedback from all participants on the quality of the data received such as delay and packet loss number.
- Provides a canonical name, a transport level identifier for each source.

- Permits each participant to be aware of all of the participants in the conversation.
- Can optionally be used to transfer information about each user such as their names and telephone numbers.

A Final protocol implemented over the transport protocol is the H.323 protocol. This is an ITU standard that enables multimedia calls to be established between two or more parties on packet network. The main contribution of H.323 is to permit the proper signaling and resources to establish and terminate a call. It is also responsible for finding the proper IP address of a given destination and for negotiating between all end users a proper format of transmission that is supported by all (essentially choosing a compression standard that is recognized by all).

H.323 standard can accommodate any multimedia calls (video), but voice only subnet of the standard is the one used for VoIP. H.323 is the most popular and most widely accepted standard to perform signaling for VoIP, but it is not the only existing standard. The Session Initiation Protocol (SIP) performs similar functions as H.323 and is a recommendation from the Internet Engineering Task Force (IETF) but is not as widely used.

2.2.2 Quality of Service

Quality of Service (QoS) refers to a guarantee on a certain minimal quality of the service done over the network [23]. Due to its origin as a network for exchanging computer data, the IP network does not guarantee any QoS to real-time applications such as VoIP. VoIP necessitates a small amount of delay that is constant for each packet that is transmitted on the network. Also, it requires that all packets of voice reach their destination properly at the first transmission since TCP cannot be used to recover from lost packets due to delay constrains.

Up to this date, IP networks are not able to guarantee that the voice packets will not be lost due to congestion or that the data will not be corrupted due to noise on the transmission line. Since QoS is not guaranteed by the IP network and that efforts to add QoS to the IP network have so far not been sufficient, the only alternative for VoIP is to accept these limitations of the IP network. A VoIP system must implement mechanisms to deal with these shortcomings of the IP network in order to have an acceptable level of service.

2.3 VoIP Components

Figure 2.4 shows the different components required to perform VoIP at the encoder and the decoder for typical VoIP system.

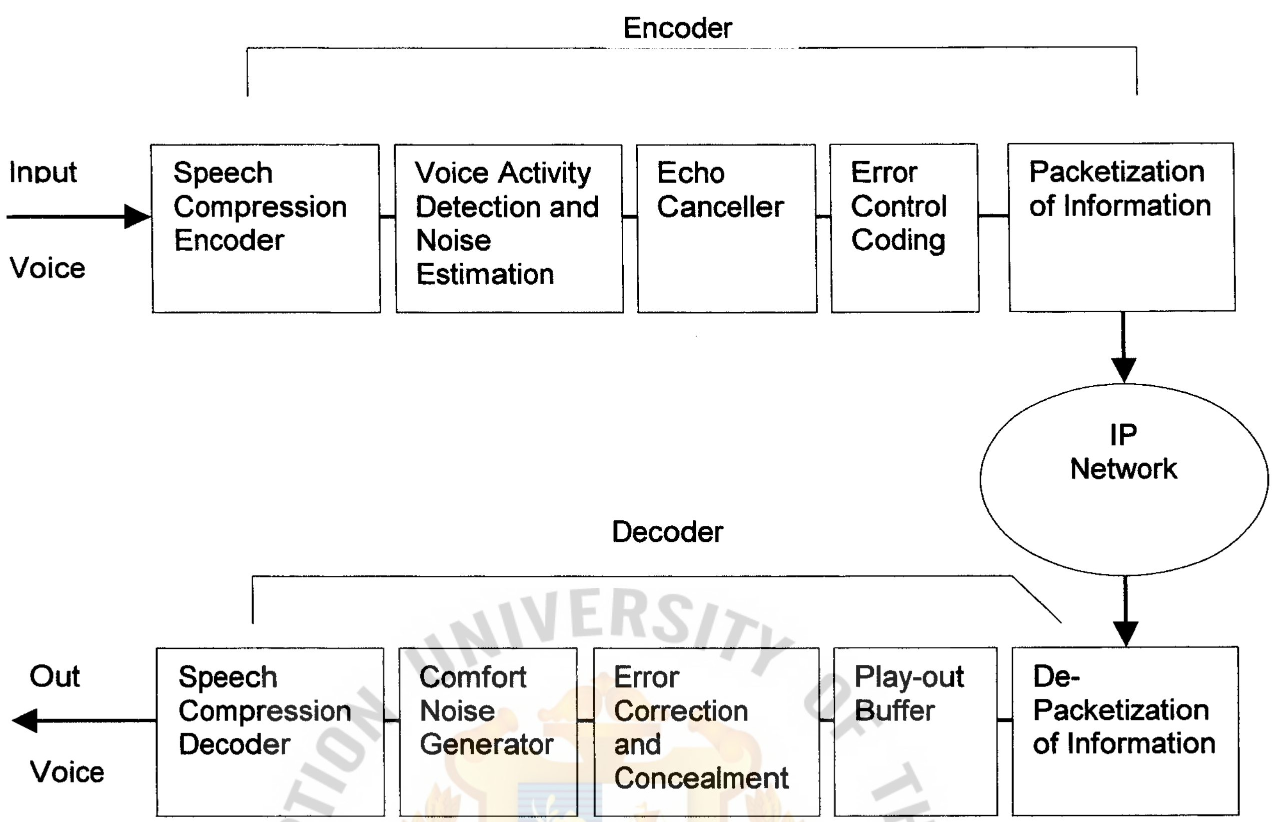

Figure 2.4: Typical VoIP Components.

The task illustrated in this figure only deal with the transmission of the voice across the IP Network. In VoIP system, signaling must also be preformed in order to establish the telephone calls. The tasks required to perform the signaling and also to perform any billing will not be discussed any further in this thesis.

A central component of VoIP system is the voice compression algorithm. It is responsible for coding the speech for a VoIP call in an efficient manner. Both parties of the VoIP call must be equipped with voice compression encoder and voice compression decoder. Note that even though the choice of specific voice compression algorithm to be used is flexible in a VoIP system, it is essential that both sides use the same algorithm in order to assure compatibility. Since voice compression is central to this thesis and to VoIP, it will be presented in more detail in chapter 3.

The Voice Activity Detection and Noise Estimation block is an optional block that is usually used in VoIP. it permits the system to save bandwidth by transmitting less data during silent periods of the speech.

The Echo canceller show here serves the same purpose as ones found on the PSTN. It removes the echo caused by the hybrid when the voice travels on the regular PSTN.

As is often used in telecommunications, Error Control Coding can be used to protect the compressed voice against bit error. At the decoder, error correction can then be performed if permitted. Given the unique nature of error in VoIP (whole packets can be lost), the error control coding and the error concealment will differ from what is usually encountered in regular telephony.

The Play-out Buffer is a component that is present to remove the jitter from the system. Jitter is one of the important impairments in VoIP.

Chapter 3: Voice Compression

3.1 Characteristics of the Voice Signal

A voice signal in our sense is sound formed by humans in order to communicate. Like all other sound, it is a perturbation of the atmospheric pressure that travels in the form of a sound wave. This sound wave can propagate through the air and is perceptible when it enters our ear.

To better understand the properties of human voice, an examination of how it is formed is in order. The vocal tract is responsible for forming voice. It is basically an acoustic tube with a non-uniform cross-section consisting of two parts: the oral tract (from lips to the vocal cords) and the nasal tract (from the valum to the nostrils). Note that the valum is a small valve that opens and closes to allow the connection of the oral tract with the nasal tracts. In some analysis, the voice signal is broken down into its smallest portions, which we call phonemes.

To use voice in communication over networks, we need a way to transform the speech signal from a sound wave to an electrical signal. This is usually done with a microphone and the result is an electrical signal where the voltage varies with the pressure of the speech signal. Like all other natural signal, the voice signal is originally an analog signal. This means that the electrical signal that we obtain using microphone is also analog. It is usually preferred to use a digital version of the speech since digital transmission is more efficient and reliable than analog transmission [24]. Voice can be represented digitally in different ways using voice encoding algorithms. If the number of bits required to represent the digital signal is reduced further, this is called voice compression. The most popular families for voice compression will be discussed in the remaining of this chapter.

3.2 Waveform-based Compression

3.2.1 Overview

Waveform-based voice compression is the simplest family of voice compression algorithms. Members of this family include pulse code modulation (PCM), differential pulse code modulation (DPCM) and adaptive differential pulse code modulation (ADPCM). These algorithms are classified as waveform-based compression algorithms because they code the voice waveform directly by representing the analog voice waveform in a digital form as faithfully as possible.

3.2.2 PCM

Pulse Code Modulation is the standard way to encode voice signal into a digital voice signal. Even though it requires a high bit rate, it is still widely used in telephony because of its simplicity.

The signals in PCM are binary; that is, there are only two possible states, represented by logic 1 (high) and logic 0 (low). This is true no matter how complex the analog waveform happens to be. Using PCM, it is possible to digitize all forms of analog data, including full-motion video, voices, music, telemetry, and virtual reality (VR).

To obtain PCM from an analog waveform at the source (transmitter end) of a communications circuit, the analog signal amplitude is sampled (measured) at regular time intervals. The sampling rate, or number of samples per second, is several times the maximum frequency of the analog waveform in cycles per second or hertz. The instantaneous amplitude of the analog signal at each sampling is rounded off to the nearest of several specific, predetermined levels. This process is called quantization. The output of a pulse code modulator is thus a series of binary numbers, each represented by some power of 2 bits.

At the destination (receiver end) of the communications circuit, a pulse code demodulator converts the binary numbers back into pulses having the same quantum levels as those in the modulator. These pulses are further processed to restore the original analog waveform [25].

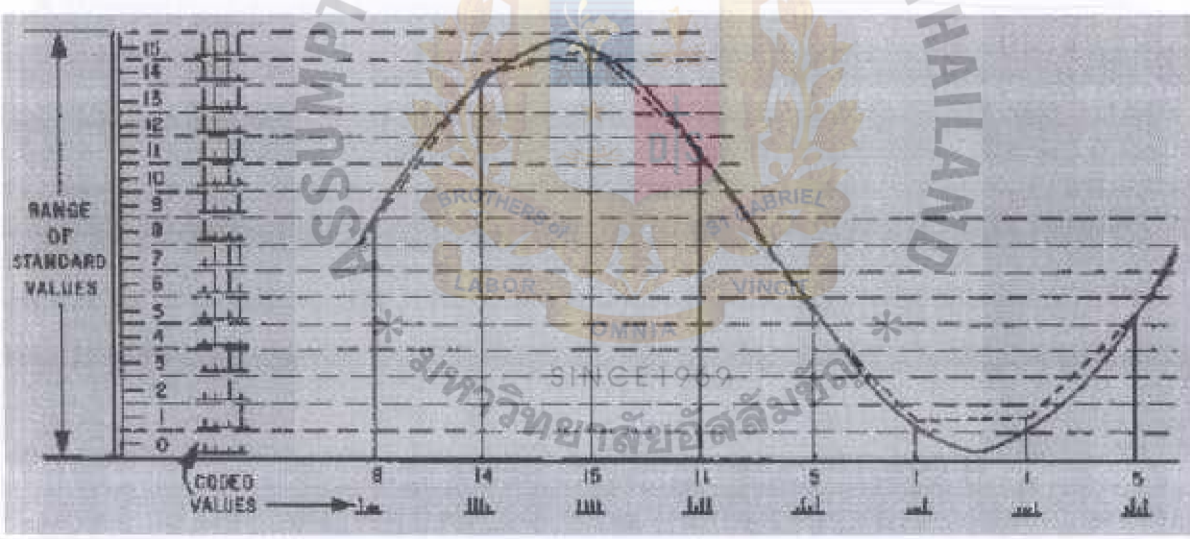

Figure 3.1: Pulse-code modulation of a quantized wave (32 bits).

The International standard for PCM is ITU-T G.711. This is the standard that is supported in conventional telephony. It uses logarithmic scalar quantization and represents each voice sample with 8 bits. The analog speech is first band-limited to the typical telephone bandwidth of 3.4 kHz and is sampled at 8 kHz. This result in a total bit rate of 64 kb/s It is widely used due to its simplicity and very low delay. The only drawback of PCM is that it requires a high bit rate.

Figure 3.2: Block Diagram of Pulse-code modulation

3.3 Perceptual-based Compression

3.3.1 Overview

The main idea behind perceptual-based voice compression is not to code the part of the voice that is inaudible. The human ear is not capable of hearing all sounds since some sounds do not meet the minimum threshold of audibility of the ear while others are masked (made not hearable) by another louder sound of similar frequency [26].

In perceptual voice compression, a perceptual model permits the encoder to distinguish the audible elements of the voice. By only encoding what is audible, a significant reduction in bit rate can be achieved with no loss of quality.

3.2.2 International Standards

All standards for perceptual compression were developed for audio compression. This permits encoding of CD quality voice at only 64 kb/s. The compression obtained by the perceptual modeling permits us to encode voice of higher quality than PCM at the same or lower bit rate.

Some international standards for perceptual compression are MPEG-1 Layer 3 (commonly known as MP3), Dolby AC-2 and AC-3, the Perceptual Audio Coder (PAC), and MPEG-2 Advanced Audio Coding (AAC). Characteristics ot these international standards are highlighted in Table 3.1.

Standard	**Sample Rate (kHz)**	**Bit Rate (kbps)**	**Year finalized**
Dolby AC-2	44.1	256/channel	1984
Dolby AC-3	44.1	32-384	1994
Perceptual Audio Coder (PAC)	44.1	128/stereo channel	1995
MPEG-1	32-48	32-448	1992
MPEG-2	16-48	16-448	1994
MPEG-2 AAC	8-96	Variable, max 48	1997

Table 3.1: International Standard for Perceptual-based voice compression.

3.2.3 Perceptual Model

What makes perceptual encoding effective as a method of audio data compression is its deviation from the PCM model. As we have seen, in a PCM system the goal is to digitally reproduce the waveform of an incoming signal as accurately as is practically possible. However, it could be argued that the implicit assumption of PCM — namely that the reproduction of sound requires the reproduction of waveforms — is simplistic, and involves a misunderstanding of the way human perception actually works.

The fact of the matter is that our ears and our brains are imperfect and biased measuring devices, which interpret external phenomena according to their own prejudices. It has been found, for example, that a doubling in the amplitude of a sound wave does not necessarily correspond with a doubling in the apparent loudness of the sound. A number of factors (such as the frequency content of the sound, and the presence of any background noise) will affect how the external stimulus comes to be interpreted by the human senses. Our perceptions therefore do not exactly mirror events in the outside world, but rather reflect and accentuate certain properties of those events.

We might therefore decide, as our goal is to reproduce a sound for the benefit of a human listener, that it is quite unnecessary to accurately recreate every characteristic of that sound's waveform. Instead we might concentrate on determining which properties of the waveform would be most important to the listener, and prioritize the recording of these properties. This is the theory behind 'perceptual coding'. To put it more simply, we might say that whilst PCM attempts to capture a waveform 'as it is', perceptual model attempts to capture it 'as it sounds'.

In order for this to be possible, a certain set of judgments as to what is or isn't meaningful to a human listener has had to be determined. This set of judgments is sometimes called a 'psychoacoustic model'. In order to understand how the psychoacoustic model works, we need to consider two important concepts in digital audio and perceptual coding: 'redundancy' and 'irrelevancy'.

Both words describe grounds on which a certain amount of audio data is deemed to be unnecessary, and sufficiently unimportant that it can be discarded or ignored without an unacceptable degradation in sound quality. CD-quality PCM audio discards frequencies higher than 22.05 kHz — the sampling frequency of 44.1 kHz can be chosen because frequencies about 22.05 kHz were deemed to be beyond the range of human hearing, and therefore redundant. Of course, if we were to decide (as some audiophiles have) that frequencies above 22.05 kHz actually do carry important information about the color and tone of sound and music, we might choose to use an increased sampling frequency, thereby capturing some of the frequencies the CD-quality system would have treated as redundant. Even if we were to do so, however, we would not have done away with redundancy altogether: we would simply have moved the goalposts (or, more accurately, the 'Nyquist Limit') so that redundancy occurred at higher frequencies than before. Redundancy, in other words, is not new as far as digital audio is concerned: it is in fact an inevitable fact of digital life.

Irrelevancy, however, is a rather more radical concept. The theory behind psychoacoustic coding argues that, because of the peculiarities of human perception, certain properties of any given waveform will be effectively meaningless to a human listener — and thus will not be perceived at all. However, because of its insistence on capturing the entire waveform, a PCM system will end up recording and storing a large amount of this irrelevant data, in spite of its imperceptibility on playback. Perceptual coding aims, by referring to a psychoacoustic model, to store only that data which is detectable by the human ear. In so doing it is possible to achieve drastically reduced file sizes, by simply discarding the imperceptible and thus irrelevant data captured in a PCM recording.

3.2.3.1 Masking

The psychoacoustic model depends upon a particular peculiarity of human auditory perception: an effect known as masking. Masking could be described as a tendency in the listener to prioritize certain sounds ahead of others, according to the context in which they occur. Masking occurs because human hearing is adaptive, and adjusts to suit the prevailing levels of sound and noise in a given environment.

Masking is what enables perceptual coding to get away with removing much of the data that conventional waveform coding would store. This does not entail discarding all of the data describing masked elements in a sound recording: to do so would probably sound bizarre and unpleasant. Instead, perceptual coding works by assigning fewer bits of data to the masked elements of a recording than to the 'relevant' ones. This has the effect of introducing some distortion, but as this distortion is (hopefully) confined to the masked elements, it will (hopefully) be imperceptible on playback. Using fewer bits to represent the masked elements in a recording means that fewer bits overall are required.

3.4 Model-based Compression

3.4.1 Overview

Model-based voice compression algorithms use a model of human voice production to extract from the voice parameters for this model. Considerable compression can be obtained by transmitting these parameters in place of the actual voice.

There are two main algorithms: Vocoders and Code Excited Linear Prediction (CELP). One of the main differences is that CELP does not make a decision on each segment to determine if it is voiced or unvoiced. Instead, each voice segment is considered to be a mixture of voiced and unvoiced voice. For this reason, the excitation is always formed from the sum of the voiced voice excitation (impulse train) and the unvoiced voice excitation (random noise). Both have their own gain to control the level of the voice. This provides considerable improvement of the voice quality given that many voice segments are not totally voiced or unvoiced but a mixture of the two. This occurs quite often in the transition period between a voiced voice and unvoiced voice. By always representing voice as a mixture of voiced and unvoiced voice, better quality can be obtained in such condition.

3.4.2 International Standards

Many different international standards exist for model-based voice compression. Most of these have been made popular by their use for cellular telephony or other specialized applications. A list of the most prominent standards is given in Table 3.2 with their associated bit rates; algorithmic delay and the year were finalized.

Standard	Bit Rate (kbps)	Algorithmic delay (ms)	Year finalized
ITU-T G.728 LD-CELP	16	0.625	1994
ITU-T G.729 CS-ACELP	8	15	1995
ITU-T G.723.1 MPC-MLQ	5.3 or 6.4	37.5	1995
ITU-T G.729A CS-ACELP	8	15	1995
GSM RPE-LTP	13	20	1987
FS-1015 LPC-10E	2.4	111.5	1984
FS-1016 CELP	4.8	37.5	1991

Table 3.2: International Standard for Model-based voice compression.

All recent entries in this table (after 1990) are CELP algorithms. This shows how popular these are for model-based speech compression. Among the algorithms in the table, the currently most popular voice compression algorithms for VoIP are ITU-T G.723.1, ITU-T G.729 and ITU-T G.729A.

