UNIVERSITÉ DE SHERBROOKE

Faculté de génie
Département de génie électrique et de génie informatique

AMÉLIORATION DE LA ROBUSTESSE DE DÉCODEURS DE PAROLE BASÉS SUR LE MODÈLE CELP EN UTILISANT LES INFORMATIONS RETARDÉES : APPLICATION AU STANDARD G.729 POUR LA VOIX SUR IP

IMPROVING THE ROBUSTNESS OF CELP-LIKE SPEECH DECODERS USING LATE-ARRIVAL PACKETS INFORMATION: APPLICATION TO G.729 STANDARD IN VoIP

Mémoire de maîtrise ès sciences appliquées
Spécialité: génie électrique

Ali KHADRA

Sherbrooke (Québec), Canada Mai 2003

SUMMARY

Voice over Internet applications is the new trend in telecommunications and networking industry today. Packetizing data/voice is done using the Internet protocol (IP). Various codecs exist to convert the raw voice data into packets.

The coded and packetized speech is transmitted over the Internet. At the receiving end some packets are either lost, damaged or arrive late. This is due to constraints such as network delay (jitter), network congestion and network errors. These constraints degrade the quality of speech. Since voice transmission is in real-time, the receiver can not request the retransmission of lost or damaged packets as this will cause more delay. Instead, concealment methods are applied either at the transmitter side (coder-based) or at the receiver side (decoder-based) to replace these lost or late-arrival packets.

This work attempts to implement a novel method for improving the recovery time of concealed speech. The method has already been integrated in a wideband speech coder (AMR-WB) and significantly improved the quality of speech in the presence of jitter in the arrival time of speech frames at the decoder. In this work, the same method will be integrated in a narrowband speech coder (ITU-T G.729) that is widely used in VoIP applications. The ITU-T G.729 coder defines the standards for coding and decoding speech at 8 kb/s using Conjugate Structure Algebraic Code-Excited Linear Prediction (CS-CELP) Algorithm.

SOMMAIRE

L'utilisation de la voix sur Internet est une nouvelle tendance dans le secteur des télécommunications et de la réseautique. La paquetisation des données et de la voix est réalisée en utilisant le protocole Internet (IP). Plusieurs codecs existent pour convertir la voix codée en paquets.

La voix codée est paquetisée et transmise sur Internet. A la réception, certains paquets sont soit perdus, endommagés ou arrivent en retard. Ceci est causé par des contraintes telles que le délai («jitter»), la congestion et les erreurs de réseau. Ces contraintes dégradent la qualité de la voix. Puisque la transmission de la voix est en temps réel, le récepteur ne peut pas demander la retransmission de paquets perdus ou endommagés car ceci va causer plus de délai. Au lieu de cela, des méthodes de récupération des paquets perdus (« concealment ») s'appliquent soit à l'émetteur soit au récepteur pour remplacer les paquets perdus ou endommagés.

Ce projet vise à implémenter une méthode innovatrice pour améliorer le temps de convergence suite à la perte de paquets au récepteur d'une application de Voix sur IP. La méthode a déjà été intégrée dans un codeur large-bande (AMR-WB) et a significativement amélioré la qualité de la voix en présence de «jitter » dans le temps d'arrivée des trames au décodeur. Dans ce projet, la même méthode sera intégrée dans un codeur à bande étroite (ITU-T G.729) qui est largement utilisé dans les applications de voix sur IP. Le codeur ITU-T G.729 défini des standards pour coder et décoder la voix à 8 kb/s en utilisant l'algorithme CS-CELP (Conjugate Structure Algebraic Code-Excited Linear Prediction).

To My Wife Randa

ACKNOWLEDGEMENTS

First of all, I would like to thank my professor Roch Lefebvre for his continuous support and understanding throughout my project work and thesis.

I express my thanks and gratitude to my part-time supervisor Redwan Salami for his support and constructive ideas.

Special thanks to my project manager Philippe Gournay for his support and advice during my project work. I also like to thank my colleague Stephane Ragot for the valuable time and consultations.

I also wish good luck to all my colleagues in the Speech Coding Group of the Department of Electrical and Computer Engineering.

Last, but not least, I thank my wife Randa for her love, care and patience throughout my studies. My special thanks go to my brother Imad and sister Barea for their support and care.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Voice over packet networks combines two different domains, the speech coding domain and the networking domain. First speech coders were designed to be deployed in circuit-switched networks such as the PSTN (Public Switched Telephony Network) and later in mobile telecommunications. In the modern telecommunications world, the recent trend is to replace circuit switched networks, such as the PSTN, with packet-switched networks like the Internet. Researchers focused on the improvement of speech quality in the transmission of voice over an IP network, which is impaired by packet delay, packet loss and delay jitter in an IP network.

The quality of real-time voice transmission over the Internet is not satisfactory because of the current Internet's delivery and scheduling mechanisms. The Internet has been traditionally designed to support non real-time data communications, but not real-time voice transmission. These real-time applications have quite different characteristics and requirements.

VoIP (Voice over IP) provides several advantages over circuit-switched voice. Bandwidth required for a voice call can be significantly reduced through voice activity detection (VAD) techniques and use of low bit-rate codecs. VAD removes silence, which accounts for as much as 40 percent (some studies claim up to 56 percent) of the voice information that is transmitted. Low bit-rate codecs reduce the amount of bandwidth for a voice call from 64 kbps to as little as 8 kbps or 4 kbps.

According to a study conducted by VDC (Venture Development Corporation), the market for VoIP networks has reached $956.5 million, and is expected to witness an average annual growth rate of 12.6% on its way to being over $1.7 billion by 2005. Check Figure 1.1 below.

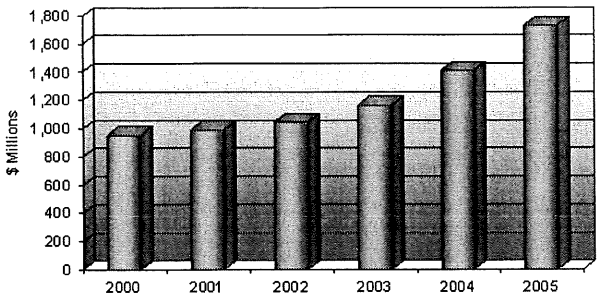


Figure 1.1 VoIP market growth

In the following we will talk about the advantages and limitations of speech coders and the packet network speech coding-related features.

1.1 Advantages and limitations of speech coders:

The main function of speech coders is to convert the analog human speech into a digital form. The more compressed the digital speech, the more easily it can be transmitted over a network or the less storage space it needs. The level of complexity and processing delay of the digitized speech signal must be as small as possible.

The speech coding is a process of digitizing and compression that is limited by a number of constraints. Codec definition involves a certain compromise. For instance, the output signal of a low-rate speech coder has a somewhat degraded quality. The degradation is due to the loss of certain signal information. Usually, subjective tests are conducted to determine an acceptable output signal.

There are four basic speech coding parameters, the signal quality, the bit-rate, the processing time (delay) and the coder complexity. Speech coders are divided into high-bit

rate (also known as high-rate), such as PCM (Pulse Code Modulation) and low-bit rate, such as G.729. High-rate coders have low complexity and introduce little delay, whereas, low-rate coders require excessive computations and cause significantly more processing delay. The lower the rate of a coder, the less bandwidth is required during transmission. As far as error concealment is concerned, high-rate coders produce an output signal that can not be efficiently concealed. On the other hand, better concealment of the output signal is achieved with-rate coders.

1.2 Advantages and limitations of packet networks:

There are various types of packet networks including corporate WANs, enterprise LANs, and the Internet. These packet networks use a variety of transmission protocols and physical layers including:

- Internet Protocol (IP)
- Asynchronous Transfer Mode (ATM)
- Frame Relay (FR)
- Ethernet

Packets sometimes encapsulate other types of packets, creating "packets within packets". Examples of such layered packetization formats include IP over ATM, and IP over Ethernet.