3.4.3 ITU G.723.1

This coder is designed to operate with a digital signal obtained by first performing telephone bandwidth filtering (Recommendation G.712) of the analogue input, then sampling at 8000 Hz and then converting to 16-bit linear PCM for the input to the encoder. The output of the decoder should be converted back to analogue by similar means. Other input/output characteristics, such as those specified by recommendation G.711 for 64 kbit/s PCM data, should be converted to 16-bit linear PCM before encoding or from 16-bit linear PCM to the appropriate format after decoding.

The coder is based on the principles of linear prediction analysis-by-synthesis coding and attempts to minimize a perceptually weighted error signal. The encoder operates on blocks (frames) of 240 samples each. That is equal to 30 ms at an 8 kHz sampling rate. Each block is first high pass filtered to remove the DC component and then divided into four sub-frames of 60 samples each. For every sub-frame, a 10th order Linear Prediction Coder (LPC) filter is computed using the unprocessed input signal. The LPC filter for the last sub-frame is quantized using a Predictive Split Vector Quantizer (PSVQ). The unquantized LPC coefficients are used to construct the short-term perceptual weighting filter, which is used to filter the entire frame and to obtain the perceptually weighted speech signal [27].

3.4.4 ITU G.729A

G.729 is CELP compression where voice is coded into 8-kbps streams. There are two variations of this standard (G.729 and G.729 Annex A) that differ mainly in computational complexity; both provide speech quality similar to 32-kbps ADPCM.

The CS-ACELP coder is based on the Code-Excited Linear-Prediction (CELP) coding model. The coder operates on speech frames of 10 ms corresponding to 80 samples at a sampling rate of 8000 samples per second. For every 10 ms frame, the speech signal is analyzed to extract the parameters of the CELP model (linear-prediction filter coefficients, adaptive and fixed-codebook indices and gains). These parameters are encoded and transmitted [28].

The G.729A algorithm has a standardized way to deal with the eventuality of packet losses. Whenever the decoder determines that the packet has been lost, the following steps are initiated to conceal the error:

- Linear Predictor Coefficient (LPC) are set to their respective values from the last properly received frame.
- Codebook gains are attenuated from the previously received values.
- For the adaptive vector v(n), use the integer part of the pitch delay from previously received good frame.
- The fixed codebook vector c(n) is chosen randomly

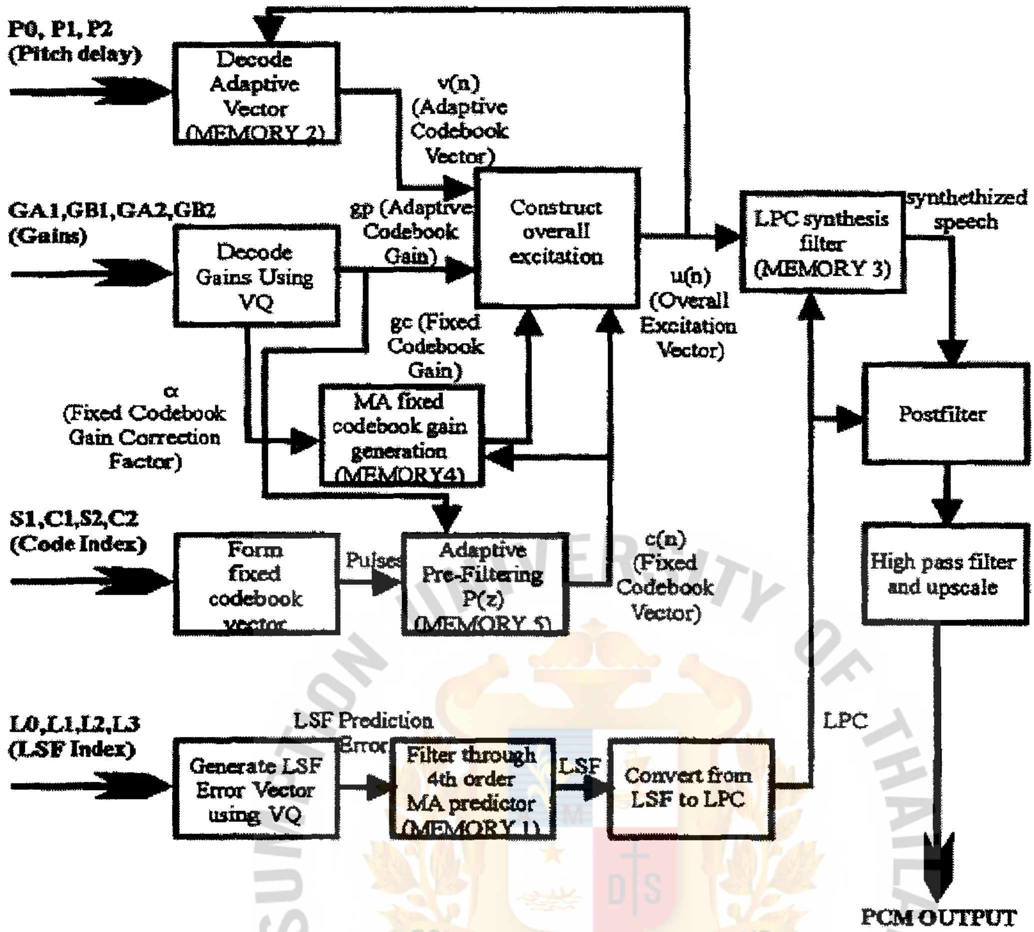

Figure 3.3: Block Diagram of G.729A Decoder

Chapter 4: Voice Compression for VoIP

4.1 Desired Characteristics of voice compression for VoIP

Before we choose an algorithm, we must determine the algorithm's desired characteristics that we are looking for. In our case, we need to select a voice compression algorithm for VoIP.

As mentioned earlier, a good VoIP voice compression algorithm has to have: low delay, good recovery from packet loss, good voice quality, low complexity and low bandwidth [29].

4.1.1 Low Delay

People generally cannot tolerate high delay in telephony system since it prevents them from communicating efficiently. Since we have seen that delay in a typical VoIP system is quite significant, it is important to keep the delay introduced by speech compression algorithm as low as possible. While other sources of delay such as the IP network are out of control, the delay introduced by the voice compression algorithm can be kept to a minimum by a good choice of the algorithm.

4.1.2 Good Recovery from Packet Loss

Packet loss is a major impairment that is unique to VoIP systems. Since most voice compression algorithms where not designed to deal with packet loss, it is important to evaluate how well such algorithms perform when we introduce a lost packet recovery mechanism.

4.1.3 Good Voice Quality

The most obvious requirement for voice compression algorithm to be used for VoIP is that it delivers a good voice quality. Since most people are used to the good toll quality voice delivered by the PSTN, that they will not be satisfied with any voice does not measure up to this quality. Toll quality voice is the minimum voice quality that is acceptable for VoIP and anything beyond that would be a real bonus.

4.1.4 Low Complexity

Even though we have CPUs and DSP chips with very high processing power, low complexity still remains a desired characteristic. The reason is that most powerful CPUs and DSP chips cost more money.

4.1.5 Low Bandwidth

Even though the bandwidth available on current telephone and data networks keeps on increasing, using less bandwidth is still a desired property of voice compression algorithm for VoIP. The reason here again is money since an algorithm that consumes less bandwidth permits us to multiplex more lines on the same wire; This results in better utilization of the existing resources and more profits to the telecommunication companies.

4.2 Advantages and Disadvantages of Different Voice Compression Families

4.2.1 Overview

Having reviewed what the desired characteristics for voice compression algorithm in VoIP system are, we will now see how the different families of voice compression fare with these characteristics. The families are: waveform-base, perceptual-based and model-based voice compression. The result of this comparison is highlighted in Table 4.1.

Characteristic	**Waveform-based**	**Perceptual-based**	**Model-based**
Delay	Very low	High	Low
Packet loss recovery	Fair	Fair	Very good
Voice Quality	Toll quality or better	Toll quality or better	Toll quality (CELP), fair (vocoder)
Complexity	Low	Medium	High
Bandwidth Required	High (PCM), Medium (ADPCM)	Medium	Low
Other	No memory problems	Permit to use CD quality voice	Does not reproduce telephone tones and music well, problem of tandem connections

Table 4.1: Comparison for Different Voice Compression Families.

4.2.2 Waveform-based Voice Compression

The main advantage of using waveform-based compression such as PCM is that it is very simple and produces good voice quality. The simplicity translates into an algorithm of low complexity that can be run easily on a CPU and does not require much memory. As well, the simplicity translates into negligible delay since each sample is ready to be transmitted as soon as it is sampled and quantized. The voice quality of PCM defines what we expect out of toll-quality since it is the standard what we use currently for telephony. All other members of the waveform-based voice compression family also offer similar voice quality. A final advantage of PCM in particular is that it does not necessitate any memory. Since each sample is encoded independently, a bit error will only affect the particular voice sample. As well, the packet loss will cause degradation only to voice samples that were lost. This differs from algorithms with memory where packet loss will degrade the voice quality for a longer duration due to corrupted memory.

The main disadvantage of waveform-based voice compression is its high bandwidth requirement. As seen before, PCM requires a bit rate of 64 kb/s, which is quite high as compared to other voice compression families.

4.2.3 Perceptual-based Voice Compression

The main advantage of using a perceptual-based voice compression algorithm is that it produces high quality voice. Since the algorithms use a perceptual model, the different between the original voice and compressed voice is generally not audible. Another advantage is that these perceptual compression algorithms can easily encode music or sounds and can provide CD-quality voice. One final advantage is that the variable bit rate of the algorithms permits an easy way to implement silence suppression. Since each frame is always encoded with the number of bits that are required to faithfully reproduce the sound, periods of silence can be encoded with very few bits.

The main disadvantage of using perceptual-based voice compression algorithms for VoIP is the high delay that they require. Since good perceptual compression requires big frame sizes, it is not rare for these algorithms to require over 100 ms of delay. Another disadvantage is that they are relatively complex. A final disadvantage is that the compression obtained for the telephone bandwidth voice is not that considerable. Since most of the masking occurs in the higher frequency, not much compression is obtained when we consider only the frequencies up to 3.4 kHz.

4.2.4 Model-based Voice Compression

The main advantage of model-based voice compression is their low bandwidth requirement. This is by far the family of voice compression algorithms that obtain the highest compression ratio as compared to PCM. Another advantage is a low delay requirement. Most of these algorithms use a small frame size due to the non-stationary of voice that results in low delay. Finally, a good advantage for model-based voice compression is that they can easily cover up a packet loss due to the predictive nature of the algorithm. Since the algorithm extracts parameters of human voice production and since these parameters tend to be quite predictable from one frame to another, a packet loss can easily be covered up.

The main disadvantage of model-based voice compression is their high complexity. Because of the complex algorithms and of the lengthy search for optimal parameters, implementing such algorithms in real time requires considerable processing power. Another disadvantage comes from the fact that model-based speech compression algorithms were designed only to compress voice. This means that these algorithms are not good enough for encoding background noise, music, fax signals and telephone tones (DTMF) [29].

4.3 Characteristics of International Standards

Having discussed the characteristics of the voice compression families in general, we will now consider, in more detail, the characteristics of some existing international standards. These characteristics are summarized in table 4.2 [30] [31] [32].

Standard	Compression Family	Bit rate (kb/s)	Algorithmic Delay (ms)
G.711 (PCM)	Waveform	64, 56 or 48	0
G.722 (SB-ADPCM)	Waveform	64, 56 or 48	0.125
G.726 (ADPCM)	Waveform	16, 24, 32 or 40	0.125
MPEG-2 AAC (for sampling at 8 kHz)	Perceptual	Variable up to a maximum of 48	256
G.723.1 (MPC-MLQ)	Model	5.3 or 6.4	37.5
G.729 (CS-ACELP)	Model	8	15
G.729A (CS-ACELP)	Model	8	15

Table 4.2: Characteristics of International Standards.

Note that the table given so far did not include the complexity of the algorithms. This is because for many of these algorithms, no standard code exists and the complexity varies from one implementation to the next. For the three most popular voice compression algorithms, the source code for their implementation has been standardized. For this reason, it is possible to evaluate the complexity of these algorithms. This complexity is illustrated in Table 4.3.

Attribute	G.711	G.723.1	G.729	G.729A
MIPS required	-	16	20	10.5
RAM requirement	-	2200 words	3000 words	2000 words

Table 4.3: Complexity of Compression Standards.

Chapter 5: Performance Evaluation of Voice Compression for VoIP

5.1 Subjective Tests

Subjective Tests rely on the fact that since a human will be the one using the telephony system, then a human should be evaluating its quality. Subjective testing is usually the preferred method to evaluate voice quality since not many effective objective methods are currently available and accepted. The main problem with such test is that they are very time-consuming and expensive to perform as they involve testing by human subjects. An additional problem is that due to their subjective nature, the results cannot be achieved and they can vary depending on the listeners.

The international standard for subjective voice quality assessment is ITU-T P.800. Another standard, ITU-T P.830, specifies how these subjective tests should be performed. Several subjective tests are described in this standard; however, the most accepted one is based on an Absolute Category Rating (ACR). In such a test, each subject listens to a list of short speech files and attributes to it a score between 1 and 5 where: Excellent = 5; Good = 4; Fair = 3; Poor = 2; Bad = 1. After the voice file has been evaluated by numerous listeners, the score attributed to the voice file is averaged out to provide a Mean Opinion Score (MOS). The higher the value of the MOS, the better the quality of the voice is. The exact testing procedure and the recording environment for the voice files are stringently regulated in the standard.

Another method of subjective testing is based on the Comparison Category Rating (CCR). This method is similar to ACR except that here the quality of the voice file is compared to a reference voice file. For each voice file, the subject listens to the reference voice and the compressed one. The order in which they are present is random in order to remove any bias from the experiment. The score assigned compares whether the second version of the voice file is better or worse than the first version. The score given can be: (3) much better, (2) better, (1) slightly better, (0) about the same, (-1) slightly worse, (-2) worse, (-3) much worse. Once each listener has attributed a score to each voice file, the scores are averaged out to give a Comparison Mean Opinion Score (CMOS). A positive value of CMOS indicates that the modified or compressed version is better while a negative value indicates that the reference version is better.

5.2 Objective Tests

Even though subjective tests are better practiced to evaluate voice quality, methods of objective tests have ignited much interest because of their inherent advantages over a subjective test. The main advantage is that objective tests cost a lot less to perform and can easily be done on a large scale since they do not require human intervention. In Subjective tests, due to the subjective nature of the tests, their validity sometimes comes in question since test results can vary with listeners and are hard to compare.

The simplest objective test is to calculate the Mean Square Error (MSE) or the Signal to Noise Ratio (SNR) between the compressed voice and the original voice. This method simply takes difference between voice sample, squares the difference and averages it out over many different test cases. The idea here is that a compressed voice file that is closer to the original voice file will sound better. Since the human ear cannot hear all elements of a voice signal, simply looking at the difference of speech file is not sufficient as this difference may not be audible to the listener.

5.2.1 VoIP Performance Monitoring paper

This paper describes a method for monitoring VoIP application based upon a reduction of the ITU-T's E-Model to transport level, measurable quantities [33]. The paper also discusses the tradeoffs between placing the monitors within the gateways versus placement of the monitors within the transport path.

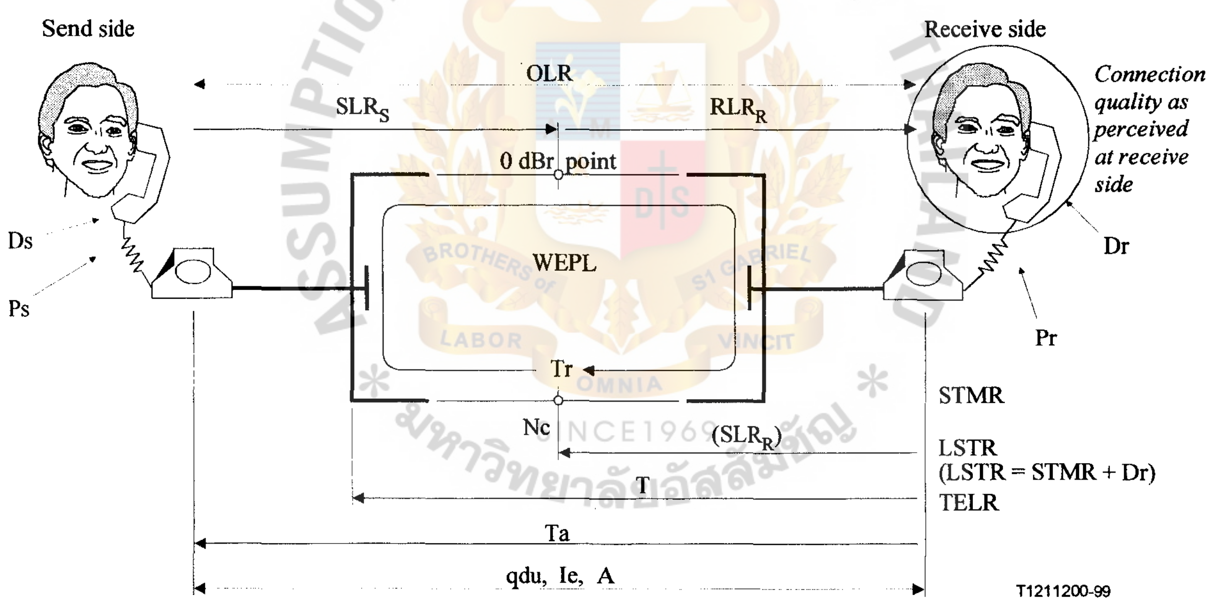

Figure 5.1: Basic reference configuration of the E-Model

The ITU-T's E-Model is a network planning tool used to design hybrid circuit-switched and packet-switched networks for carrying high quality voice applications. The tool estimates the relative impairments to voice quality when comparing different network equipment and network designs. The tool provides means to estimate the subjective Mean Opinion Score (MOS) rating of voice quality over these planned network environments.

The specific method the paper advocates is to:

- Measure the low-level transport metrics (characterizing the channel), which impact voice performance, i.e., delay, delay variation and packet loss.
- Combine the packet loss and delay variation measurements, de-jitter buffer operations, packet size and coder frame size into an error mask (the exact sequence of good and bad coder frames) that can be characterized in a simple manner (e.g., average frame loss rate along with some measure of burstiness).
- Combine the characterized error mask with the coder and its frame-loss concealment algorithm via a look-up table (or curve fit) based on subjective testing to produce an E-Model equipment impairment factor (I_{ef}).
- Combine the I_{ef} with other E-Model low-level measurable elements, i.e., delay and echo, to produce a predicted opinion score on the quality of the voice conversation.

The paper's goal is to build a relatively simple VoIP performance monitoring capability.