Packet networks have many advantages over the PSTN for moving data, voice, and video traffic. Data, voice, and video in packet format are often compressed. For example, compressed voice can take as little as 1/10 of the bandwidth required for normal PCM voice signals. This allows many more voice channels to be carried over a given bandwidth.

Voice over packet allows voice traffic to bypass long-distance toll charges. This application of voice over packet is usually referred to as toll-bypass. Recently deployed networks have shown that packet voice networks can be implemented for between 10% and 20% of the cost to deploy an equivalent capacity PSTN network [12].

One of the most important packet networks limitations is that too many sent or received packets can lead to congestion. Packets that cannot be stored or delivered might be discarded by routers and switches.

Another limitation is due to the fact that packets can arrive at different times and in a different order than when they were sent. This is a problem for telephone conversations.

1.3 Using network knowledge to optimize speech decoders:

The network knowledge is essential for improving the Quality of Service (QoS) and optimizing speech decoders. Network parameters that affect VoIP applications can be sent to the decoder via a special dedicated feedback link. This helps decoders take proper action to resolve network-related transfer delay (Latency), delay variation (Jitter), congestion, packet loss, packet out-of-order and available bandwidth problems before they become potential QoS issues.

1.4 Overview of the thesis:

The main aim of this work is to improve packet loss recovery using late frames. Late frames can play an important role in increasing the robustness of the decoder without affecting the overall end-to-end delay. The thesis is organized as follows. Chapter 1 is an introduction to the thesis. In Chapter 2, we talk about the Quality of Service (QoS) issues that must be taken into consideration in any Voice over IP (VoIP) application. Chapter 3 talks about the network protocols for VoIP. Chapter 4 is an overview of the G.729 standard. Chapter 5 discusses the principles and the implementation of late-arrival frames technique. Chapter 6 is dedicated to presenting the subjective and objective results of the implementation. Finally, the thesis ends with a conclusion.

CHAPTER 2

QUALITY OF SERVICE (QoS) ISSUES

2.1 Introduction:

Human speech is burdened with a tremendous amount of redundant information that is necessary for communications to occur in the natural environment, but which is not needed for a conversation to occur over a communications network. Analysis of a representative voice sample shows that only 22 percent of a typical conversation consists of essential speech components that need to be transmitted for complete voice clarity. The balance is made up of pauses, background noise, and repetitive patterns [15].

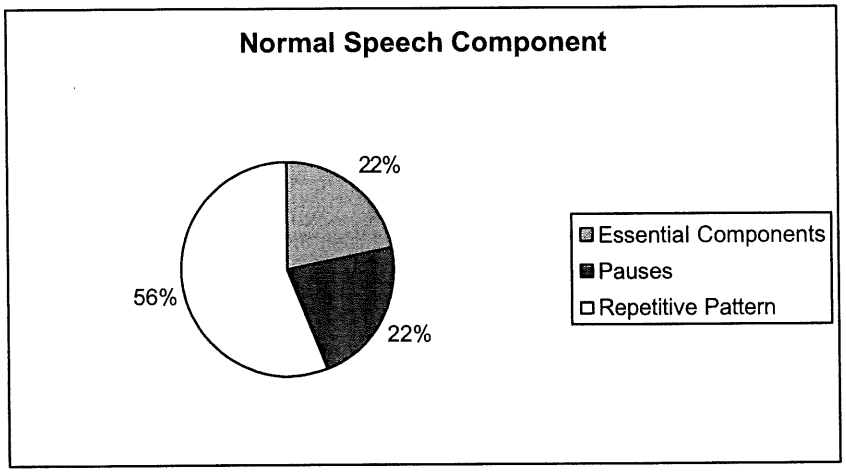


Figure 2.1 Normal speech component chart

2.2 VoIP processing:

Packetized voice is possible and low-bit rates are attained by analyzing and processing only the essential components of the voice sample, rather than attempting to digitize the entire voice sample (with all the associates pauses and repetitive patterns). Current speech processing technology takes the voice digitizing process several steps further than conventional encoding methods.

2.2.1 Voice Compression:

Compression of voice results from the removal of silent periods and redundant information found in human speech. Voice compression is used to reduce the amount of information needed to recreate the voice at the destination end. Uncompressed digitized voice requires a large amount of bandwidth. This often makes it impractical to transmit these signals over low-speed access links. The use of low-bit rate voice compression algorithms can make it possible to provide high quality speech while using bandwidth efficiently.

Voice compression algorithms can be classified into Waveform coders, Vocoders and Hybrid coders. A well-known example of a waveform coder is PCM (Pulse Code Modulation). PCM was the original standard (G.711) for 64 Kbps digital transmission. It's a simple speech coding technique that uses a high bit-rate to generate toll quality speech. PCM is near ubiquitous in circuit switched telephone networks and is the standard by which other voice compression techniques are measured.

In vocoders, not the signal samples but the parameters of a source filter speech model are quantized and transmitted. This source-filter synthesis representation closely follows the model of speech production.

This intermediate class between waveform coders and vocoders are dominating the state-of-the-art solutions of speech coders for medium bit rates and a high quality, with applications particularly in digital wireline and mobile communication systems. The majority of modern hybrid speech coders are based on the principle of linear-predictive analysis-by-synthesis coding also known as CELP (Code-Excited Linear Prediction). In chapter 4, we will talk about G.729, which is an ITU-T speech coding algorithm and standard. G.729 is based on CELP predictive coding. Predictive coders main disadvantage is that when faced with channel impairments (late or lost packets or bit errors), the predictor memories at the decoder diverge from the predictor memories at the encoder leading to error propagation due to encoder/decoder de-synchronization This reduces the voice quality above a given Frame Error Rate (FER).

Later in this thesis we will introduce a technique to improve voice quality by minimizing error propagation using information carried in late-arrival frames, which are usually considered as lost.

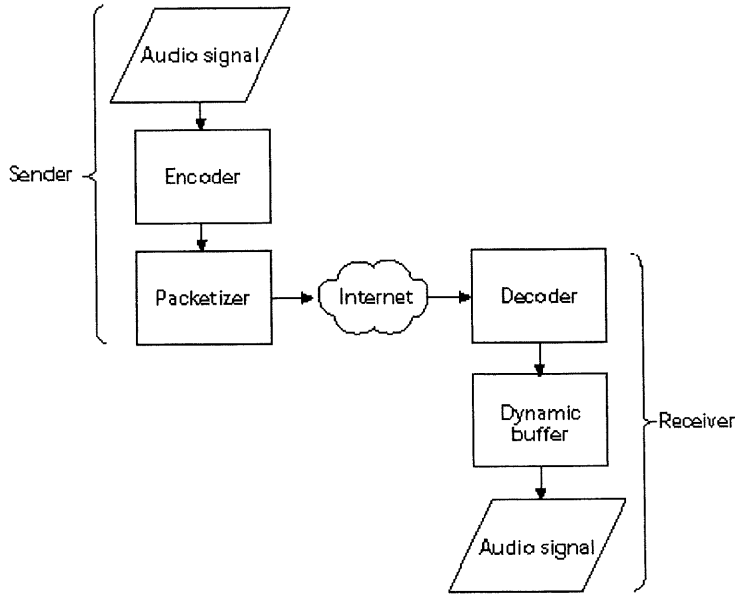


Figure 2.2 VoIP processing at the sender and the receiver

2.2.2 Silence suppression:

A person speaking does not provide a continuous stream of information (regardless of how fast they talk). Pauses between words and sentences, and those gaps that come at the end of one person talking but before the other begins, also can be removed. The pauses may be represented in compressed form and they can be re-created at the destination side of the call to maintain the natural quality of the spoken communication. The suppression and removal of silent periods can also significantly improve bandwidth utilization.

2.2.3 Voice frame formation:

The removal of silent periods and redundant information through advanced techniques enables voice to be efficiently "compressed". After the removal of repetitive patterns and

silent periods, the remaining speech information may then be digitized and placed into voice packets suitable for transmission over packet networks These packets or frames (both terms are often used interchangeably) also tend to be smaller than average data frames. The use of smaller packets helps to reduce transmission delay across an IP network. The concepts introduced above provide the basis for voice to efficiently use the smallest amount of bandwidth possible for transmission over an IP network.

The general function of these strategies is to scrutinize the speech signal more carefully, to eliminate the redundancies in the signal more completely, and to use the available bits to code the non-redundant parts of the signal in an efficient manner. As the available bit rate is reduced from 64 Kbps to 32, 16, 8, and 4 Kbps or below, the strategies for redundancy removal and bit allocation need to be ever more sophisticated. Low cost general-purpose DSP (Digital Signal Processing) processors and other advanced compression algorithms allow the possibility of accomplishing voice compression within VoIP capable devices at lower and lower bit rates.

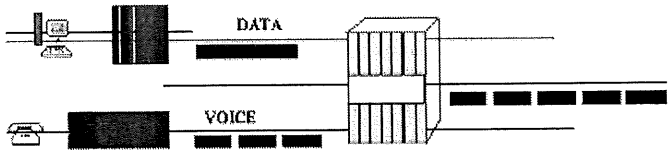


Figure 2.3 Voice frame formation [15]

2.2.4 Echo Cancellation:

Echo is a phenomenon found in voice networks. Echo occurs when the transmitted voice is reflected back to the point from which it was transmitted. In voice networks, echo cancellation devices are used within a carrier's network when the propagation delay increases to the point where echo results. The longer the distance, the more delay, and the more likely echo will result. Voice transmitted over a packet network will also face propagation delays. As the end-to-end delay increases the echo would become noticeable

to the end-user if it is not canceled. Since IP carriers do not use echo cancellation equipment in their networks, it is up to the equipment vendor to address echo cancellation in the equipment.

2.2.5 Delay and Delay Variation (Jitter):

The bursty nature and variable frame sizes of IP networks may result in variable delays between consecutive packets. The variation in the time difference between each arriving packet is called "jitter". Jitter can impede the ability of the receiving-end customer premise equipment to smoothly regenerate voice. Since voice is inherently a continuous wave form, a large gap between the regenerated voice packets will result in a distorted sound. To avoid dropping frames, data can be buffered at the speech decoder sufficiently to account for the worst case delay jitter through the network.

2.2.6 Frame Loss:

Voice over IP can usually withstand infrequent packet loss better than data. If a voice over IP packet is lost, the user will most likely not notice. If excessive frame loss occurs, it is equally unacceptable for voice over IP and data traffic. Error concealment techniques play an important role in solving the frame loss impairments.

2.2.7 Prioritization:

Voice, fax and some data types are delay sensitive. This means that if the end-to-end delay, or delay variation exceeds a specified limit, the service level will degrade. To minimize voice traffic delay, a prioritization mechanism that provides service to the delay sensitive traffic first can be employed. Vendors offering equipment capable of integrating voice over IP may choose to use a variety of proprietary mechanisms to ensure a balance between voice and data transmission needs. Although they may differ, the concept remains essentially the same. For example, each input traffic type may be configured into one of several priority queues. Voice traffic can be placed in the highest-priority queue, for expeditious delivery to the network. Lower-priority data traffic can be buffered until the higher-priority voice packets are sent.

2.2.8 Fragmentation:

Fragmentation is used to break up larger blocks of data into smaller, less delay-creating frames. This is another means used to ensure the highest voice quality level possible. Fragmentation attempts to ensure an even flow of voice frames into the network, minimizing delay. The fragmentation often involves all of the data in the network to retain consistent voice quality. This is because even if the voice information is fragmented, delay will still occur if a voice frame is held up in the "middle" of the network behind a large data frame. This fragmentation of data packets assures voice packets are not unacceptably delayed behind large data packets. Additionally, fragmentation reduces jitter because voice packets can be sent and received more regularly. Fragmentation, especially when used with prioritization techniques, is used to ensure a consistent flow of voice information. The objective of this and other techniques is to enable VoIP technology to provide service approaching toll voice quality.

CHAPTER 3

NETWORK PROTOCOLS FOR VoIP

3.1 Introduction:

In this chapter we will talk about the different network protocols used for Voice over IP (VoIP). These protocols are needed for connection setup, signaling and data transmission. The Internet has much evolved by the introduction of real-time protocols that were designed to serve the wide-spreading real-time applications such as iPhones (Internet Phones for Voice or Video calls), and the real-time broadcasting (streaming) of radio and TV stations over the Internet. These real-time applications have quite different characteristics and requirements.

The first significant characteristic of real-time applications is high delay sensitivity. The flow of packets on the Internet does not always take the same or shortest path. Sometimes packets belonging to the same connection take a different path due to network congestion. This ultimately causes a high delay and as a result packets arrive at their destination with a high delay. If the packets arrive after a certain time limit, they are automatically considered as lost. Retransmission is not possible as it would cause even more delay.

The second significant characteristic is that most real-time applications do not require received data to be hundred percent precise, unlike non real-time applications which require data be delivered correctly and reliably all the time. This characteristic is very useful because the receiver can tolerated certain level of loss of data with a significant degradation in performance.

The above two characteristics define the potential problems that should be considered in order to develop a high quality, real-time voice transmission application.

3.2 Networking and the Internet:

The Internet is a vast internetworking of computer networks operating using the so-called the Internet Protocol (IP). These computer networks are mostly heterogeneous. A good question arises here on how are these heterogeneous computer networks connect? The answer is simply using a modular open system generally known as the OSI (Open System Inter-connection) Reference Model [17].

3.3 The OSI Reference Model:

The OSI Model is based on a proposal developed by the International Standards Organization (ISO) as a first step towards an international standardization of the protocols used in the various layers. The Model is called OSI (Open Systems Interconnection) Reference Model.

The OSI Model has seven layers. Each of the seven OSI layers perform a well defined function. The function of each layer is to provide services to the layer above it. In this case layer n is called the service provider and layer n+1 is called the service user.

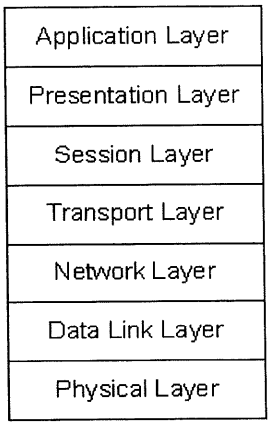


Figure 3.1 The OSI Model

The following is a description of each layer in the OSI Model:

3.3.1 The Physical Layer:

Layer one conveys the bits (1 and 0) which can be an electrical impulse, a light or a radio signal through the network at the electrical and mechanical level. It provides the hardware means of sending and receiving data on a carrier, including defining cables, cards and physical aspects. Fast Ethernet , RS232, and ATM are protocols with physical layer components.

3.3.2 The Data Link Layer:

At layer two, data frames are encoded and decoded into bits. It furnishes transmission protocol knowledge and management and handles errors in the physical layer, flow control and frame synchronization. The data link layer is divided into two sub-layers: The Media Access Control (MAC) layer and the Logical Link Control (LLC) layer. The MAC sub-layer controls how a computer on the network gains access to the data and permission to transmit it. The LLC layer controls frame synchronization, flow control and error checking.

3.3.3 The Network Layer:

Layer three provides switching and routing technologies, creating logical paths, known as virtual circuits, for transmitting data packets from node to node. Routing and forwarding are functions of this layer, as well as addressing, internetworking, error handling, congestion control and packet sequencing.

3.3.4 The Transport Layer:

Layer four provides transparent transfer of data between end systems, or hosts. and is responsible for end-to-end error recovery and flow control. The transport layer creates a distinct network connection for each transport connection required by the session layer. If the transport connection requires a high throughput, however, the transport layer might create multiple network connections, dividing the data among the network connections to improve throughput. It ensures complete data transfer.