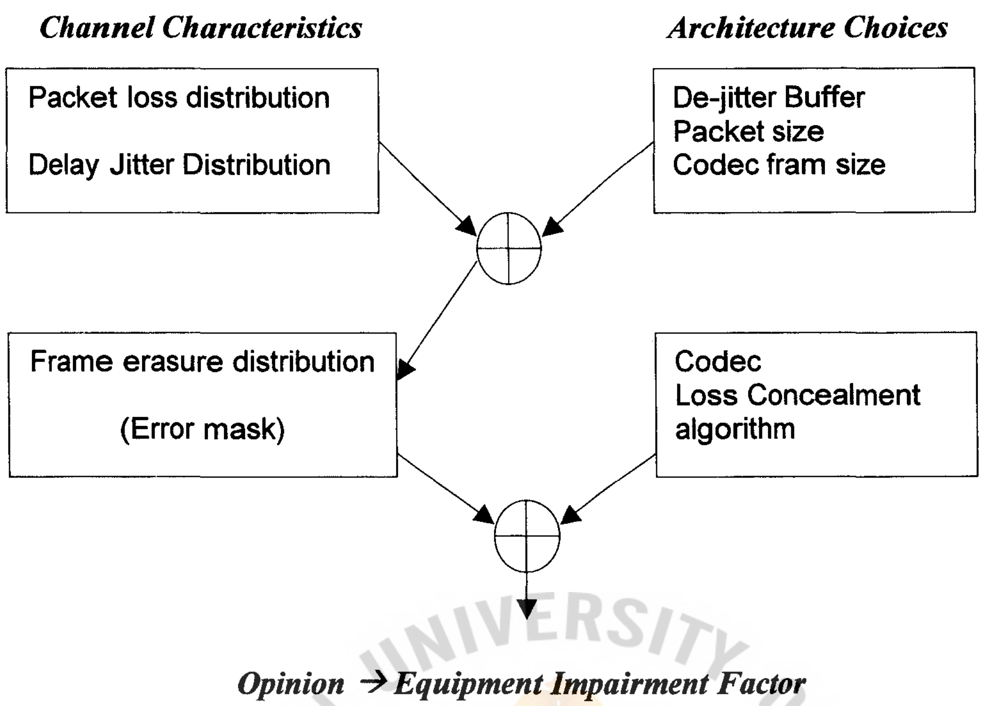

Figure 5.2: Measurement and data reduction methodology for VoIP quality monitoring, which highlight the equipment impairment factor elements

Benefits:

Among the benefits of this paper we can find that the paper suggests a relatively simple VoIP performance monitoring, the paper taking into consideration the delay, delay variation, packet loss, jitter and the de-jitter buffer loss.

Drawbacks:

The paper does not take into consideration the speech compression. It is true that types of delay will affect the quality of the voice. The speech compression type may also affect the voice quality.

5.3 A Novel Objective Method to Evaluate Voice Quality of Voice Compression for VoIP

5.3.1 The method

I build a two-way communication VoIP program (see Appendix A an B) which has been used to transfer different compressed voice files. Transfer of the files has been done in 2 different speeds: 10 Mb/s and 44 Kb/s. The received file was recorded and checked with WaveLab program.

The test bed block diagram can be seen below in figure 5.3

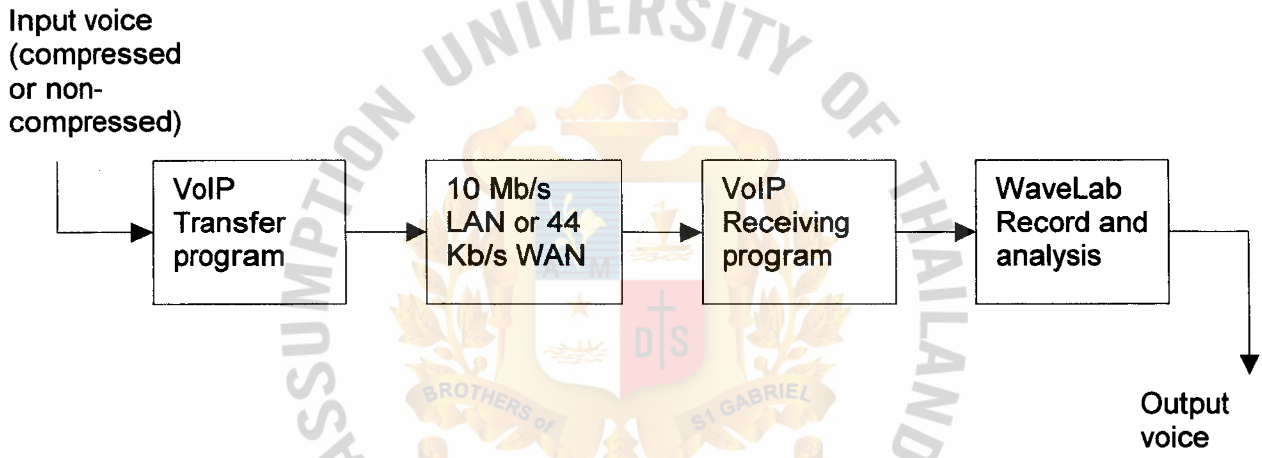

Figure 5.3: Block diagram of the test bed

Figure 5.4 shows the two different network environments that have been used in the testing. In the LAN environment 20 computers network have been used when 2 computers are communicating to each other. In the WAN environment, one computer was connected to a 128 Kb/s leased line while the other computer was connected with a dial up connection at the speed of 44 Kb/s to the Internet.

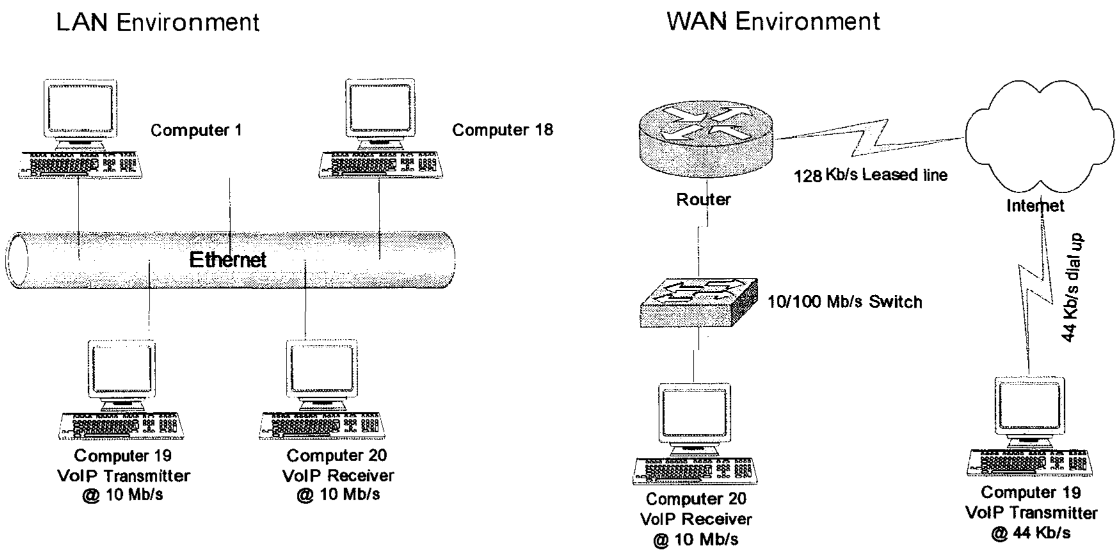

Figure 5.4: Test bed network environment

Two types of analysis have been done thru the WaveLab program: 1. Spectrum Analyzer window which uses FFT (Fast Fourier Transform) to display a continuous frequency graph; and 2. 3D Frequency Analysis. Graphs will be shown later in this chapter.

The parameter to decide which compression method is more suitable to be used with VoIP according to the graphs will be: the compressed file which its graph is more close in terms of strength and frequencies peeks to the graph of the non compressed file will be considered as better compression method.

The Transmitted files contain speech in Japanese of 4 situations: man voice, woman voice, man voice in noisy environment, women voice in noisy environment.

While transmitting the files, Iris, network sniffer has been used to sniff the network and to check the packet on both the transmitting side and the receiving side. The result will be shown later in this chapter.

The parameter to decide which compression method is more suitable to be used with VoIP according to the packet analyzing will be: the compressed file which has fewer packets to be transmitted / received will be considered as more suitable compression method.

Also the size of the files and its duration have been recorded and will be shown later in this chapter.
In the case of file size, the parameter to decide which compression method is more suitable to be used with VoIP will be: the compressed file that will have the smallest size will be considered as the best compression method.

In order to confirm the results, subjective tests have been performed by giving 53 people to listen to the received files, MOS, STD and VAR have been calculated and will be shown also later in this chapter. The 53 listeners are between the age of 20 to 35, most of them with university level education and are approximately half men and half women.

5.3.2 Spectrum Analyzer results

Each of the received compressed voice file is compared against the original, uncompressed voice file. The following figures can show the results.

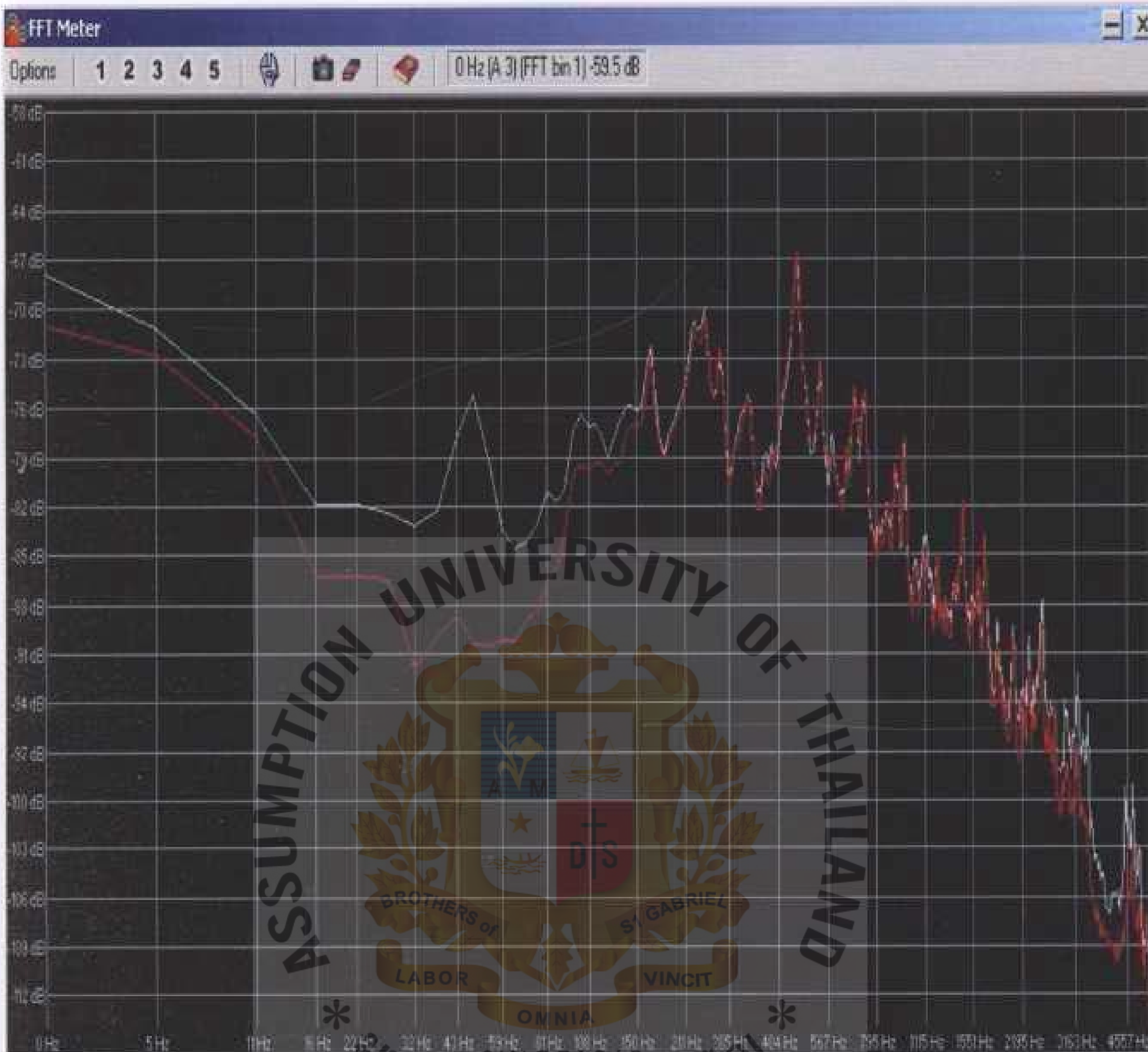

Figure 5.5: FFT Comparison between Original (white) file and G.729 Compressed file (red) over 44 Kb/s

In the graph, we can see a compression between the non compressed file (white) and the G.729 Compressed file (red). The strength of the compressed file is lower in the lower frequencies.

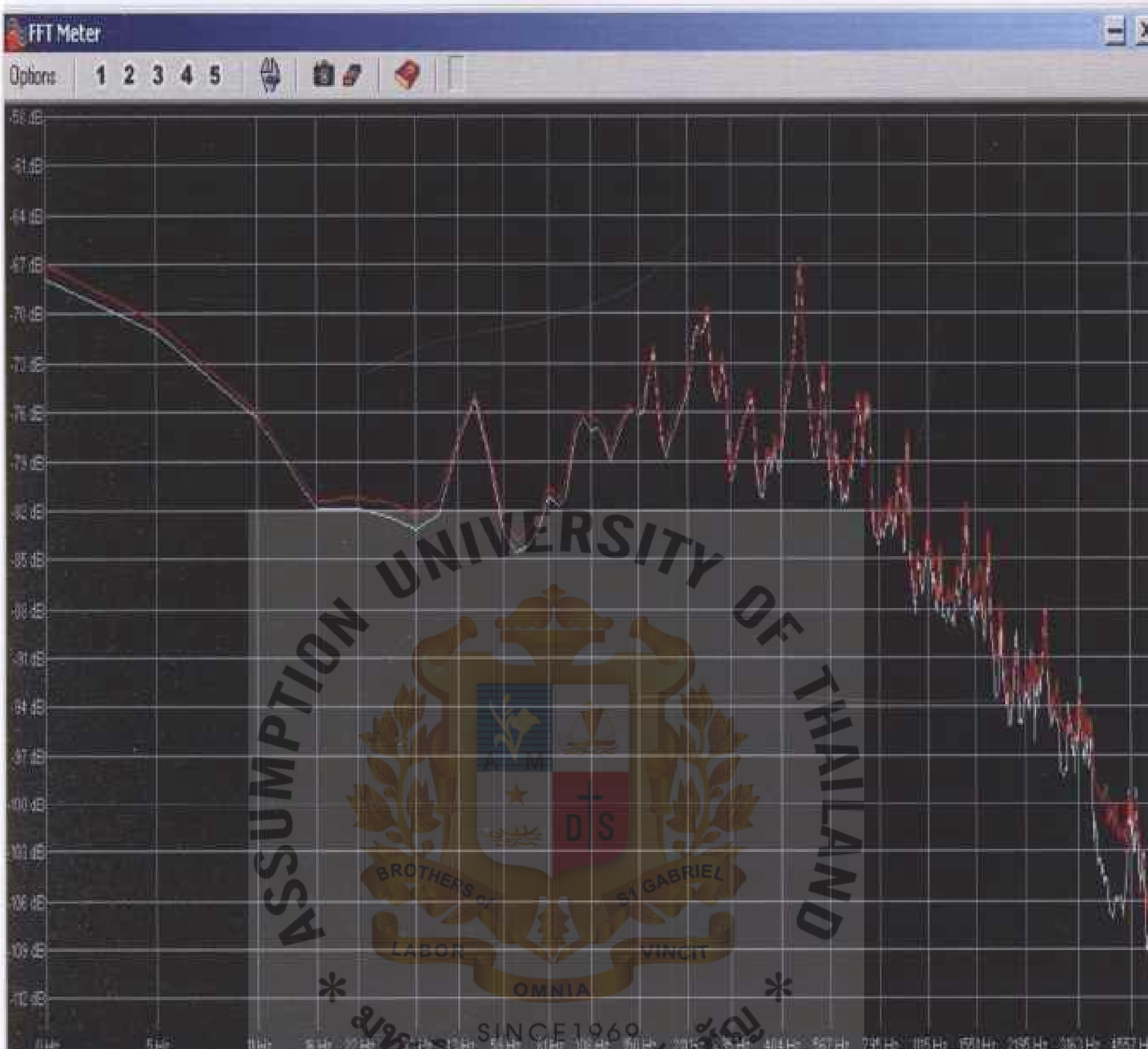

Figure 5.6: FFT Comparison between Original (white) file and G.726 16 kbit/s bit rate Compressed file (red) over 44 Kb/s

In the graph, we can see a compression between the non compressed file (white) and the G.726 Compressed file (red). The strength of the compressed file is slightly higher than the original non compressed file.

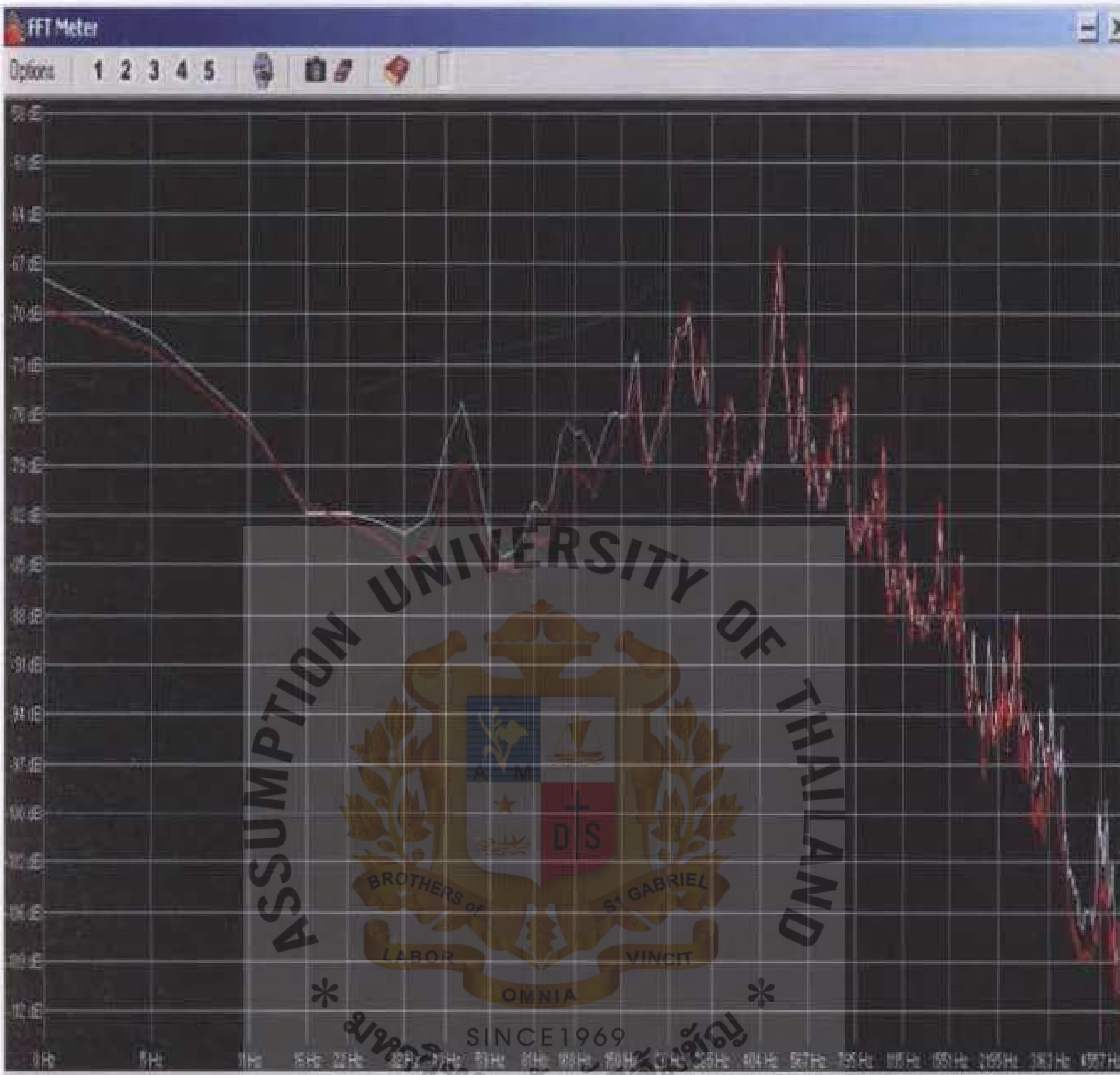

Figure 5.7: FFT Comparison between Original (white) file and G.723.1 5.3 kbit/s bit rate Compressed file (red) over 44 Kb/s

In the graph, we can see a compression between the non compressed file (white) and the G.723.1 Compressed file (red). The strength of the compressed file is slightly lower than the original non compressed file.