3.3.5 The Session Layer:

Layer five establishes, manages and terminates connections between applications. The session layer sets up, coordinates, and terminates conversations, exchanges, and dialogues between the applications at each end. It deals with session and connection coordination.

3.3.6 The Presentation Layer:

Layer six provides independence from differences in data representation (e.g., encryption) by translating from application to network format, and vice versa. The presentation layer works to transform data into the form that the application layer can accept. This layer formats and encrypts data to be sent across a network, providing freedom from compatibility problems.

3.3.7 The Application Layer:

Layer seven supports application and end-user processes. Communication partners are identified, quality of service is identified, user authentication and privacy are considered, and any constraints on data syntax are identified. Everything at this layer is application-specific. This layer provides application services for file transfers, e-mail, and other network software services.

3.4 The TCP/IP Reference Model:

Let us now turn the OSI Reference Model to the Reference Model used for the Internet. The architecture is widely known as TCP/IP Reference Model, after its two primary protocols. The Internet infrastructure was designed to be flexible, scalable and robust as much as possible. Flexibility was needed since applications with divergent requirements were envisioned. Scalability would insure the future expansion of the Internet network. Robustness to sudden failure insures that the network be able to survive loss of subnet with existing connections not being broken off. In other words, the connections remain intact as long as the source and destination machines were functioning.

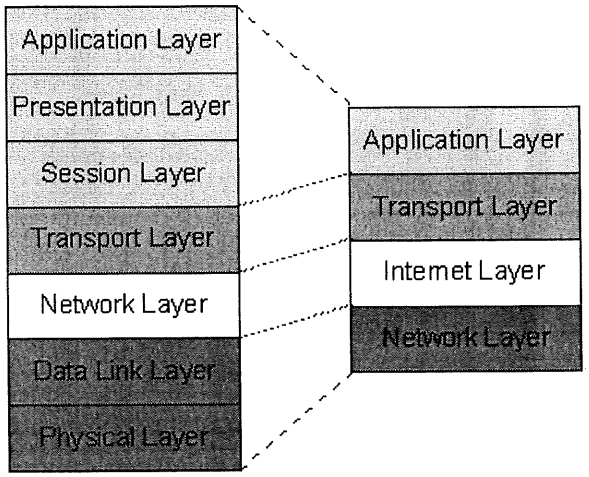


Figure 3.2 Relation between OSI and TCP/IP models

The TCP/IP Reference Model consists of four layers only. The Session and the Presentation layers of the OSI Reference Model were omitted, the Physical and the Data Link layers were joined together forming one layer called the Network layer.

3.4.1 The Network Layer:

Layer one is the lowest layer of the TCP/IP protocol hierarchy. Protocols in this layer define how to use a specific network to transmit an IP datagram. Protocols in the Network Layer must know the details of the underlying network (packet structure, packet length, addressing, etc.) to correctly format the data being transmitted over the network. For new hardware technologies, new Network Layer protocols must be developed. Because of this, there are many protocols, one for each physical network standard. Functions performed at this level include encapsulation of IP datagrams into frames for transmission over the network and mapping of IP addresses to physical network addresses.

3.4.2 The Internet Layer:

Layer two is the linchpin that holds the whole architecture together. This defines one protocol that is called the *Internet Protocol* (IP). The job of the network layer is to deliver IP packets to their destinations. Clearly, the main job of this layer is to route the packets to their destinations while avoiding network congestion. Hence, addressing is a useful concept here. Each IP packet is designated to travel to a host computer, so IP addresses are assigned to each host that participates on the network.

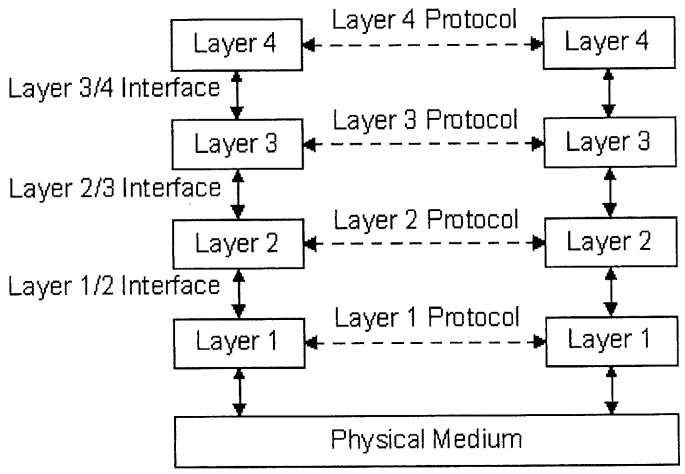


Figure 3.3 The TCP/IP Architecture

3.4.3 The Transport Layer:

Layer three has been designed to allow peer entities to carry out a conversation. There are two end-to-end protocols defined here: *Transmission Control Protocol* (TCP) and *User Datagram Protocol* (UDP). The former is a protocol that provides guaranteed and connection-oriented services to the layer above. In other words, TCP ensures that a stream of data is delivered in order and without error from its source to its destination on the network. The connection is only released when the communication process of sending and receiving is complete. This flow of data is bi-directional.

UDP on the other hand, provides best-effort and connectionless services to applications that do not want to use TCP. A best-effort service suggests that this protocol will attempt,

with an affordable effort, to provide a service that is acceptable but delivery is not guaranteed. So in some sense, UDP is unreliable although its service is faster than TCP service. Moreover, the connectionless service that UDP provides will route each message independent of the rest. Unlike a connection-oriented service, two packets that were sent to the same destination may not arrive in the same order as they were sent.

TCP allows fragmentation of data into discrete messages before passing them to the network layer. Symmetrically, defragmentation of data on the receiving end would be in order. TCP also handles flow control to ensure that a fast sender will not choke a slow receiver with more messages than it can cope. UDP, on the other hand, does not provide such services. It is however, widely used by applications for one-shot deliveries where speed is more important than accuracy.

3.4.4 The Application Layer:

Layer four contains all the higher-level protocols. The early protocols included the virtual terminal (Telnet), the file transfer protocol (FTP), the electronic mail (SMTP), etc. The virtual terminal protocol allows a user on one machine to log into a distant machine and work there. The file transfer protocol provides a way to move data efficiently from one machine to another. Electronic mail was developed for exchanging e-mails. Many other protocols have been added to these over the years. Among the added protocols we find the Domain Name Service (DNS) used for mapping host names onto their network addresses, the Network News Transport Protocol (NNTP) used for moving news articles around, the HyperText Transfer Protocol (HTTP) used for fetching pages on the World - Wide Web, and many others.

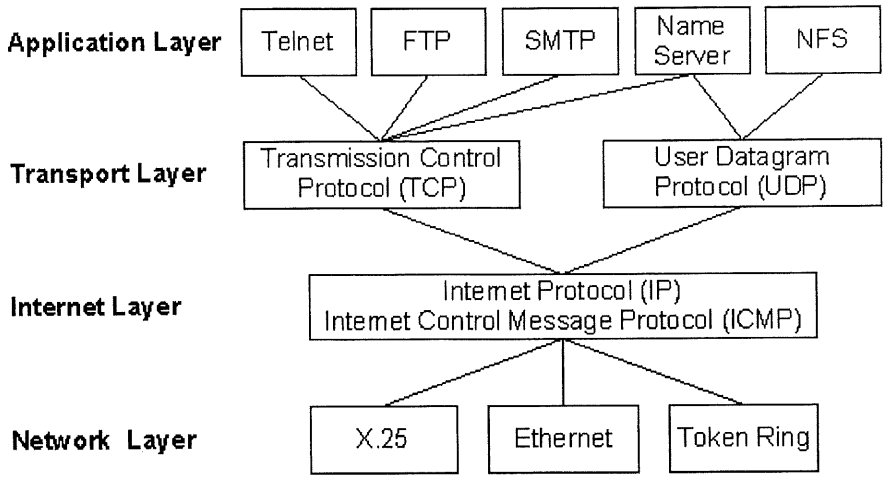


Figure 3.4 The TCP/IP Protocol Stack

3.5 Communications Protocols:

A Protocol is a set of rules governing the format and meaning of the frames, packets, or messages that are exchanged by the peer entities within a layer.

The IP Protocol:

The IP is the Protocol of the Network Layer of the OSI and TCP/IP Models. Unlike most of the protocols, it was designed from the beginning with internetworking in mind. There are two main versions of this protocol, that is IPv4 and IPv6, which is also known as IP-NG (Next Generation).

3.5.1 IPv4:

An IP Datagram consists of a header and a data field. The header has a 20-byte fixed part and a variable-length data part. The header format is shown in Figure. It is transmitted in big endian order: from left to right, with high-order bit of the Version going first.

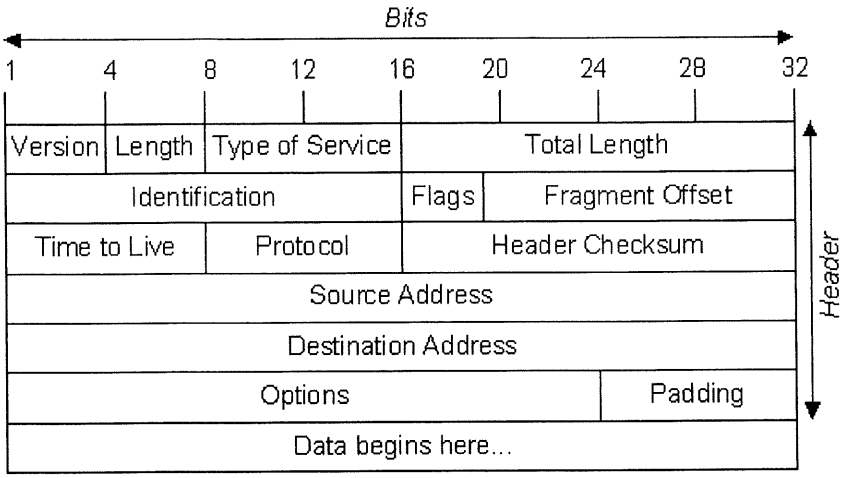


Figure 3.5 IPv4 header format

Version: 4 bits

The Version field indicates the format of the internet header.

Length or IHL: 4 bits

Internet Header Length (IHL) is the length of the internet header in 32 bit words, and thus points to the beginning of the data. Note that the minimum value for a correct header is 5.

Type of Service: 8 bits

Flags: 3 bits This field allows the host to tell the subnet what kind of service it wants. Various combinations of reliability and speed are possible. The major choice is a three way tradeoff between low-delay, high-reliability, and high-throughput. For VoIP, fast delivery bearts accurate delivery.

Various control flags:

Bit 0: reserved, must be zero

Bit 1: (DF) 0 = May Fragment, 1 = Don't Fragment.

Bit 2: (MF) 0 = Last Fragment, 1 = More Fragments.

Identification: 16 bits

An identifying value assigned by the sender to aid in assembling the fragments of a datagram.

Fragment Offset: 13 bits

This field indicates where in the datagram this fragment belongs. The fragment offset is measured in units of 8 octets (64 bits). The first fragment has offset zero.

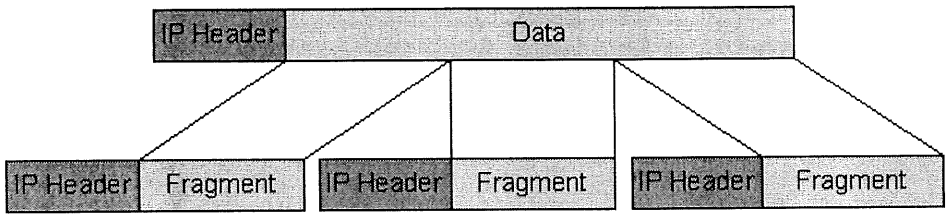


Figure 3.6 Fragmentation of an IP datagram

Time to Live: 8 bits

This field indicates the maximum time the datagram is allowed to remain in the internet system. If this field contains the value zero, then the datagram must be destroyed. This field is modified in internet header processing. The time is measured in units of seconds, but since every module that processes a datagram must decrease the TTL by at least one even if it processes the datagram in less than a second, the TTL must be thought of only as an upper bound on the time a datagram may exist. The intention is to cause undeliverable datagrams to be discarded, and to bound the maximum datagram lifetime.

Protocol: 8 bits

This field indicates the next level protocol used in the data portion of the internet datagram.

Header Checksum: 16 bits

A checksum on the header only. Since some header fields change (e.g., time to live), this is recomputed and verified at each point that the internet header is processed.

Source Address: 32 bits

The source address.

Destination Address: 32 bits

The destination address.

Options: variable

The options may appear or not in datagrams. They must be implemented by all IP modules (host and gateways). What is optional is their transmission in any particular datagram, not their implementation. Currently five options are defined:

Security:	specifies how secret the datagram is
Strict source routing:	gives the complete path to be followed
Loose source routing:	gives a list of routers not to be missed
Record route:	makes each router append its IP address
Timestamp:	makes each router append its address and timestamp

Padding: variable

The internet header padding is used to ensure that the internet header ends on a 32 bit boundary. The padding is zero.

3.5.2 IPv6:

IPv6 or IPng is an enhanced version of IPv4. With the explosion of interest in the Internet and the fact the IP addresses are running out, it was about time that a new version of the IP be developed.

In 1990, the IETF started working on a new version of IP, one which would never run out of address, would solve a variety of other problems, and be more flexible and efficient as well. Some of these new features are specially important for VoIP:

- Reduced size of routing tables
- Simplified protocol to allow routers to process packets faster
- Better security (authentication and privacy)
- Special attention to type of service, particularly for real-time data

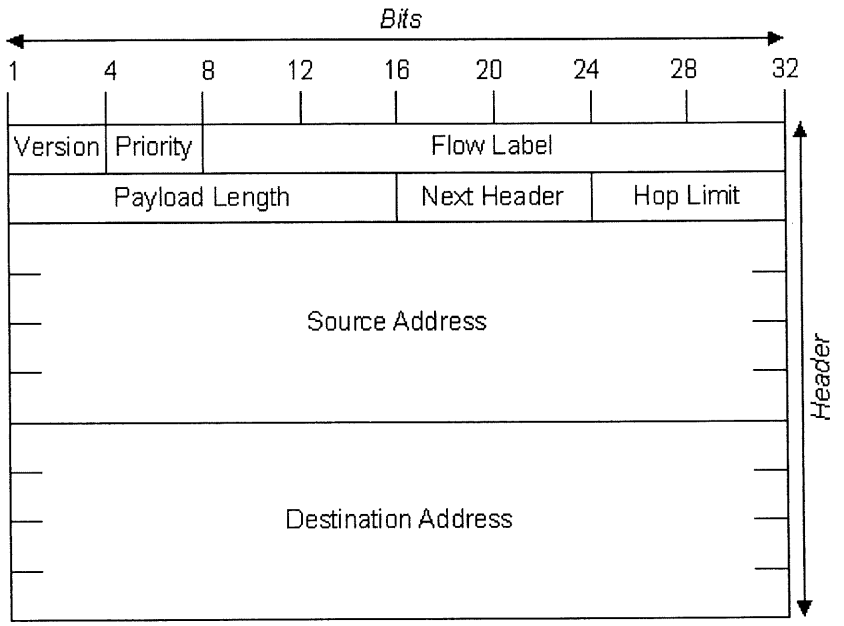


Figure 3.7 IPv6 header format

Version:

4 bits. IPv6 version number.

Traffic Class:

8 bits. Internet traffic priority delivery value.

Flow Label:

20 bits.Used for specifying special router handling from source to destination(s) for a sequence of packets.