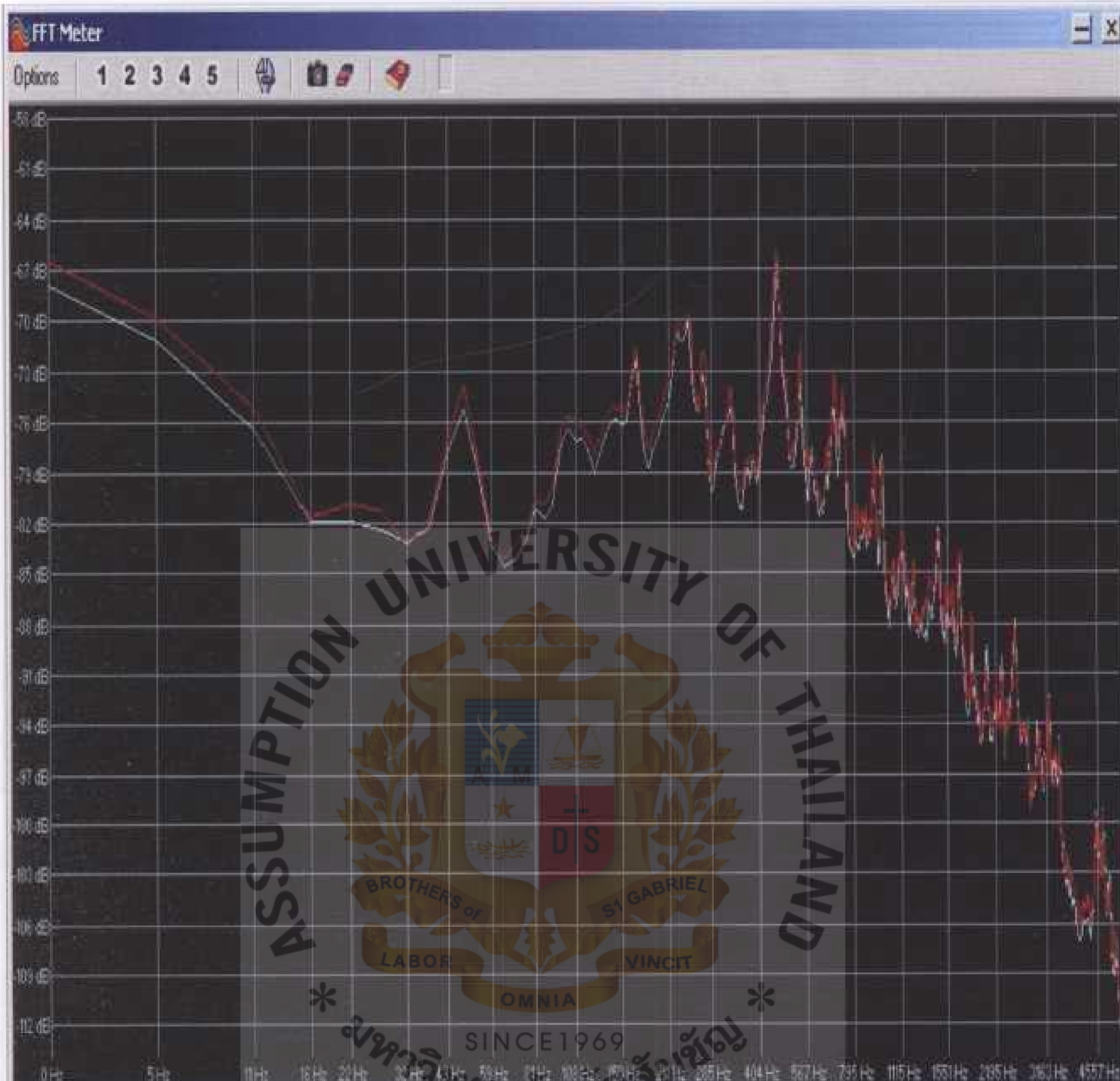

Figure 5.8: FFT Comparison between Original (white) file and G.711A Compressed file (red) over 44 Kb/s

In the graph, we can see a compression between the non compressed file (white) and the G.711A Compressed file (red). The strength of the compressed file is slightly higher than the original non compressed file.

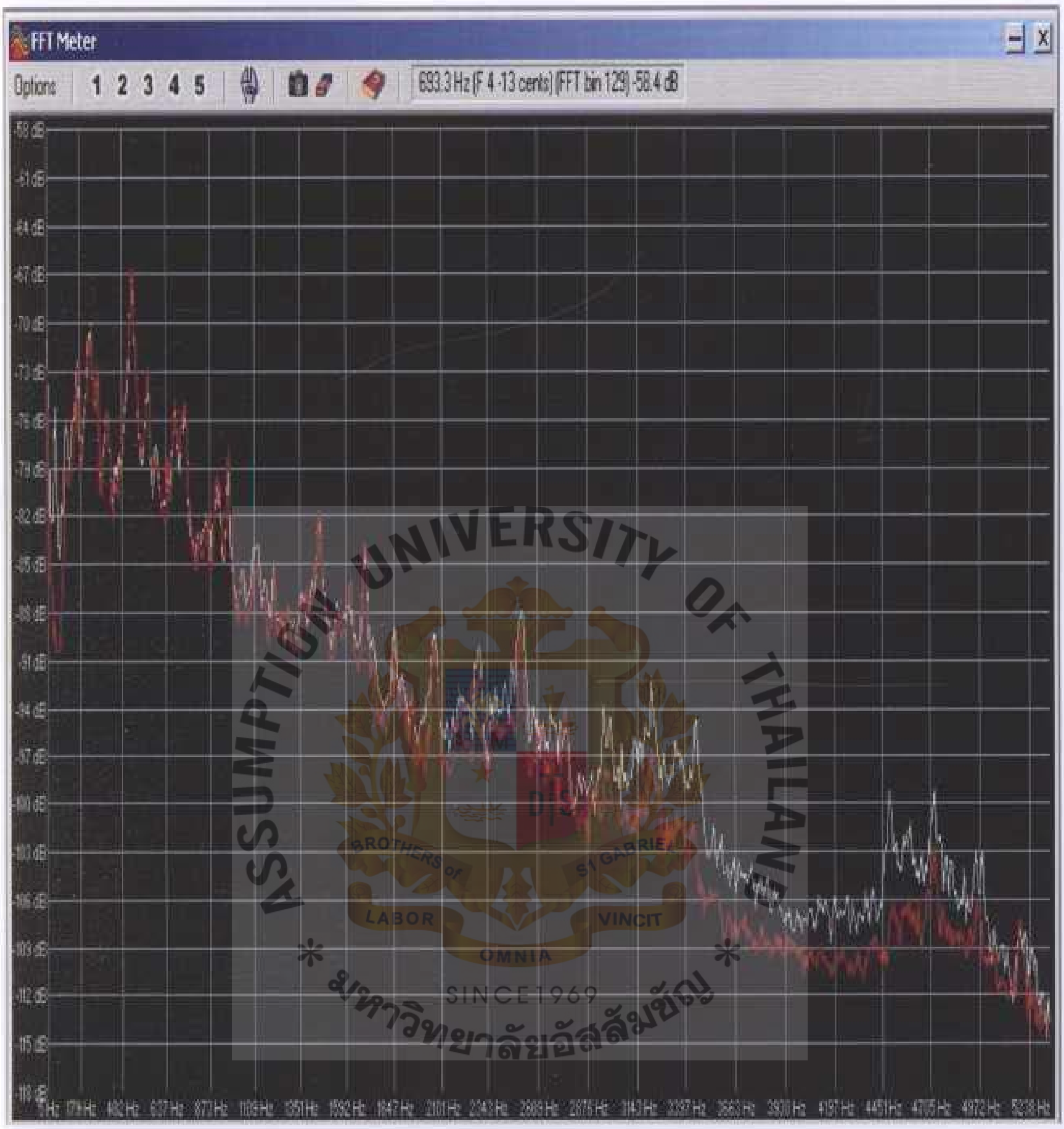

Figure 5.9: FFT Comparison between Original (white) file and G.729 Compressed file (red) over 44 Kb/s in logarithmic frequency scale.

This graph show us the another comparison between the original file and the G.729 compressed file. This comparison is represented in logarithmic frequency scale so we can see the differences in the high frequency more clearly.

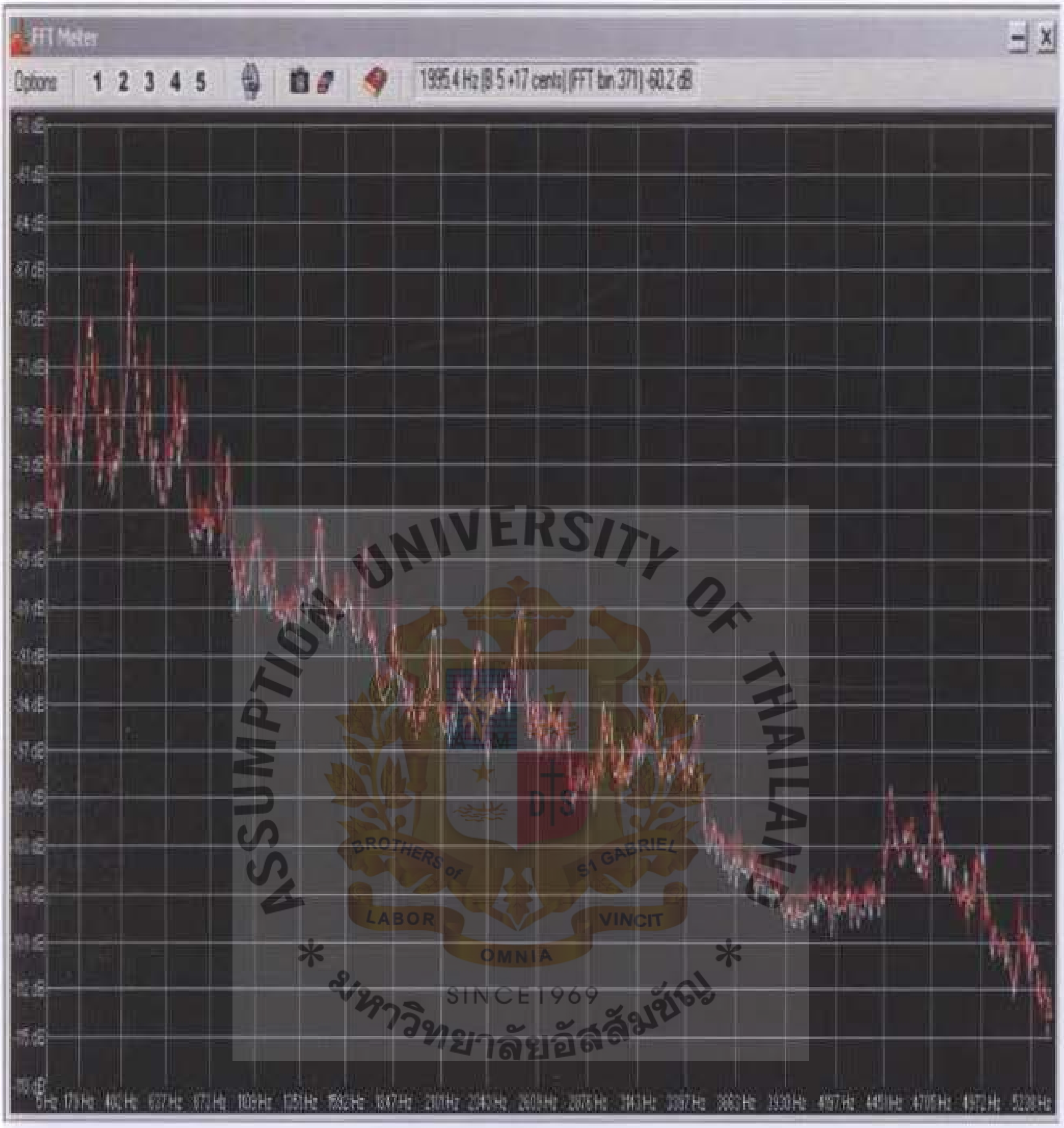

Figure 5.10: FFT Comparison between Original (white) file and G.711 Compressed file (red) over 44 Kb/s in logarithmic frequency scale.

This graph show us the another comparison between the original file and the G.711 compressed file. This comparison is also represented in logarithmic frequency scale.

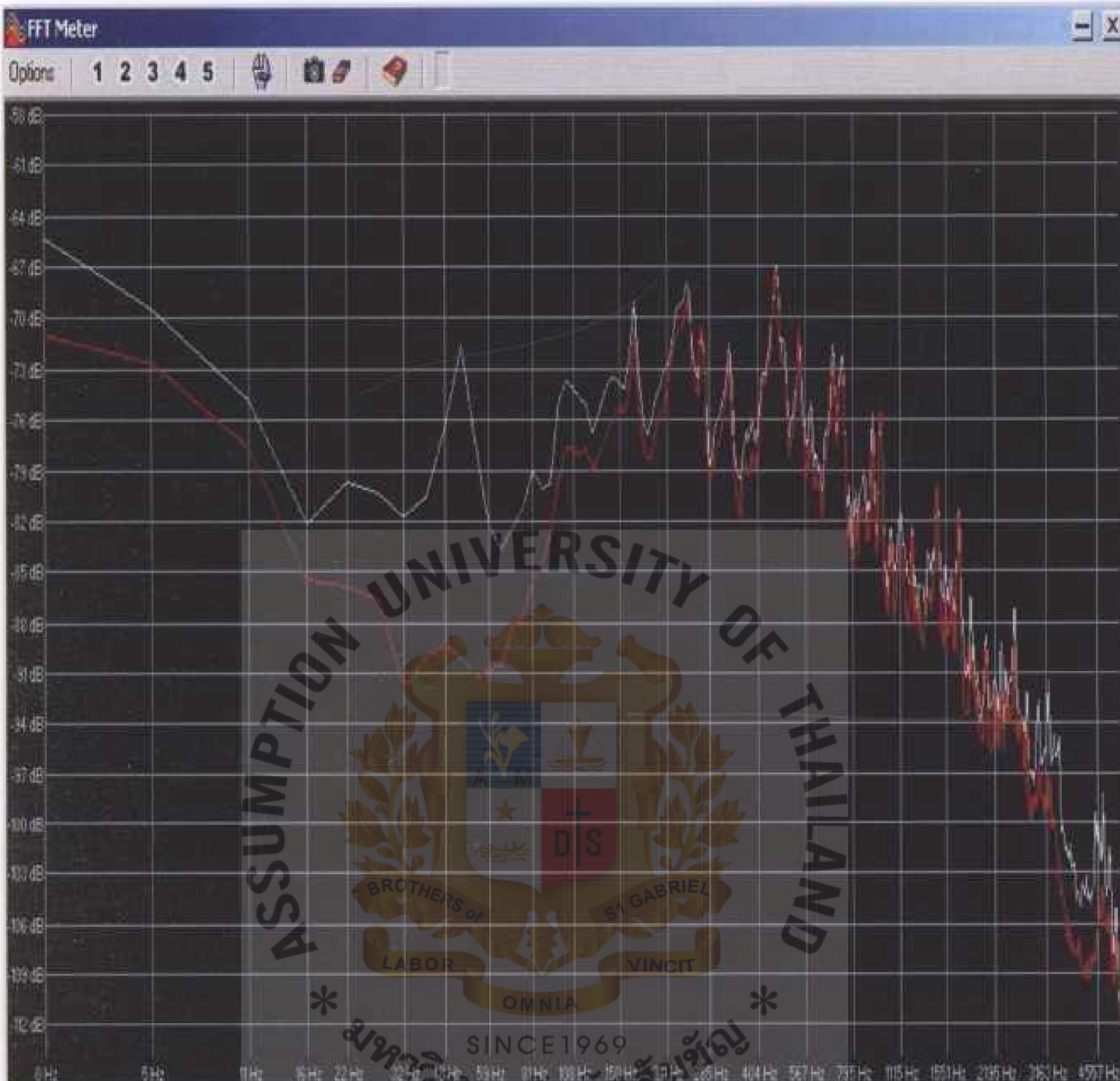

Figure 5.11: FFT Comparison between Original (white) file and G.729 Compressed file (red) over 10 Mb/s

Again, in the graph we can see a compression between the non compressed file (white) and the G.729 Compressed file (red). The strength of the compressed file is lower especially in the lower frequencies.

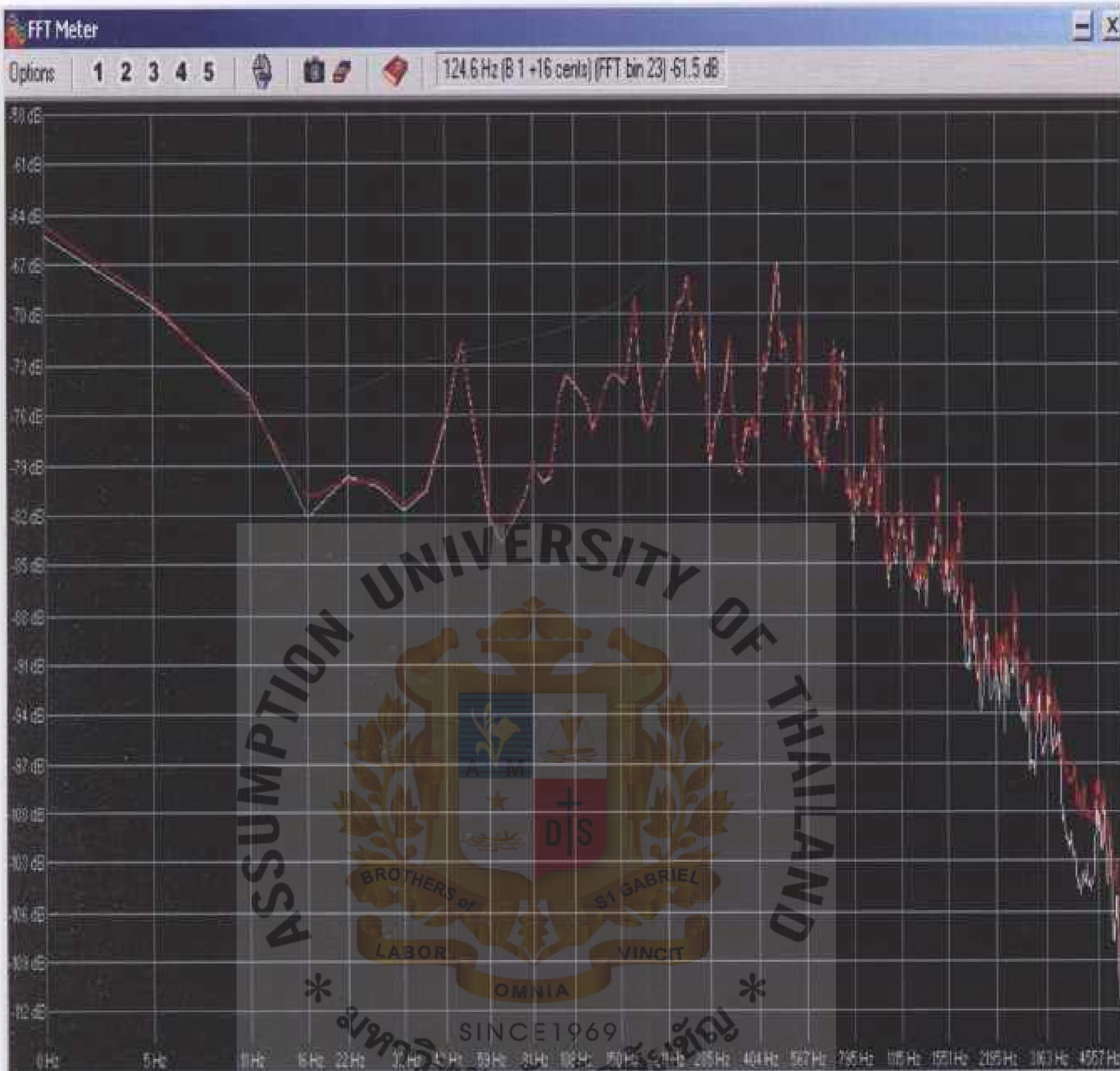

Figure 5.12: FFT Comparison between Original (white) file and G.726 16 kbit/s bit rate Compressed file (red) over 10 Mb/s

In the graph, we can see a compression between the non compressed file (white) and the G.726 Compressed file (red). The strength of the compressed file is slightly higher than the original non compressed file.