Payload Length:

16 bits, unsigned. Specifies the length of the data in the packet. When set to zero, the option is a hop-by-hop Jumbo payload.

Next Header:

8 bits. Specifies the next encapsulated protocol. The values are compatible with those specified for the IPv4 protocol field.

Hop Limit:

8 bits, unsigned. For each router that forwards the packet, the hop limit is decremented by 1. When the hop limit field reaches zero, the packet is discarded. This replaces the TTL field in the IPv4 header that was originally intended to be used as a time based hop limit.

Source address:

The IPv6 address of the sending node (16 bytes)

Destination address:

The IPv6 address of the destination node (16 bytes)

IPv6 extension headers:

Some of the missing fields are occasionally still needed so IPv6 has introduced the concept of an optional Extension Header. These headers can be supplied to provide extra information. If more than one header is provided, then they must appear directly after the fixed header, and preferably in the order listed below:

- Hop-by-Hop Options Header
- Routing Header
- Fragment Header
- Authentication Header
- Encrypted Security Payload
- Destination Options Header

IPv6 major renovations:

- Simpler header format
- Flow labeling
- The support for extensions and options has been improved
- Authentication and Security Extensions
- The size of the IP address is increased to128 bits
- Simpler auto-configuration of IP addresses
- Multicast routing has been improved by adding a scope field to the multicast addresses
- Anycast addressing has been added.

Comparison of IPv6 and IPv4 headers:

IPv6 header has 6 fields plus Source and Destination Addresses IPv4 header has 10 fields, Source and Destination Addresses plus options

Fields deleted from IPv6:

- Header length
- Type of service
- Identification, flags, fragment offsets
- Header checksum

Fields added to IPv6:

- Priority
- Flow label

Fields renamed:

- Length became Payload length
- Protocol became Next header (UDP, TCP, etc)
- Time to live became Hop limit

Redefined fields:

- Option mechanism

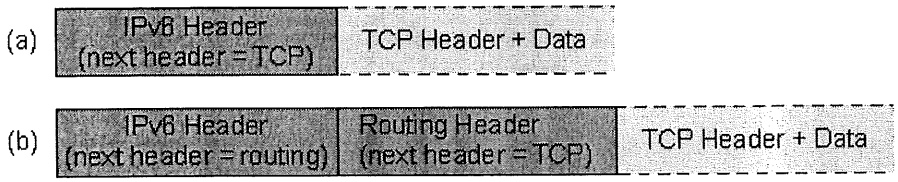


Figure 3.8 IPv6 next header

3.5.3 TCP:

TCP or the Transmission Control Protocol is found at the Transport layer of the OSI and the TCP/IP Models. The TCP is a connection-oriented protocol. It provides a reliable end-to-end byte stream over an unreliable network. The IP protocol gives no guarantee that datagrams will be delivered properly, so it is up to TCP to time out and retransmit them as need be. It also puts datagrams that arrive in the wrong order in their proper sequence and reassembles them into messages. TCP is not used in VoIP as it introduces big delays during retransmission of lost or timed-out datagrams.

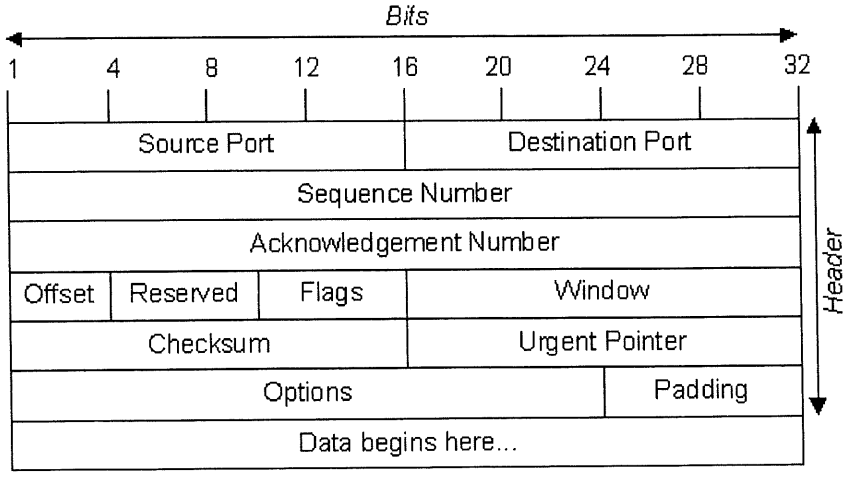


Figure 3.9 The TCP header

3.5.4 UDP:

UDP or the User Datagram Protocol also belongs to the transport layer. It has a smaller header and is less complex than TCP. It is a connectionless protocol, which means that datagrams are sent without having to establish a connection.

Unlike TCP, UDP does not ensure the retransmission of lost or delayed datagrams. It also offers no delivery guarantee nor does it handle the reassembling of datagrams that arrive out of order. This makes UDP an unreliable but a faster transmission protocol. VoIP uses UDP along with the IP protocol. Lost or delayed packets are mostly handled by the voice decoder [4].

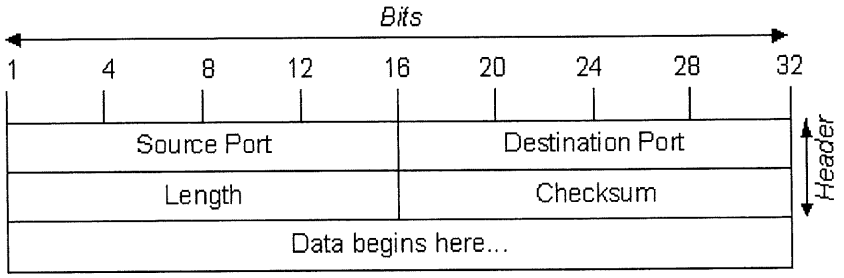


Figure 3.10 The UDP header

Source Port:

16 bits. This is an optional field. If it is not used, it is set to zero. Otherwise, it specifies the port of the sender.

Destination Port:

The port this packet is addressed to (16 bits)

Length:

16 bits. The length in bytes of the UDP header and the encapsulated data. The minimum value for this field is 8.

Checksum:

16 bits. This is computed as the 16-bit one's complement of the one's complement sum of a pseudo header of information from the IP header, the UDP header, and the data, padded as needed with zero bytes at the end to make a multiple of two bytes. If the checksum is set to zero, then checksumming is disabled. If the computed checksum is zero, then this field must be set to 0xFFFF.

3.6 Related Internet Telephony Standards: [1]

So far we have covered VoIP Networking Protocols. Before we continue with other VoIP related protocols, namely signaling, resource reservation and media transport protocols, we will talk about VoIP related standard organizations. Resolution of VoIP technical issues first started at the IETF, then spread to other internationally well-known standards institutions such as ITU-T and ETSI TIPHON.

3.6.1 IETF:

The IETF or Internet Engineering Task Force has four work groups in the VoIP field. Every work group deals with specific VoIP issue.

3.6.2 ETSI:

The TIPHON or Telecommunications and Internet Protocols Harmonization Over Networks is a project that identifies requirements and develops global standards for various aspects of communications between an IP network-based user and PSTN-based user. This includes a scenario where either a PSTN or an IP network is the origination, termination or transit network for the call.

3.6.3 ITU-T:

The ITU-T has launched an IP project to encompass all the ITU-T IP related work. The project will be regularly updated as work progresses and as the various ITU-T SGs (Study Group) expand their activities in support of the IP-related work.

3.7 Signaling protocols: [1]

3.7.1 SIP:

The Session Initiation Protocol (SIP) is an application-layer control protocol for creating, modifying and terminating sessions with one or more participants. It is a signaling protocol developed by the IETF MMUSIC (Multiparty Multimedia Session Control). SIP controls sessions such as Internet multimedia conferences, Internet telephone calls and multimedia distribution. Members in a session can communicate via multicast or via a mesh of unicast relations, or a combination of these.

SIP invitations used to create sessions carry session descriptions, which allow participants to agree on a set of compatible media types. Users can register their current location and SIP supports user mobility by proxying and redirecting requests to the users. SIP is not tied to any particular conference control protocol. It is designed to be independent of the lower-layer transport protocol and can be extended with additional capabilities. SIP is a text-based protocol

3.7.2 H.323:

The H.323 standard provides a foundation for audio, video, and data communications across IP-based networks, including the Internet. It is an end-to-end smart protocol derived from the ISDN standard. It was developed by ITU-T as a suite of multimedia communications. H.323 is an umbrella recommendation from the International Telecommunications Union (ITU) that sets standards for multimedia communications over Local Area Networks (LANs) which do not provide a guaranteed Quality of Service (QoS). H.323 standards are important building blocks for a broad new range of collaborative, LAN-based applications for multimedia communications.

It includes parts of H.245 RTP/RTCP for control, H.225.0/Q.931 for call setup, H.332 for large conferences, H.450.1, H.450.2 and H.450.3 for supplementary services, H.235 for security, H.246 for interoperability with the circuit switched service. It also includes audio codecs such as G.711, G.723.1, G.728, etc. and video codecs such as H.261 and H.263 that compress and decompress media streams.

Media streams are transported on RTP/RTCP, where RTP carries the actual media and RTCP carries status and control information.

3.7.3 MGCP:

MGCP or Media Gateway Control Protocol is a result of combining two previous protocols, Simple Gateway Control Protocol (SGCP) and Internet Protocol Device Control (IPDC). In MGCP, the call control intelligence is resident in the external call agents, having the MG to be responsible for connection control only. MGCP communicates between the MGC or call agents and the media gateway in a master/slave manner. It is meant to simplify standards for VoIP technology by eliminating the need for

complex, processor-intense IP telephony devices, thus simplifying and lowering the cost of these terminals.

MGCP does not compete with the other two well-known signaling protocols in VoIP, H.323 and SIP. It rather complements them by allowing either H.323 or SIP to set up a call between the call agents (MGC) and the VoIP client.

3.7.4 MEGACO:

MEGACO or Media Gateway Control describes a control model and protocol that operates between a media gateway (MG) and a media gateway controller (MGC). It allows the MGC to control the MG. Megaco was developed as part of the convergence movement, which brings voice and data together on the packet-switched Internet. It is the result of a joint cooperation between the IETF and the ITU-T, which identifies Megaco as H.GCP or H.248. The resulting standard Megaco/H.GCP comprises a group of interfaces and functions to be used to decompose gateways into MGs and MGCs.

The MG converts media provided in one type of network to the format required in another type of network. The MGC, on the other hand, controls parts of the call state that pertain to connection control for media channe ls in an MG. The MGC is sometimes referred to as Call Agent.

2.7.5 Difference between H.323, MGCP/MEGACO and SIP:

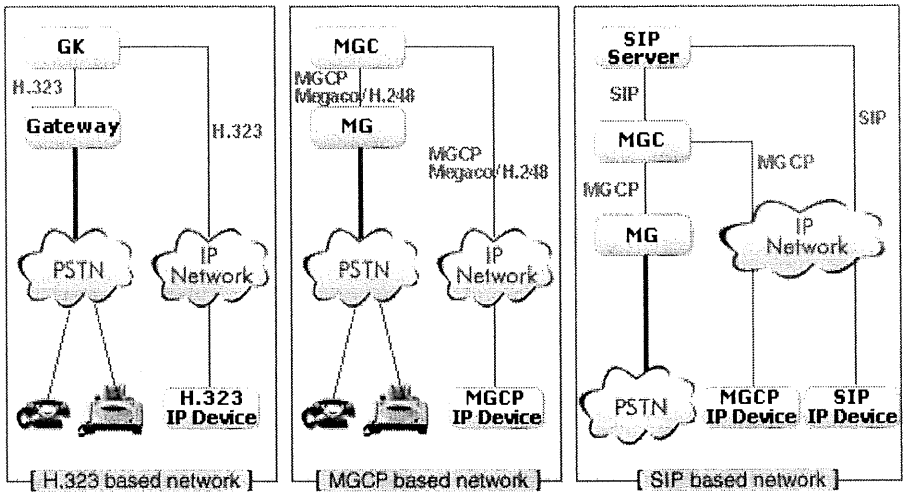


Figure 3.11 Difference between H.323, MGCP/MEGACO and SIP [18]

3.8 Media transport protocols:

2.8.1 RTP:

The RTP or Real-Time protocol is the Internet-standard protocol for the transport of real-time data, including audio and video [11]. It can be used for media-on-demand as well as interactive services such as Internet telephony. RTP consists of a data and a control part. The latter is called RTCP. RTP and RTCP are designed to be independent of the underlying transport and network layers. The protocol supports the use of RTP-level translators and mixers.

The data part of RTP is a thin protocol providing support for applications with real-time properties such as continuous media (e.g., audio and video), including timing reconstruction, loss detection, security and content identification.

RTP does not address the issue of resource reservation or quality of service control; instead, it relies on resource reservation protocols such as RSVP, YESSIR, etc.

3.8.2 RTCP:

RTCP or Real-Time Control Protocol provides support for real-time conferencing of groups of any size within an internet. This support includes source identification and support for gateways like audio and video bridges as well as multicast-to-unicast translators. It offers quality-of-service feedback from receivers to the multicast group as well as support for the synchronization of different media streams.

3.8.3 RTSP:

RTSP or Real Time Streaming Protocol is an application-level protocol for control over the delivery of data with real-time properties. RTSP provides an extensible framework to enable controlled, on-demand delivery of real-time data, such as audio and video. Sources of data can include both live data feeds and stored clips. This protocol is intended to control multiple data delivery sessions, provide a means for choosing delivery channels such as UDP, multicast UDP and TCP, and provide a means for choosing delivery mechanisms based upon RTP.

3.9 Resource reservation protocols:

3.9.1 RSVP:

RSVP or Resource ReSerVation Protocol is part of a larger effort to enhance the current Internet architecture with support for Quality of Service (QoS) [9]. The RSVP protocol is used by a host to request specific qualities of service from the network for a particular application data streams or flows. RSVP is also used by routers to deliver QoS requests to all nodes along the path(s) of the flows. It establishes and maintains state to provide the requested service. RSVP requests will generally result in resources being reserved in each node along the data path.

RSVP requests resources for simplex flows, i.e., it requests resources in only one direction. Therefore, RSVP treats a sender as logically distinct from a receiver, although the same application process may act as both a sender and a receiver at the same time. RSVP operates on top of IPv4 or IPv6, occupying the place of a transport protocol in the protocol stack. However, RSVP does not transport application data but is rather acts an Internet control protocol. RSVP is not itself a routing protocol; RSVP is designed to

operate with current and future unicast and multicast routing protocols. An RSVP process consults the local routing database(s) to obtain best routes. Like the implementations of routing and management protocols, an implementation of RSVP will typically execute in the background, not in the data forwarding path.

3.9.2 YESSIR:

YESSIR or Yet another Sender Session Internet Reservations is a new proposed mechanism that generates resource reservation requests by senders to reduce the processing overhead [16]. YESSIR was developed to address the two major problems of RSVP protocol, which are complexity and scalability. RSVP resulted in heavy message processing overhead at end systems and routers. This implies that in a backbone environment the amount of bandwidth consumed by refresh messages and the storage space needed to support a large number of flows at a router is too large.

YESSIR builds on top of RTCP and uses soft state to maintain reservation states. It supports shared reservation and associated flow merging and is backward compatible with the IETF Integrated Services models.

3.10 Header compression techniques:

Header compression is a technique used to compress and decompress the header information of a packet on a per-hop basis, utilizing redundancy within individual packets and between consecutive packets within a packet stream. There are two types of header compression, Transparent and Non-transparent. In the case of transparent compression, the header reconstructed by the decompresser matches the original header bit by bit. While in the case of non-transparent compression, reconstructed the header does not match the original header. Some fields, such as sequence counts, timestamps, UDP CRC, are altered.