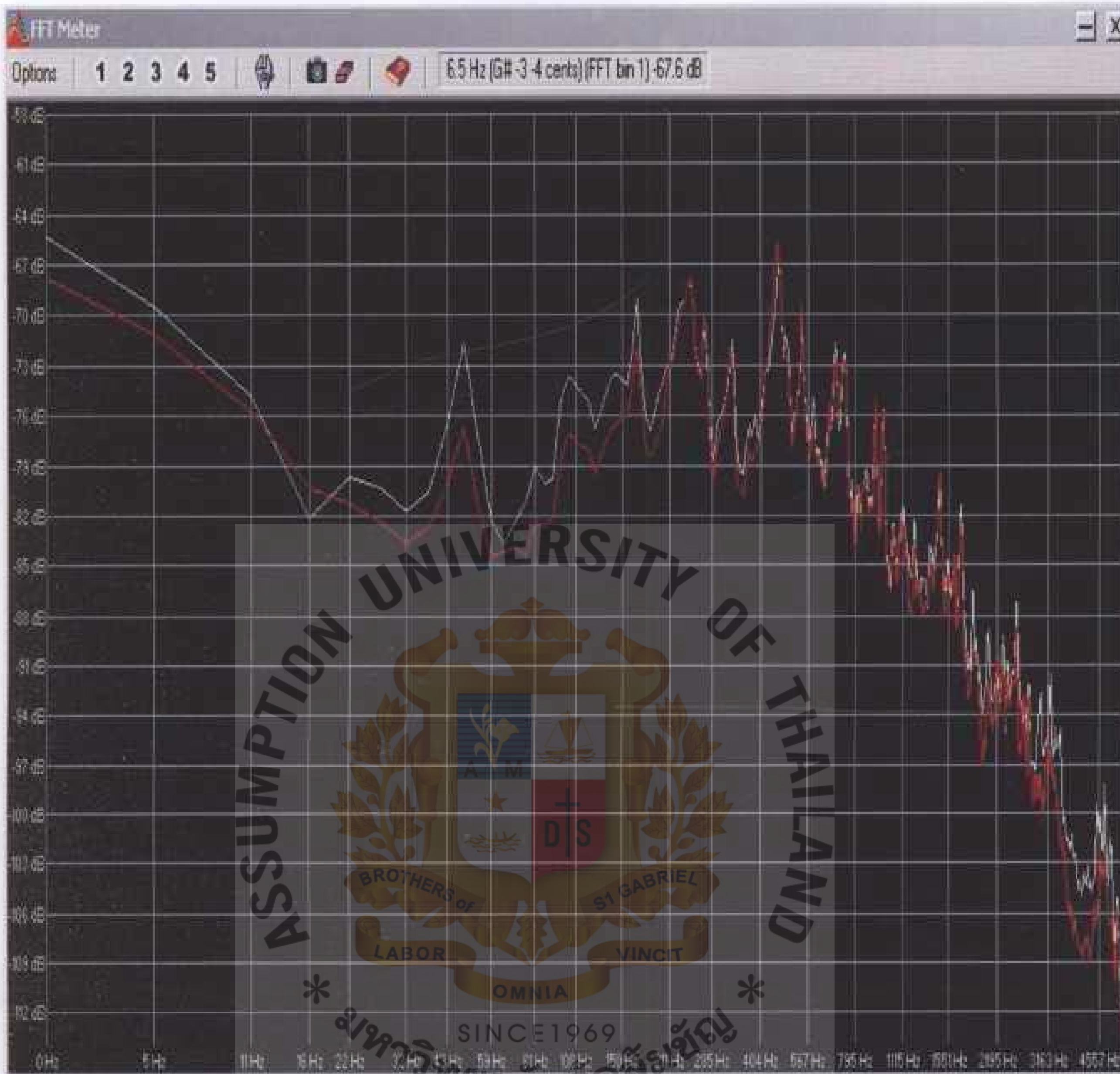

Figure 5.13: FFT Comparison between Original (white) file and G.723.1 5.3 kbit/s bit rate Compressed file (red) over 10 Mb/s

In the graph, we can see a compression between the non compressed file (white) and the G.723.1 Compressed file (red). The strength of the compressed file is slightly lower than the original non compressed file.

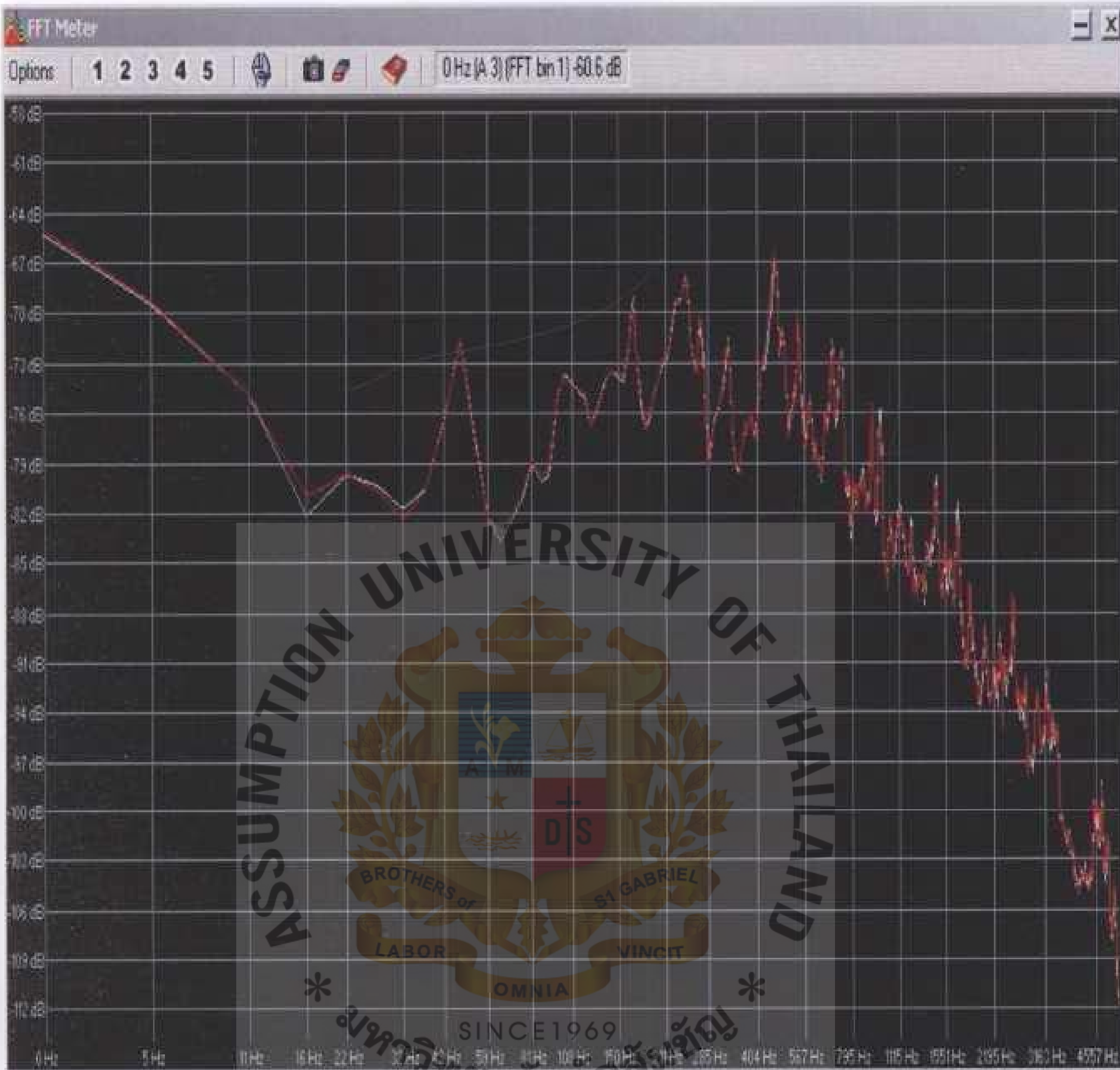

Figure 5.14: FFT Comparison between Original (white) file and G.711A Compressed file (red) over 10 Mb/s

In the graph, we can see a compression between the non compressed file (white) and the G.711A Compressed file (red). The strength of the compressed file is slightly higher than the original non compressed file.

We can see from the graphs that both over 10 Mb/s and 44 Kb/s, the shape of the graphs have not been changed and that the same compression method gives us the same graph shape in both speeds. The compressed voice file of G.726 and G.711A are at the same level or slightly better than the level of the original file and according to the graph can be considered as better compression type. The files of the other two compressing methods G.729 and G.723.1 are at lower level than the original voice file.

Table 5.1 summarized the information regarding the peak and the average of the loudness obtained from the graphs.

	G.729 (dB)	G.726 (dB)	G.723.1 (dB)	G.711A (dB)	Original (dB)
Peak @ 44 Kb/s	-31.02	-29.99	-31	-29.92	-30.04
Average @ 44 Kb/s	-52.73	-51.87	-52.02	**-51.5**	-52.39
Peak @ 10 Mb/s	-31.02	-30.03	-31	-30.05	-30.15
Average @ 10 Mb/s	-51.81	-50.28	-50.59	**-50.17**	-50.47

Table 5.1 Peak and the average of the loudness.

5.3.3 3D Frequency Analysis results

Each of the received files has been analyzed thru the 3D Frequency analyzer. The following figures show the results.

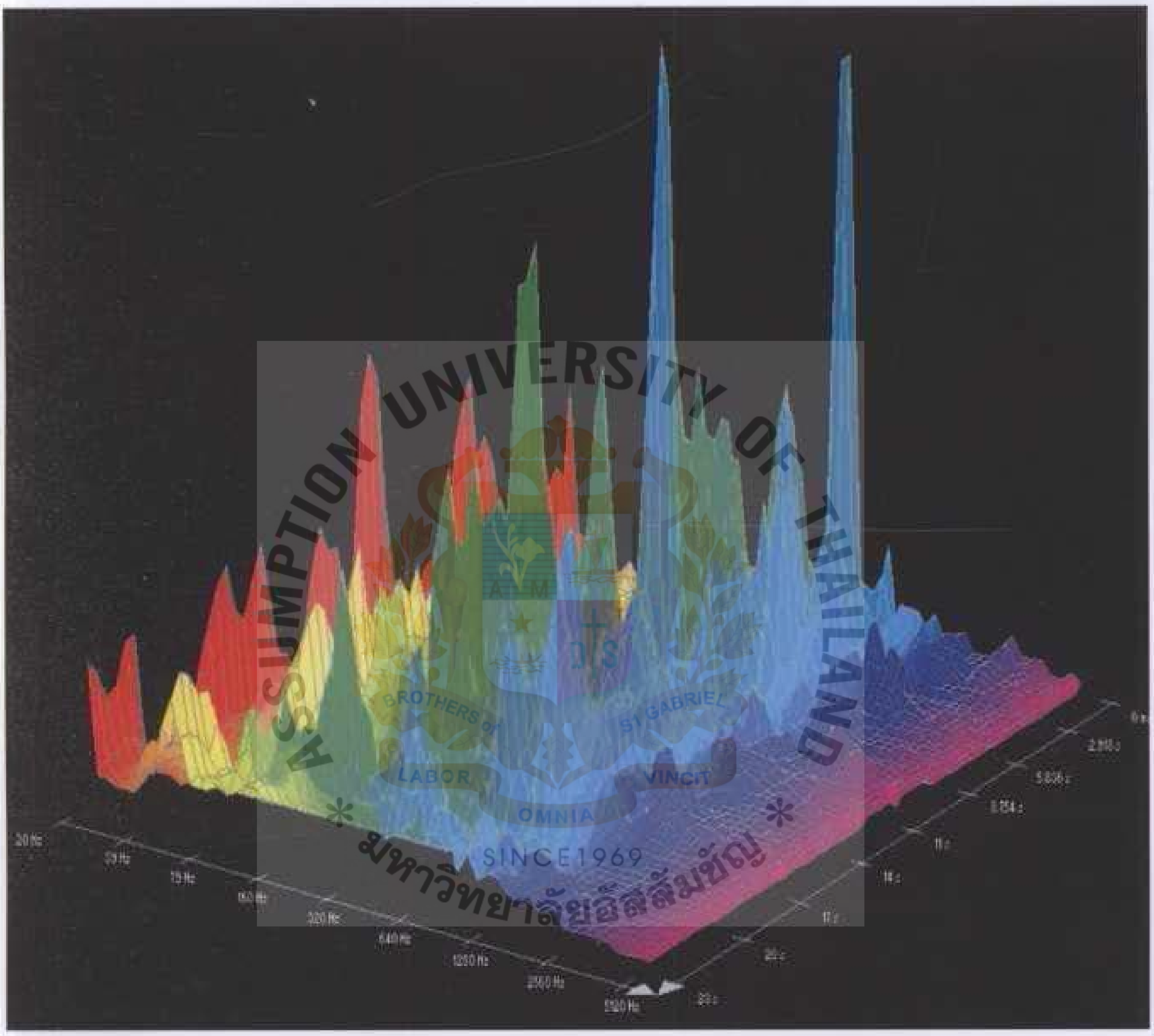

Figure 5.15: 3D Frequency Analysis of Original file over 44 Kb/s

In this original, non compressed file graph we can see a lot of frequency peaks.

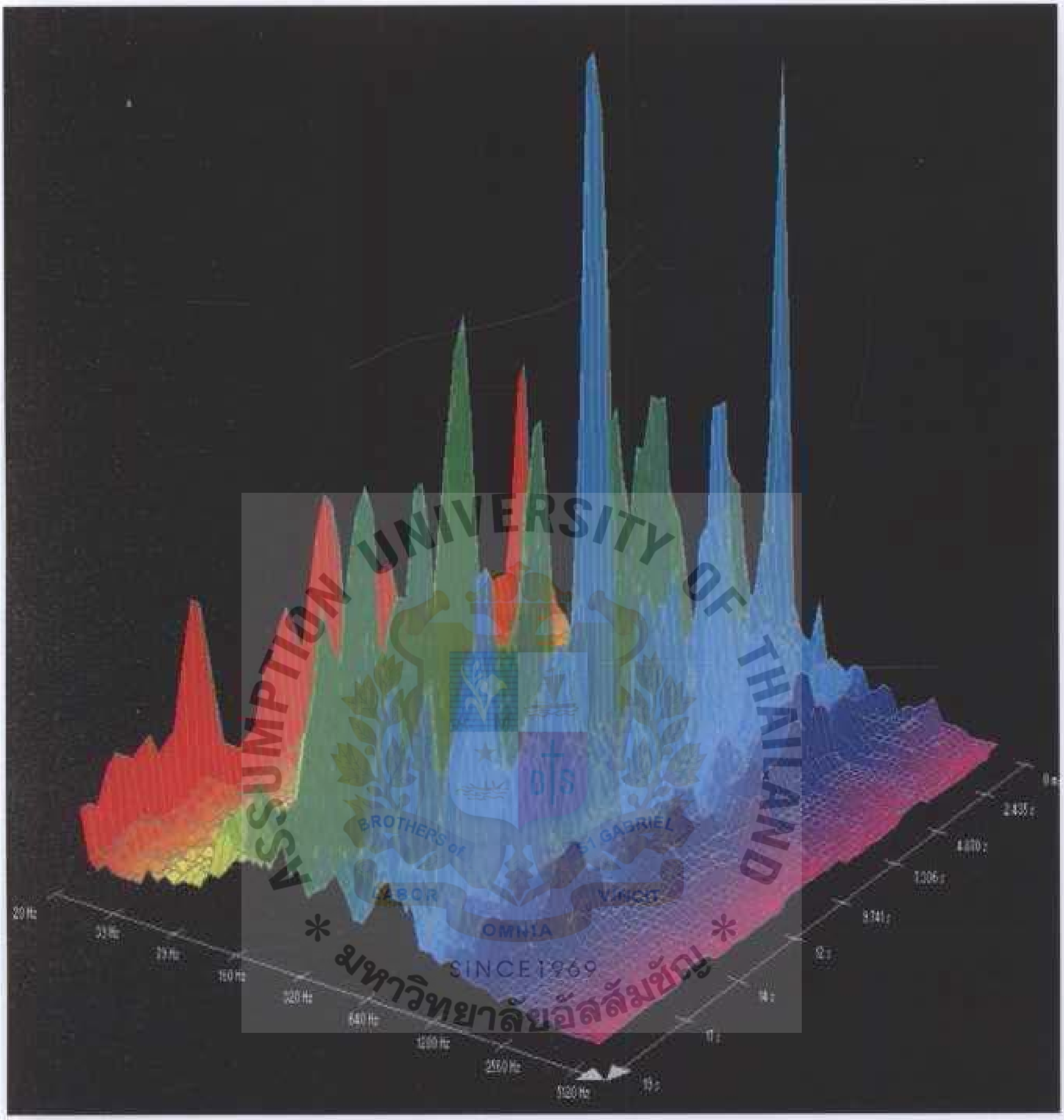

Figure 5.16: 3D Frequency Analysis of G.729 Compressed file over 44 Kb/s

In this G.729 compressed file graph we can see that the peaks of the lower frequencies (yellow) are not presented.

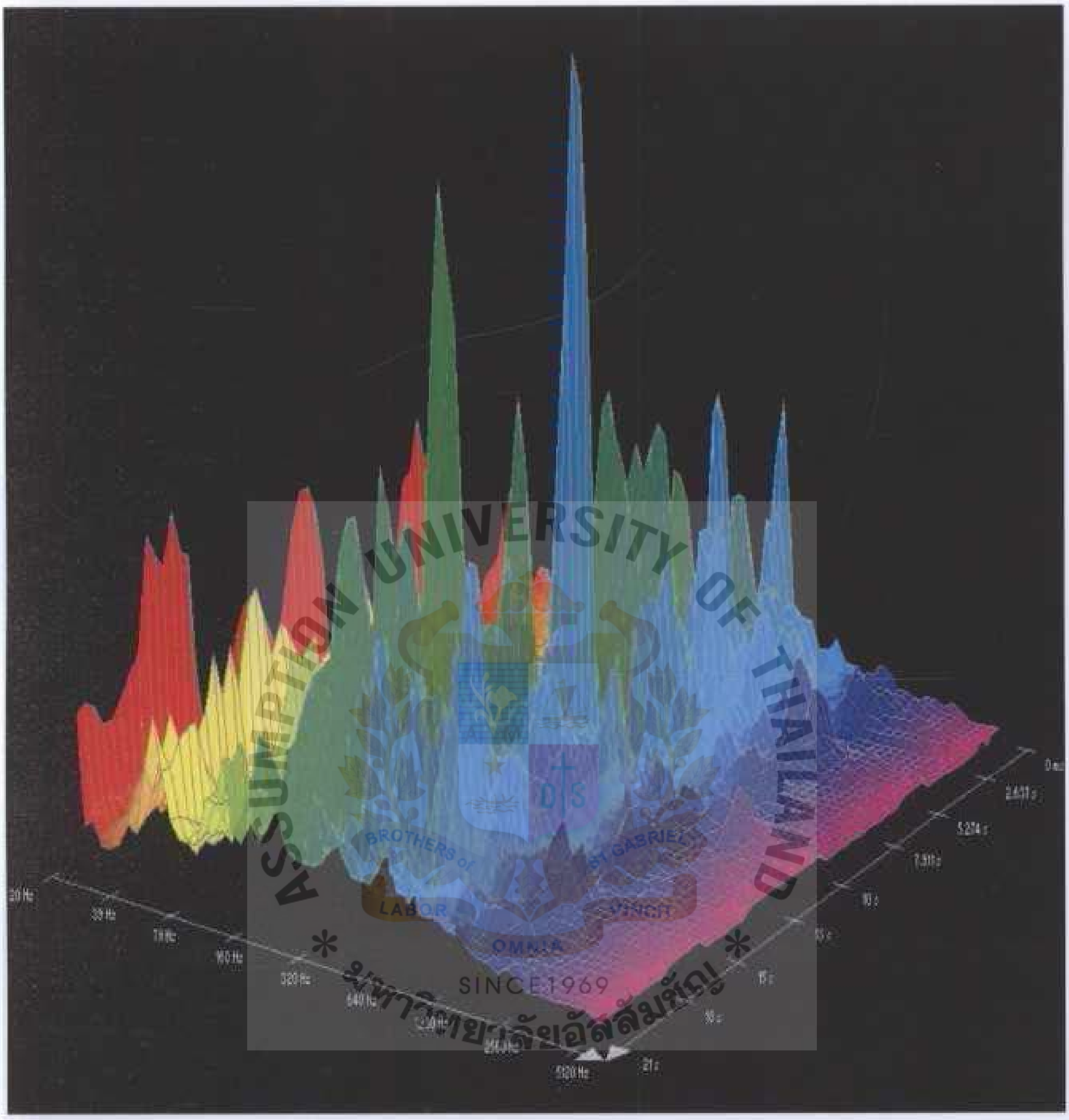

Figure 5.17: 3D Frequency Analysis of G.726 Compressed file over 44 Kb/s

In this G.726 compressed file graph we can see that one of the peaks of the medium frequencies (light blue) are not presented.

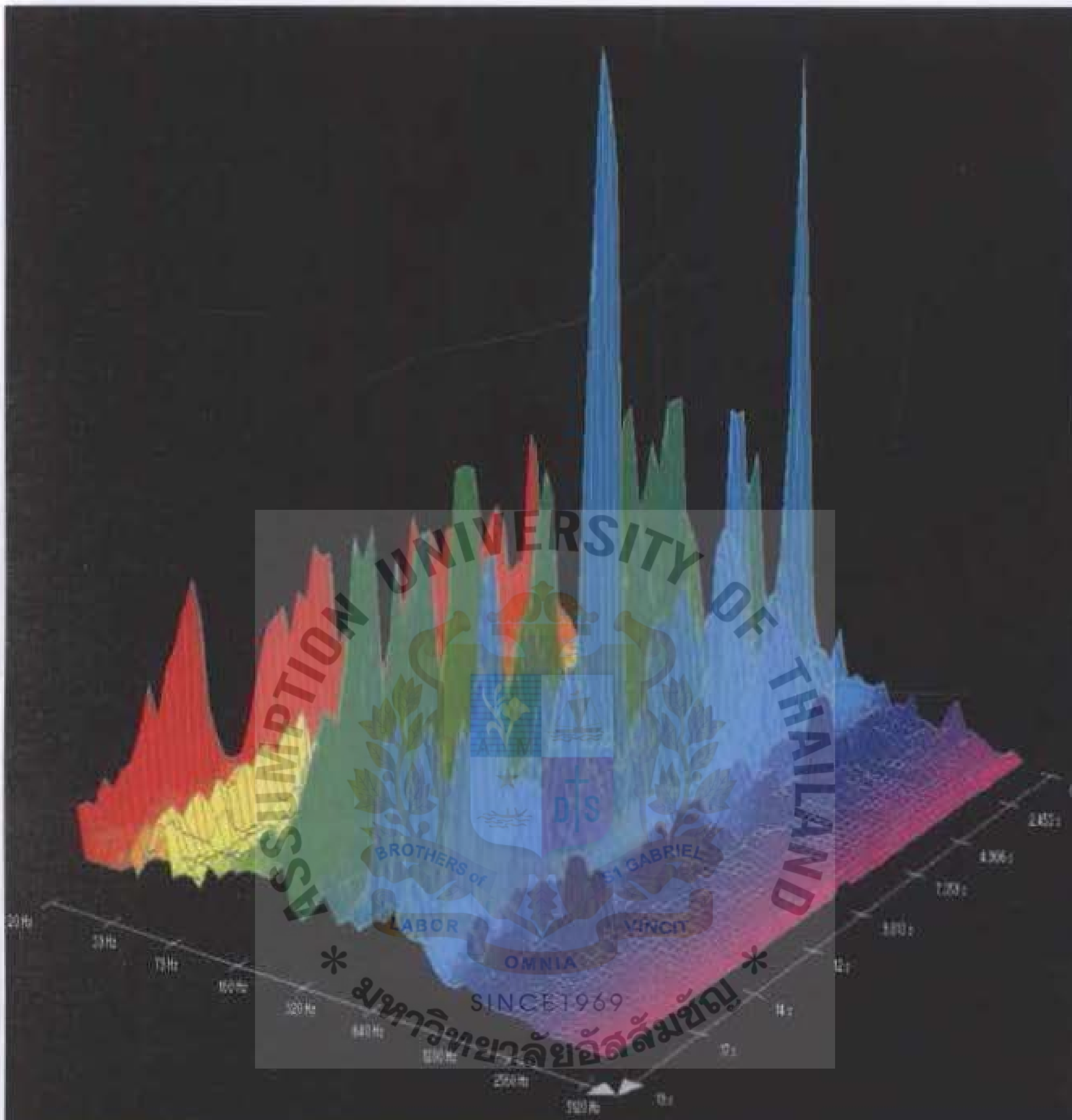

Figure 5.18: 3D Frequency Analysis of G.723.1 Compressed file over 44 Kb/s

In this G.723.1 compressed file graph, we can see that the heights of the peaks of the green and red frequencies have been reduced.

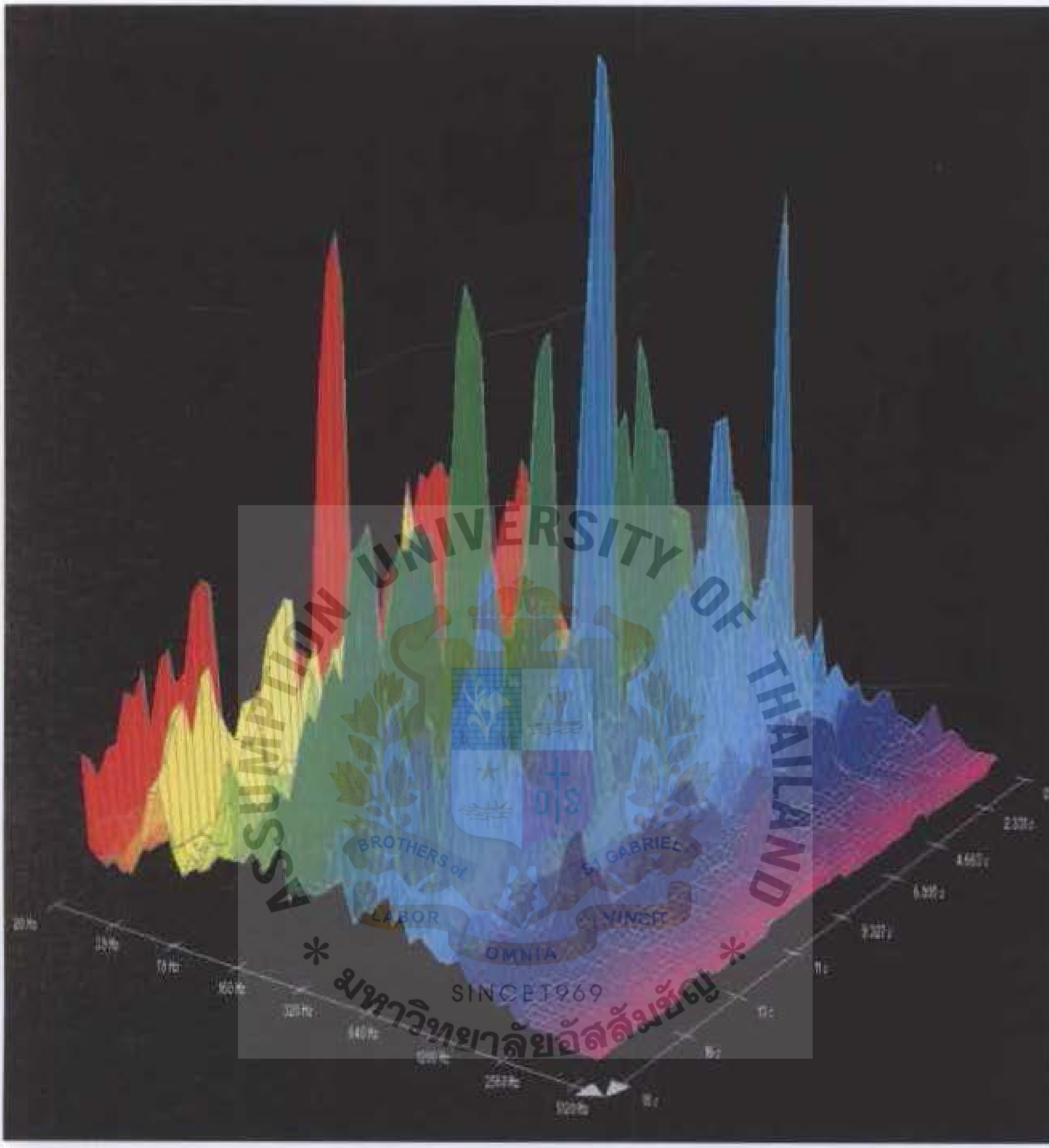

Figure 5.19: 3D Frequency Analysis of G.711A Compressed file over 44 Kb/s

In this G.711A compressed file graph we can see that all peaks of the original, non compressed file are presented and some frequencies (yellow) are even in a higher strength.

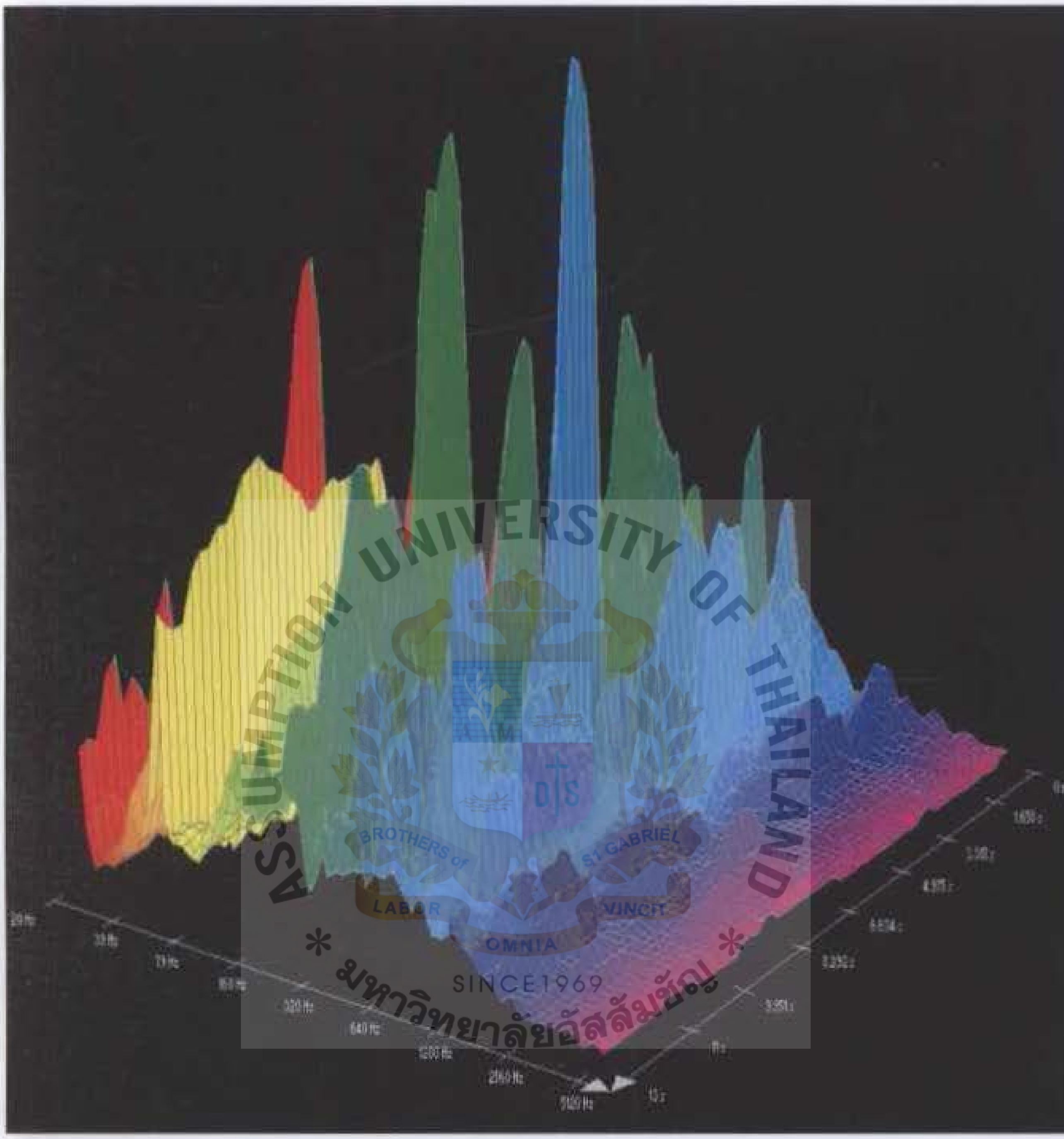

Figure 5.20: 3D Frequency Analysis of Original file over 10 Mb/s

In this original, non compressed file graph, we can see much higher peaks than those of the slower network of 44 Kb/s.

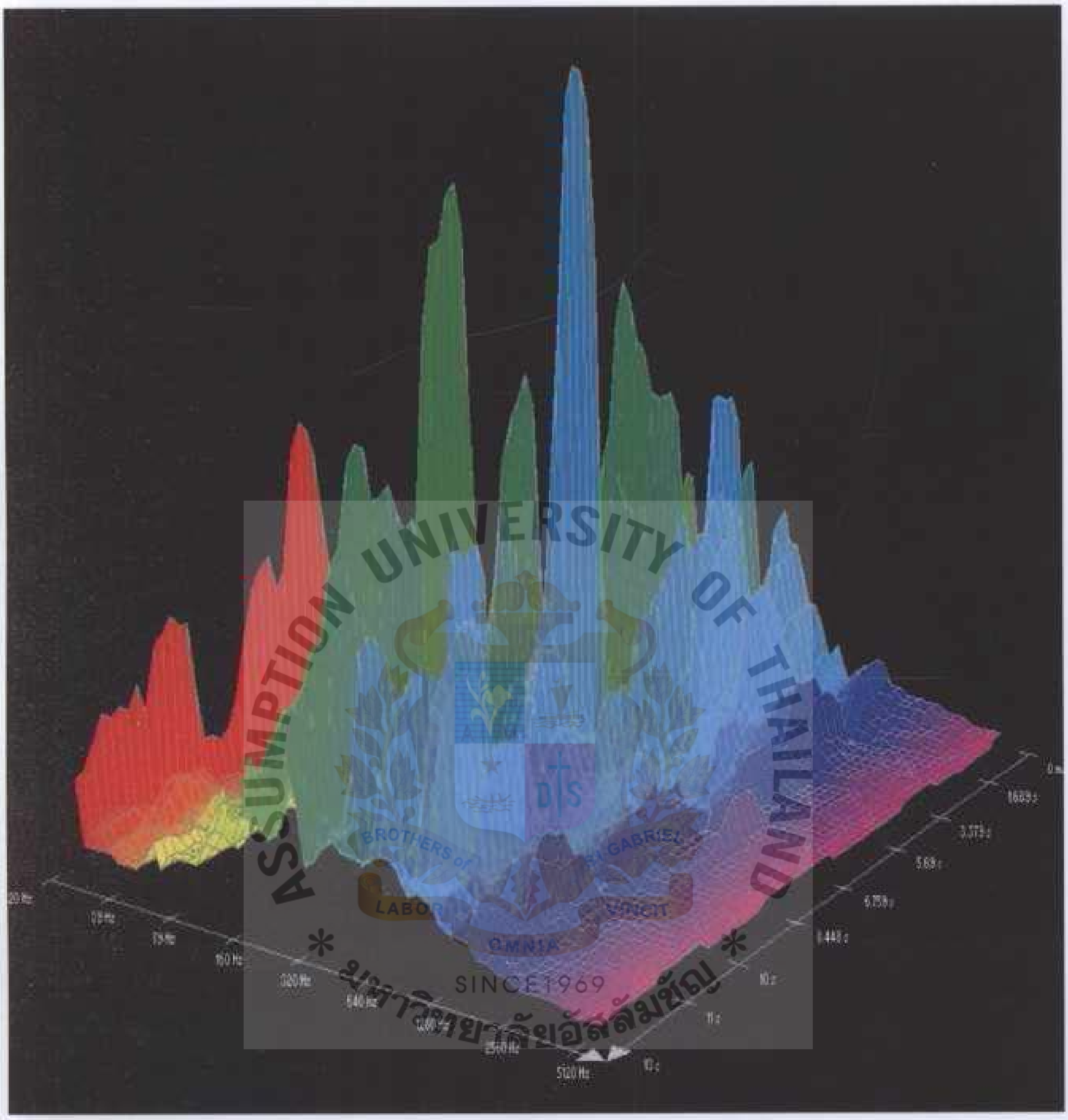

Figure 5.21: 3D Frequency Analysis of G.729 Compressed file over 10 Mb/s

Again, in this G.729 compressed file graph, we can see that the peaks of the lower frequencies (yellow) are not presented.

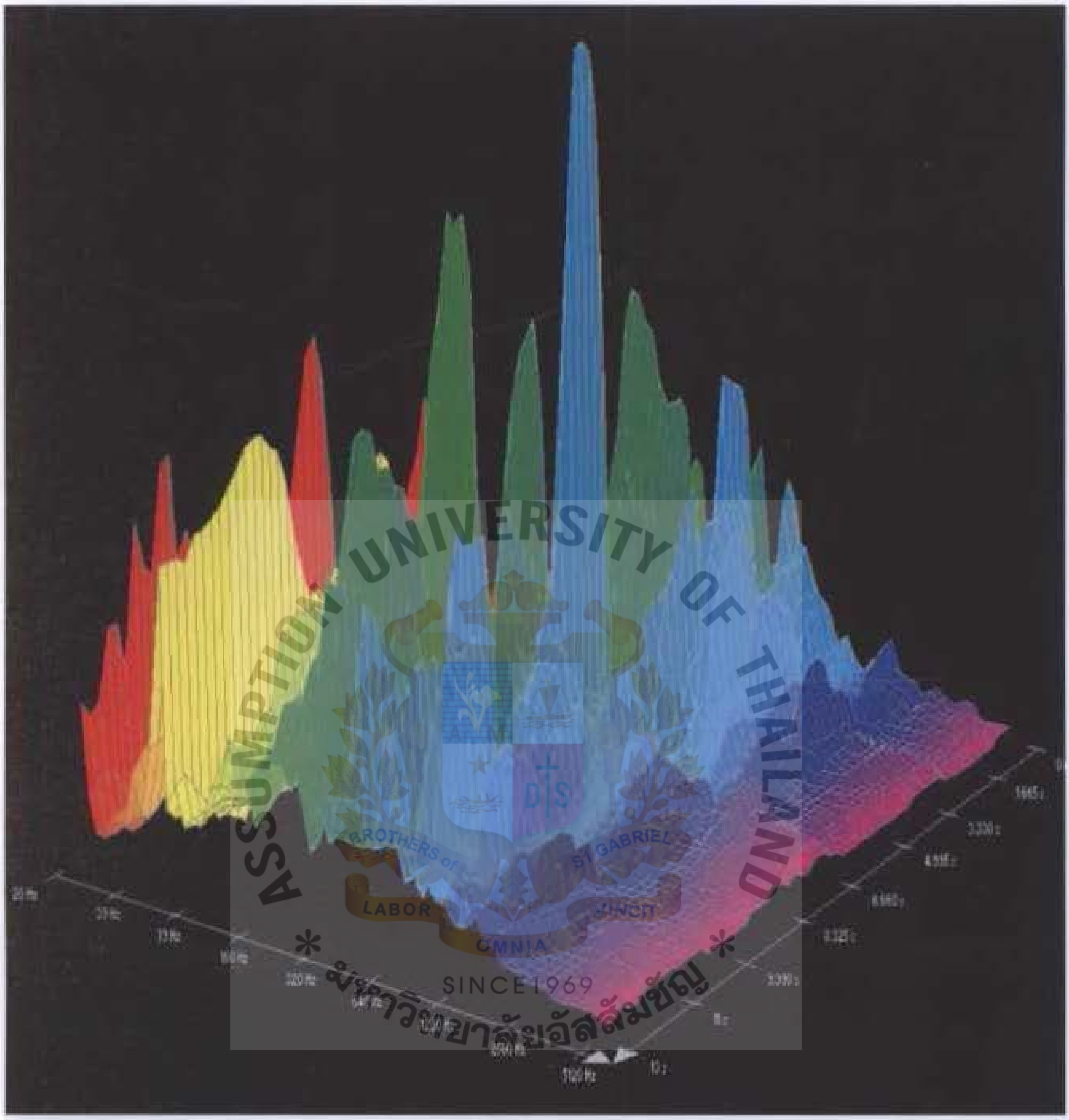

Figure 5.22: 3D Frequency Analysis of G.726 Compressed file over 10 Mb/s

In this G.726 compressed file graph, we can see that some of the peaks of the lowest frequency (red) are smaller than the original, non compressed file.

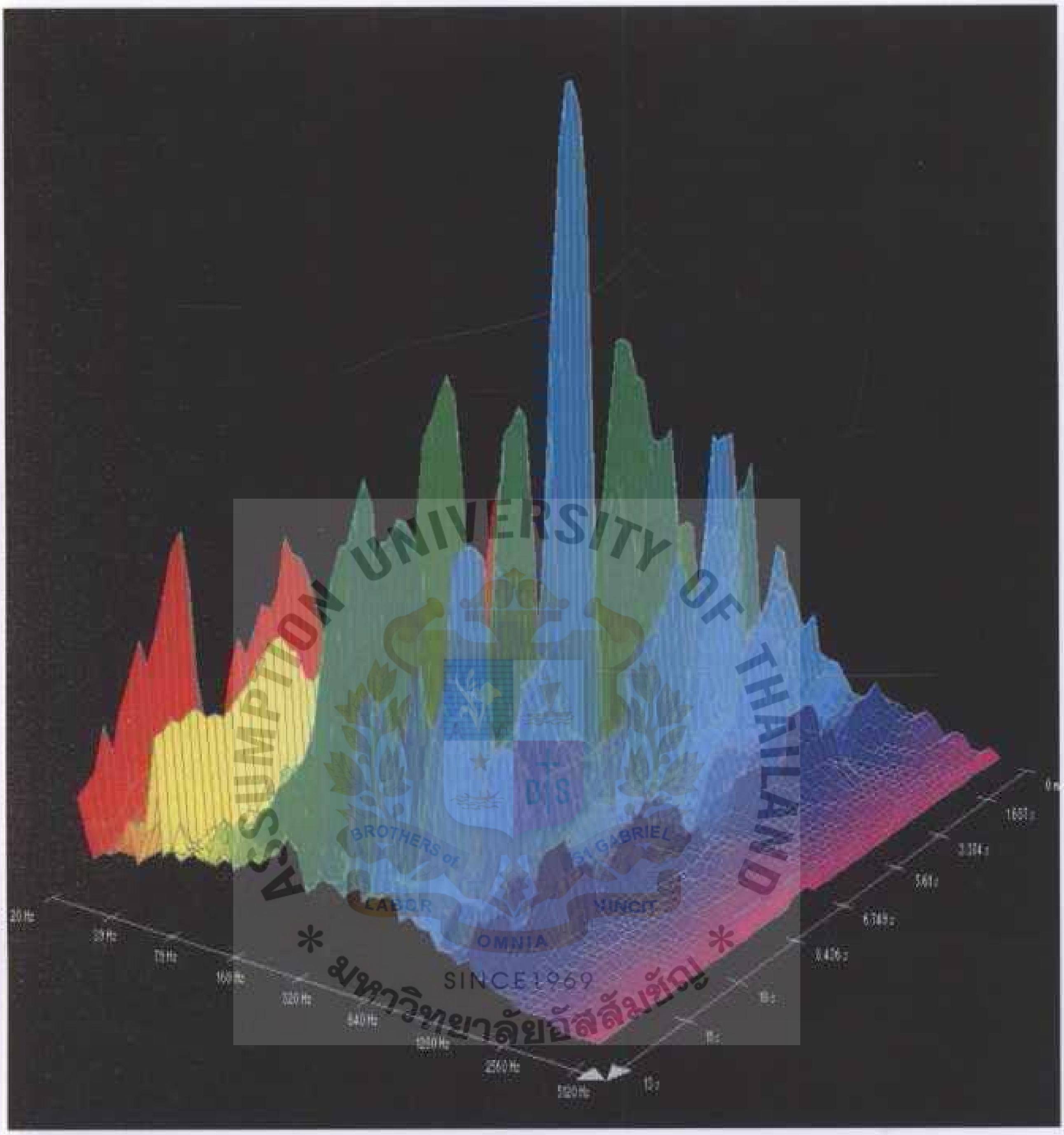

Figure 5.23: 3D Frequency Analysis of G.723.1 Compressed file over 10 Mb/s

In this G.723.1 compressed file graph, we can see that some of the peaks of the low frequencies (red and yellow) are smaller than the original, non compressed file.

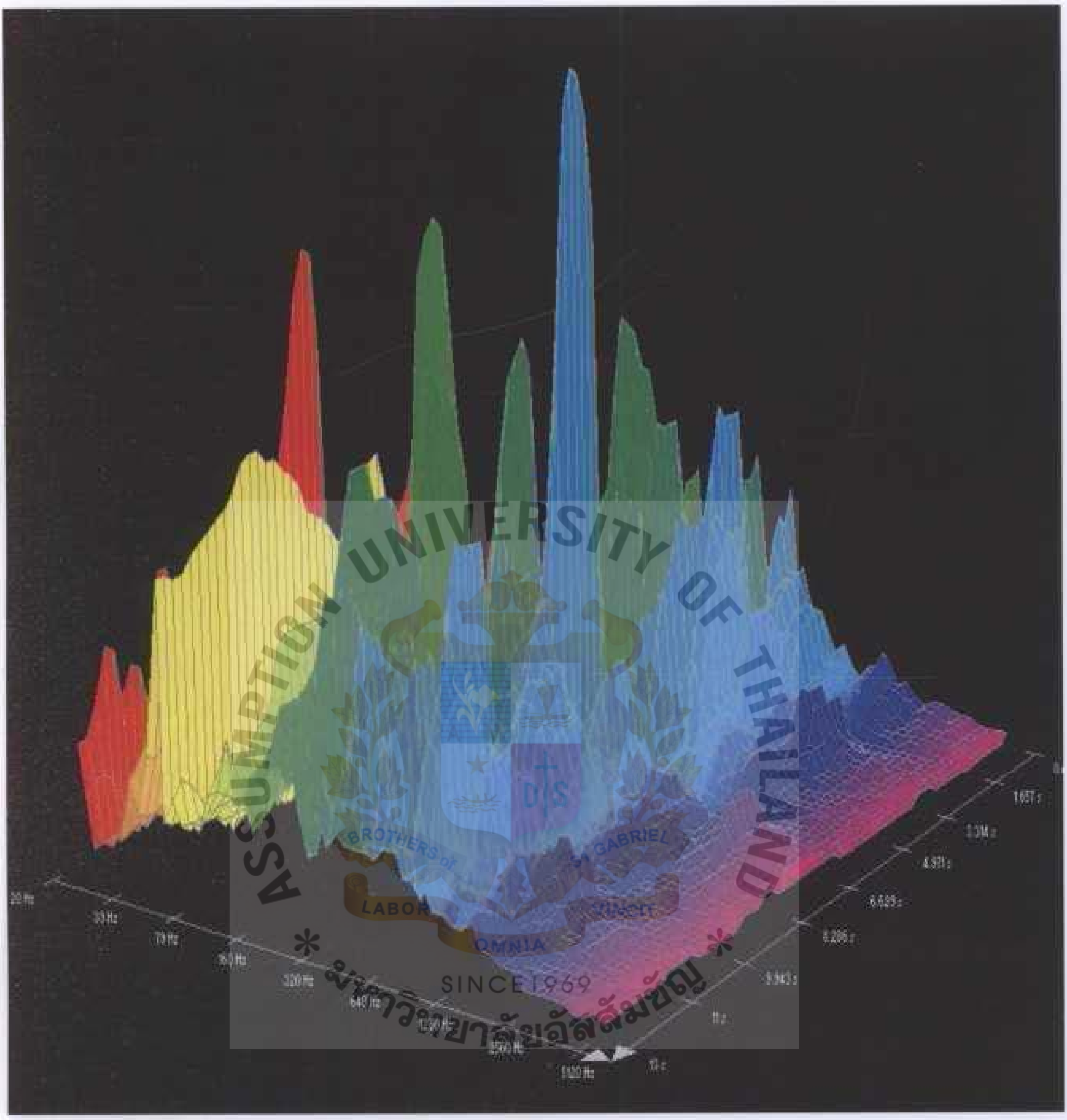

Figure 5.24: 3D Frequency Analysis of G.711A Compressed file over 10 Mb/s

In this G.711A compressed file graph we can see that all peaks of the original, non compressed file are presented and some frequencies (light blue) are even in a higher strength.

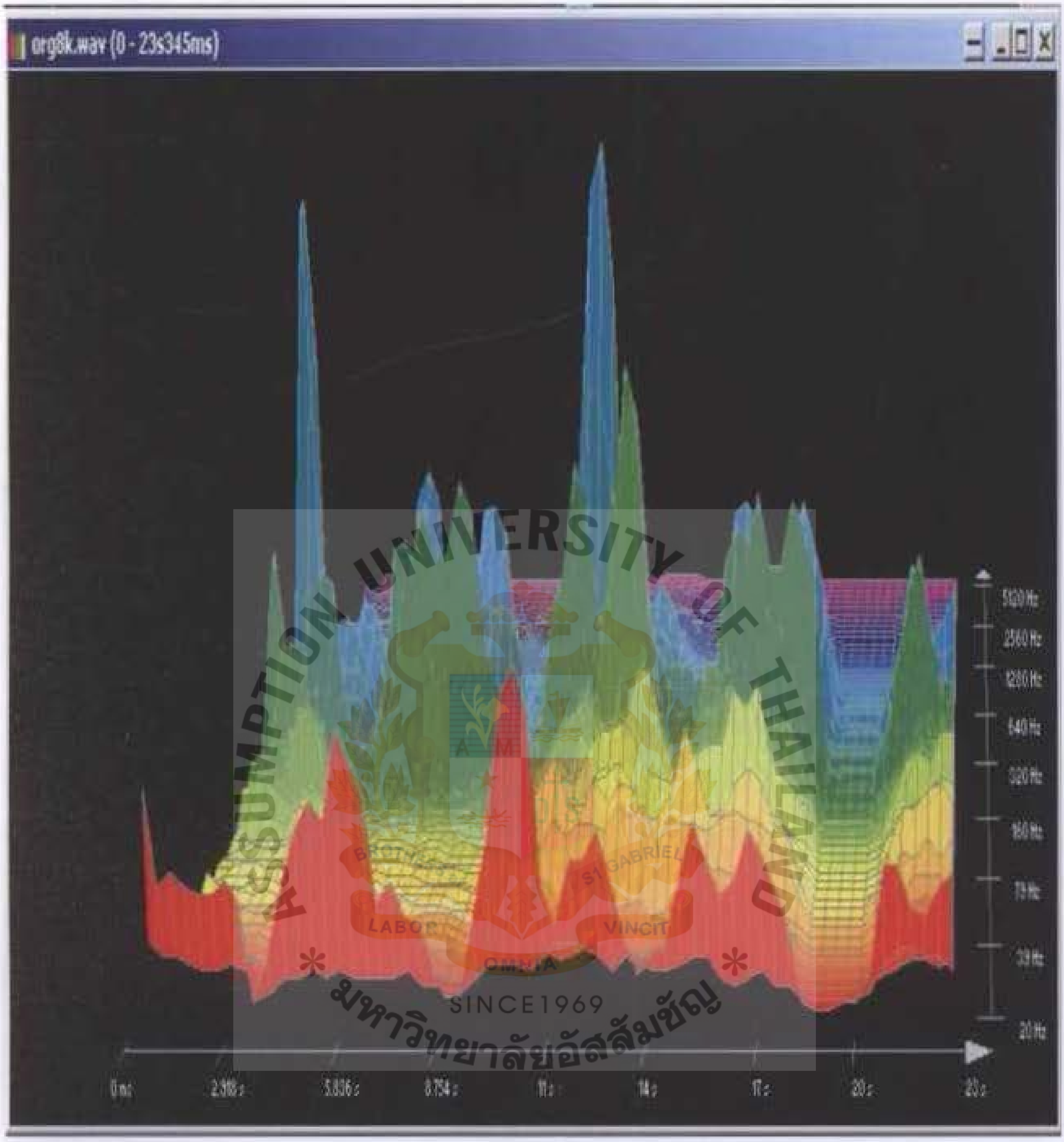

Figure 5.25: 3D Frequency Analysis of Original file over 44 Kb/s in a different perspective

This figure shows us the missing parts caused by delay or lost of packets.

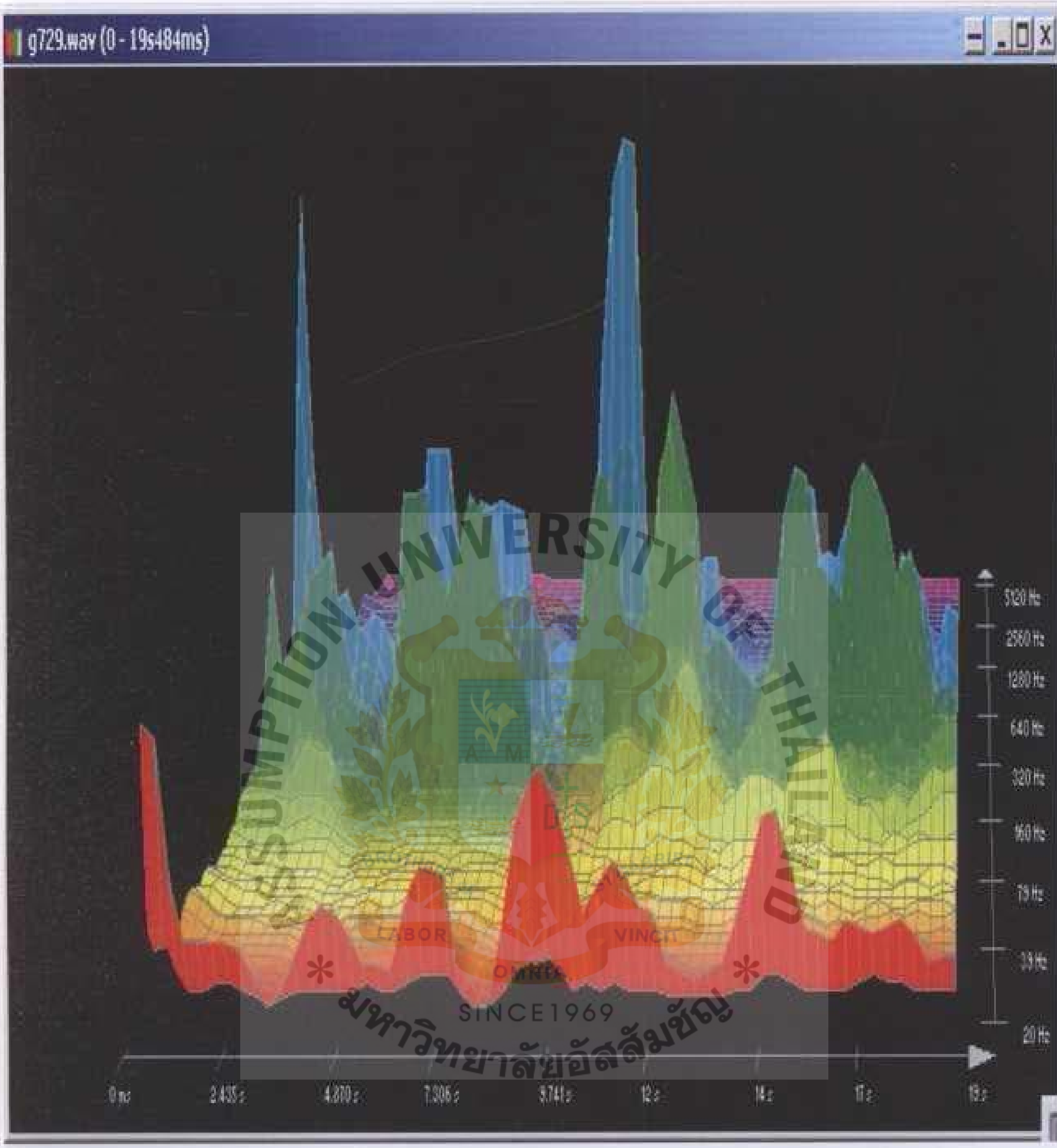

Figure 5.26: 3D Frequency Analysis of G.729 compressed file over 44 Kb/s in a different perspective

A repetition of graph 5.16 in a different perspective, show us silences part in the stream.

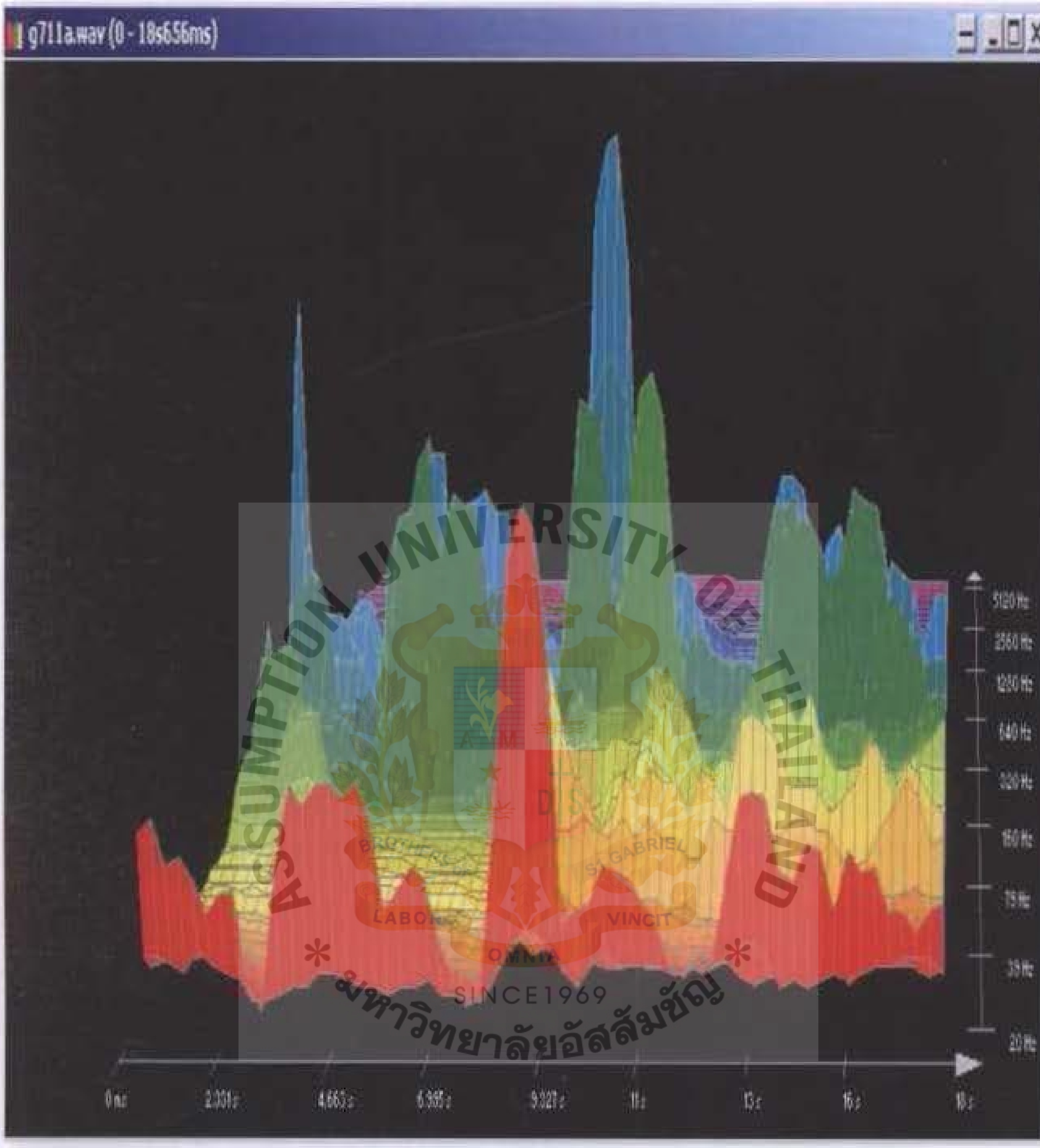

Figure 5.27: 3D Frequency Analysis of G.711 compressed file over 44 Kb/s in a different perspective

A repetition of graph 5.19 in a different perspective, show us silences part in the stream.

From the above graphs we can see that the peeks level - over both 44 Kb/s and 10Mb/s - of the G.711A compressed file are the closest to the one of the original transmitted file. After that G.726 is the second closest and G723.1 is the third. Finally the least closest is the received G.729 file.

5.3.4 Packet Analysis and file size results

The average amount of the packets that has been transmitted and received on the network is shown in Table 5.2.

Type	Transmitter	Receiver
Original file	274	275
G.711A	253	258
G.729	250	251
G.723.1	246	250
G.726	247	251

Table 5.2: Average amount of packets on the network over 44 Kb/s.

We can see that the lowest amount of packet that has been transmitted and received belongs to the G.723.1 voice compression type while the original file has the highest amount of packets. The amount of packet that has been sent or received can give us an idea about the file size.

Table 5.3 shows us the average received size file in both 44 Kb/s and 10 Mb/s. The original transmitted file size was 218 Kb.

	Size over 44 Kb/s (Kb)	Size Over 10 Mb/s (Kb)
Original file	766	440
G.711A	614	440
G.729	641	448
G.723.1	645	448
G.726	693	442

Table 5.3: Averaged Received File size.

The table can easily show us that over fast network the file sizes that have been received with or without voice compression are not much in a different.
Over slower network, we witness bigger differences in the received file size. We can learn from the table that G.711A compressed file yield to the smallest received file size, while G.726 results in the biggest file size among the compressed files. Above all the sizes of the uncompressed, the original file is the biggest.

5.3.5 Subjective Tests results

In order to confirm the result that will be concluded later in this chapter, subjective tests have been conducted by letting 53 different people to listen to the received voice files. The listener had to give scores between 1 and 5 where: Excellent = 5; Good = 4; Fair = 3; Poor = 2; Bad = 1. After the voice file has been evaluate by all listeners, the score attributed to the voice file was averaged out to provide a Mean Opinion Score (MOS).

Table 5.4 shows the MOS / STD / VAR as been collected from the listeners to the different compression methods over 44 Kb/s.

Type	Original	**G.711A**	G.723.1	G.726	G.729
	2	2	2	1	2
	1	2	3	1	2
	2	3	2	1	2
	1	2	2	1	1
	1.5	3.5	1.5	1	1
	2.5	3	2	1	2
	1	2	1	1	2
	1	2	2	1	1
	1	2	2	1	1
	2	2	2	1	2
	2	2	2	1	1
	2	1	1.5	1	2
	1.5	2	1.5	2	1
	1	1	1	1.5	1.5
	1	2	1	1	1
	1	2	2	2	1.5
	1	2	2	1	1
	1	1	2	1	2
	1	1	1	1	1
	1	4	2	2	2
	2	4.5	3.5	1	3
	1	2.5	2	1.5	2
	1	2	2	2	1
	2	2	1.5	1	1
	0.96	1.69	0.81	1.15	0.5
	1.5	2	1.5	1	1
	2	2	2	1	1
	2	3	2	1	2
	2	3	2	2	2
	2	2	2	1	2
	2.5	2	2	1	1
	2	2	1	1	1
	1	1	1	1	1
	1	2	1	1	1
	2	3	2	1	2
	1	3	2	1	1
	1	2.5	2.5	1	1.5
	1	2.5	1	1	1
	2	1.5	3	1	2.5
	1	1	1	1	1
	1	2	1.5	1	1.5

	1	2	2	1	1
	2	3	2	1	2
	1.5	1.5	1.5	1.5	1.5
	2	3	3	1.5	1
	2	4	3	1	2
	1	1	1	1	1
	1	1	1	1	2
	1	1.5	1.5	1	2
	2	3	2	2	2
	2	3	2	1	2
	1.5	2	2	1	1
	1	2.5	2	1	2
MOS	1.4615	**2.1923**	1.8077	1.1538	1.5000
STD	0.50882745	0.8132	0.60378	0.33309	0.54596
VAR	0.26581815	0.67137	0.37741	0.11908	0.30359

Table 5.4: MOS over 44 Kb/s Network.

Table 5.5 shows the MOS / STD / VAR results that have been collected from the listeners to the different compression methods over 10 Mb/s network.

Type	Original	G.711A	G.723.1	G.726	G.729
	4	5	2	3.5	5
	3	4	3	4	3
	4	4	4	5	4
	4	5	3	3	4
	2.5	2.5	3	2.5	2.5
	3.5	5	3	3	3.5
	2.5	4	3	3	3
	3	5	3	3	1
	3	4	4.5	4.5	3
	3	2.5	4	5	5
	4	5	3	3	5
	4	3	4	2	3
	3	4	3	2	3
	3	3	3	3	3
	2.5	4	2.5	2	3.5
	3.5	4	3	2.5	3.5
	3	3	3	3	3
	3	4	4	4	3

	4	3	3	3	3
	3	3	3	2	4
	5	5	5	3	5
	3	4	4	5	5
	3	3	3	3	4
	2	4	3	3	2
	2.54	3.85	1.81	1.57	2.52
	3.5	4	3	3	3.5
	4	5	3	3	4
	4	4	3	3	4
	5	4	4.5	3.5	5
	5	4.5	3.5	3.5	4.5
	3	3	4	3	3
	5	5	3	3	5
	4	4	3	3	4
	3.5	5	3.5	3	3.5
	4	4.5	3	3	4
	4	3	4	3	4
	4	4	3	3	4
	4	4	4	3	4
	3	3	3	3	3
	2.5	3	4	4	2
	3	4	3	3	3
	3	4	3	3	3
	3	3	3	3	3
	3.5	4	3.5	2	3.5
	4	3	3.5	3.5	4
	4	4	4	4	4
	3	3	3	3	3
	4	4	3	3	4
	3	3	3	3	3
	5	3	3	2	5
	4	3	3	3	4
	4.5	5	4	3	5
	4	4	4	2	4
MOS	3.5385	**3.8462**	3.3077	3.0768	3.6325
STD	0.73828	0.75663	0.6068	0.74009	0.88828
VAR	0.66744	0.72484	0.48721	0.62673	0.89674

Table 5.5: MOS over 10 Mb/s Network.

According to table 5.4 and table 5.5 we can see that most users chose to give better scores to the G.711A compressed file, even higher than the original, non compressed file scores.

5.3.6 Result Analysis

In this chapter, we thoroughly examined the performance of different types of voice compression. It was shown through experimentation that when sending voice files over IP networks, compressed files will perform and give better result at the receiving side.

The best voice compression algorithm according to my result is the most simple PCM G.711A compression algorithm standard. The result was checked by subjective test and founded correct according to the listener opinion. The listener gave to the G.711A voice compression algorithm standard the best score even higher than the received non compressed file.

Chapter 6: Discussion and Conclusion

This section is summarized the contributions of this thesis. These contributions are a good indication of how well the objectives set out for this thesis were met. The contributions are:

- An in-depth study of VoIP and related subject. This thesis presented a thorough review of VoIP by combining information from many different sources and presenting it in a clear and efficient manner. The thesis established the current state of research on the subject. Since understanding the problems related to VoIP compression is an important part of any solution, this is an important contribution of this thesis
- A comparative study of voice compression algorithm for VoIP. This study uses known information on several voice compression algorithms to find which one is better suited for VoIP.
- Implementation of an objective measure to evaluate the subjective voice quality. The thesis shows that the existing objective methods to evaluate the voice quality were not sufficient for the evaluation of the different compression algorithms. We introduced a novel objective method to evaluate the subjective voice quality. We compared the result with subjective results from listeners. This is a very important contribution of this thesis. The criteria to judge which type of voice compression gave better results were different on each test. The criteria in the FFT graphs was the louder is the better and the shape of the graph, the more it close to the original file shape, the better the compression type. The criteria in the 3D graphs was the graph with the picks that are similar or the most close to the picks of the original considered to be

the best voice compression technique. The criteria in the file size compression were the smallest the file the better the compression type.

- The result of the subjective test could be different if other listeners or different amount of listeners was conducting the test.

Appendix A: VoIP Transmitting Algorithm

```
BEGIN   /**AVTransmit**/

        BEGIN   /** Input voice locator **/

        INPUT

        parameters: locator (media locator)

        IF      locator = null

                THEN try to locate again

                ELSE locator = media locator

        END IF

        END

        BEGIN   /**start the transmission**/

        parameters: start (error or not)

        IF      transmission started

                THEN start = null

                AND create an RTP session to transmit the voice to the specified IP

                address and port number

                AND stop transmission if other transmission already started

                ELSE start = string with a fail reason

        END IF

                BEGIN   /** try to create a processor to handle the input media

                locator**/

                IF no processor

                        THEN return "could not create processor"

                ELSE get the track from the processor
```

AND set the output to be RAW_RTP

AND get the output data source of the processor

END IF

END

BEGIN /** create session manager for each media track of the processor**/

FOR i=0 TO i=number of trucks

create new session manager, add source details, create RTP session

END FOR

END

END

BEGIN /**the interface of the program that allow the user to operate it**/

DO create an interface which will allow the user to enter the IP address and the port number of the destination

INPUT

parameters: IP (IP address of destination)

port (port of the destination)

source (source of voice to be transmitted)

IF IP=null

THAN error to user

ELSE IF port =null

THAN error to user

ELSE IF source = null

THAN error to user

```
                END IF
            END IF
        END IF
        END
        BEGIN   /** start the transmission for a period of time or until stopped **/
        FOR time=0 TO 30 minutes DO
            IF stop = null
            transmit the voice
            ELSE stop
            END IF
        END FOR
        END
END
```

Appendix B: VoIP Receiving Algorithm

```
BEGIN   /**AVReceive**/

        BEGIN   /** transmitted voice locator **/

        DO open RTP session

        AND check the given session address

        IF fail to open session

                THAN error to user

                ELSE create RTP session in specific buffer size

        END IF

        END

        BEGIN  /** waiting for data **/

        DO wait for data to arrive

        PRINT waiting for data to arrive

        IF no data received

                THAN error to user

                ELSE get host information /**user name**/

                DO start session

        END IF

        END

        BEGIN  /** open GUI for the player**/

        FOR data received <> null

        DO open Java Media Player

        END FOR

        END
```

```
        BEGIN  /** the interface of the program that allow the user to operate it**/
        DO create an interface which will allow the user to enter the IP address and
        the port number of the destination
        INPUT
         parameters: IP (IP address of destination)
                        port (port of the destination)
                        source (source of voice to be transmitted)
        IF IP=null
        THAN error to user
        ELSE IF port =null
                THAN error to user
                ELSE IF source = null
                     THAN error to user
                     END IF
                END IF
        END IF
        END
END
```

References:

[1] Schulzrinne H. "Converging on Internet Telephony: service for telecom version 2", IEEE Internet computing, May/June 1999, pp. 40-43.

[2] Jacobson, V., "VAT manual pages", Lawrence Berkeley Laboratory, Feb. 1992; www.nrg.ee.lbl.gov/vat/.

[3] Hardman, V., Kinstein, P., Sasse, A. and I. Kouvelas, "Robust Audio Tool (RAT)", www.mice.cs.ucl.ac.uk/muldmedia/software/rat/index.htm.

[4] Jiri Kuthan "Architecture of Internet Telephony" www.iptel.org"

[5] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: a transport protocol for real-time applications," Request for Comments 1889, Internet Engineering Task Force, Jan. 1996.

[6] V. Paxson, "End-to-end internet packet dynamics," in SIGCOMM Symposium on Communications Architectures and Protocols, (Cannes, France), Sept. 1997.

[7] V. Paxson, *Measurements and Analysis of End-to-End Internet Dynamics*. PhD thesis, University of California at Berkeley, Berkeley, California, May 1997.

[8] International Telecommunication Union (ITU), "Transmission systems and media, general recommendation on the transmission quality for an entire international telephone connection; one-way transmission time," Recommendation G.114, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, Mar. 1993.

[9] H. Sanneck and N. T. L. Le, "Speech property-based FEC for Internet Telephony applications," in *Proceedings of the SPIE/ACM SIGMM Multimedia Computing and Networking Conference (MMCN)*, (San Jose, California), pp. 38–51, Jan. 2000.

[10] N. S. Jayant, "Effects of packet losses on waveform-coded speech," in *Proceedings of the Fifth International Conference on Computer Communications*, (Atlanta, Georgia), pp. 275–280, IEEE, Oct. 1980. 310

[11] J. Wroclawski, "Specification of the controlled-load network element service," Request for Comments 2211, Internet Engineering Task Force, Sept. 1997.

[12] S. Shenker, C. Partridge, and R. Guerin, "Specification of guaranteed quality of service," Request for Comments 2212, Internet Engineering Task Force, Sept. 1997.

[13] R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource ReSerVation protocol (RSVP) – version 1 functional specification," Request for Comments 2205, Internet Engineering Task Force, Sept. 1997.

[14] R. Braden and L. Zhang, "Resource ReSerVation protocol (RSVP) – version 1 message processing rules," Request for Comments 2209, Internet Engineering Task Force, Sept. 1997.

[15] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated service," Request for Comments 2475, Internet Engineering Task Force, Dec. 1998.

[16] K. Nichols, S. Blake, F. Baker, and D. Black, "Definition of the differentiated services field (DS field) in the IPv4 and IPv6 headers," Request for Comments 2474, Internet Engineering Task Force, Dec. 1998.

[17] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured forwarding PHB group," Request for Comments 2597, Internet Engineering Task Force, June 1999.

[18] V. Jacobson, K. Nichols, and K. Poduri, "An expedited forwarding PHB," Request for Comments 2598, Internet Engineering Task Force, June 1999.

[19] Stallings W. "Data and Computer Communications" 5th edition, Prentice Hall, 1997

[20] http://www.erg.abdn.ac.uk/users/gorry/course/inet-pages/ip-packet.html

[21] http://www.isi.edu/div7/rsvp/overview.html

[22] Schulzrinne H. et al, RFC 1889, "RTP: A Transport Protocol for Real-Time Applications", January 1996.

[23] Luoma M. et al. "Quality of Service for IP Voice Services – Is it necessary?", Internet Routing and Quality of Service, proceedings of SPIE 1998.

[24] Haykin S. "Communication Systems: 3rd Edition", John Wiley & Sons, Toronto, 1994.

[25] http://searchnetworking.techtarget.com/sDefinition/0,,sid7_gci214284,00.html

[26] Bradenburg K. "Perceptual Coding of High Quality Audio", Applications of Digital Signal Processing to Audio and Acoustics, Kluwer Academic Press, 1988.

[27] ITU-T Recommendation G.723.1, 1996

[28] ITU-T Recommendation G.729, 1996

[29] Juang B.H. and Tshuhan C. "The past, Present and Future of Speech Compressing", IEEE Signal Processing Magazine, vol. 15, no. 3, May 1998.

[30] Bosi M. et al "ISO/IEC MPEG-2 Advanced Audio Coding", jornal of the audio engineering society, vol. 45, no. 10, October 1997.

[31] Cox R.V. "Three New Speech Coders from the ITU Cover a Range of Applications", IEEE Communication Magazine, September 1997.

[32] Soulhi S. "Telephony over Packet Networks", IEEE review, 1999

[33] Cole, R.G. and Rosenbluth, J.H., "Voice over IP performance monitoring", April 2001