3.10.1 CRTP:

CRTP or Compressed RTP is used to avoid the unnecessary consumption of available bandwidth. The RTP header compression feature is used on a link-by-link basis. CRTP

compresses the IP/UDP/RTP header in an RTP data packet from 40 bytes to approximately 2 to 5 bytes. CRTP accrues major gain in terms of packet compression because although several fields in the header change in every packet, the difference from packet to packet is often constant, and therefore the second-order difference is zero. The decompressor can reconstruct the original header without any loss of information.

The CRTP reduction in line overhead for multimedia RTP traffic results in a corresponding reduction in delay; CRTP is especially beneficial when the RTP payload size is small, for example, for compressed audio payloads of 20 to 50 bytes. CRTP can be used for media-on-demand and interactive services such as Internet telephony. As with RTP, CRTP provides support for real-time conferencing of groups of any size within the Internet. This support includes source identification and support for gateways such as audio and video bridges and for multicast-to-unicast translators. CRTP can benefit both telephony voice and multicast backbone (MBONE) applications running over slow links.

3.10.2 ROHC:

ROHC or RObust Header Compression was developed as a set of generic header compression schemes that perform well over links with high error rates and long roundtrip times [10]. Good performance of ROHC includes both minimal loss propagation and minimal added delay. Due to the limited packet loss robustness of CRTP, and the demands of the cellular industry for an efficient way of transporting voice over IP over wireless, the main focus of ROHC has so far been on compression of IP/UDP/RTP headers. ROHC RTP has become a very efficient, robust and capable compression scheme, able to compress the headers down to a total size of one octet only. Also, transparency is guaranteed to an extremely great extent even when residual bit errors are present in compressed headers delivered to the decomposer. ROHC has multiple compression schemes, where some are particularly suited to specific link layer technologies. In addition to generic TCP and UDP/RTP compression, applications of particular interest are voice and low-bandwidth video.

3.11 IP/UDP/RTP:

A typical IP telephony data packet starts with an IP, UDP, and RTP headers. If the IP protocol used is version 4 (IPv4), the headers' total will be 40 bytes. That is 20 bytes for IPv4, 8 bytes for UDP and 12 bytes for RTP headers plus payload. While if IP version 6 (IPv6) is used, the headers' total will be 60 bytes. That is 40 bytes for IPv6, 8 bytes for UDP and 12 bytes for RTP headers plus payload. These headers contain protocol information needed to properly transport the data. Included in this protocol information is data such as the source and destination IP addresses, the IP port number, packet sequence number, etc.

TABLE 3.1 OVERHEAD PARAMETERS

Overhead per packet	IPv4	IPv6	UDP	RTP
Bytes	20	40	8	12

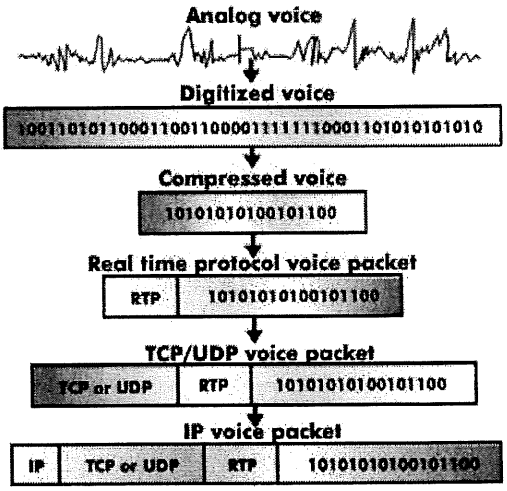


Figure 3.12 VoIP realization stages [12]

CHAPTER 4

G.729 SPEECH CODER

4.1 Introduction:

The G.729 coder, also known as CS-ACELP (Conjugate-Structure Algebraic Code-Excited Linear Prediction), is specified by the ITU (International Telecommunications Union). It compresses speech from 16 bit, 8 kHz samples (128 kbps) to 8 kbps, and was designed for cellular and networking applications. It provides a "toll quality" speech (i.e. as good as the telephone network), works well with background noise, and has been designed to perform well under error conditions. It can be used in a wide range of applications including wireless communications, digital satellite systems, packetized speech and digital leased lines.

G.729 fits into the general category of CELP (Code Excited Linear Prediction) speech coders. These coders are all based on a model of the human vocal system. In that model, the throat and mouth are modeled as a linear filter, and voice is generated by a periodic vibration of air exciting this filter. In the frequency domain, this implies that speech looks somewhat like a smooth response (called the envelope), modulated by a set of discrete frequency components. CELP coders all vary in the manner in which the *excitation* is specified, and the way in which the coefficients of the filter are represented. All of them generally break speech up into units called *frames*, which can be anywhere from 1ms to 100ms in duration.

TABLE 4.1 G.729 CODEC PARAMETERS

Codec	G.729	G.729A
Bit-rate	8 kb/s	8 kb/s
Frame size	10 ms	10 ms
Processing delay	10 ms	10 ms
Lookahead delay	5 ms	5 ms
Frame Length	10 bytes	10 bytes
DSP MIPS	20	10.5
RAM	3000	2000

4.2 G.729 overview:

G.729 is based on the CELP coding model. The speech quality of this coder is equivalent to G.726, also known as ADPCM (Adaptive Differential Pulse Code Modulation) at 32 kbps.

G.729 works with speech frames of 10 ms. This is equivalent to 80 samples at a sampling rate of 8000 samples per second. The speech signal is processed every 10 ms frame in order to extract the CELP model parameters (linear-prediction filter coefficients, adaptive and fixed-codebook indices and gains). These parameters are encoded and transmitted. At the decoder side, transmitted parameters are decoded to get the excitation and synthesis filter parameters. The original speech is reconstructed by passing the excitation through a short term synthesis filter, as is shown in Figure 1. The short-term synthesis filter is a Linear Prediction (LP) filter of a 10^{th} order. The long-term synthesis filter, or pitch synthesis filter is realized using an adaptive-codebook approach. The output speech signal is enhanced by a postfilter.

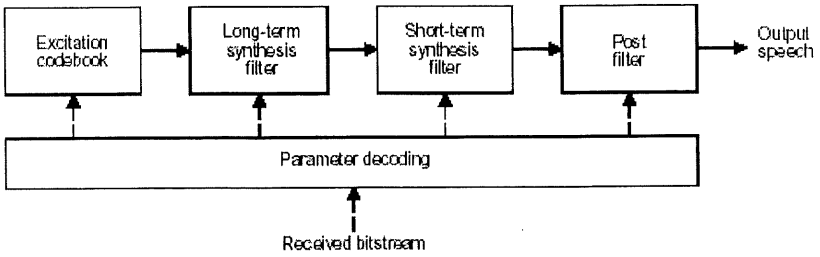


Figure 4.1 Block diagram of conceptual CELP synthesis model[13]

4.2.1 The encoder:

The input signal is filtered and scaled through a high-pass filter in the pre-processing block. Following the pre-processing, the signal is passed through an LP analysis block. This is done once per 10 ms frame in order to compute the LP filter coefficients. The computed coefficients are then converted into Line Spectrum Pairs (LSP). The new LSPs are quantized using a predictive two-stage Vector Quantization (VQ) with 18 bits. The encoding principle is shown in Figure 2.

The excitation signal is determined using an analysis-by-synthesis search procedure. This helps minimizing the error between the original and reconstructed speech signal according to a perceptually-weighted distortion measure. This is done by filtering the error signal through a perceptual weighting filter, whose coefficients are derived from the unquantized LP filter. The amount of perceptual weighting adaptively changes to improve the performance of input signals with a flat frequency-response.

The excitation parameters (fixed and adaptive-codebook parameters) are determined once per 5 ms (40 samples) subframe. Quantized and unquantized LP filter coefficients are used for the second subframe, whereas in the first subframe only interpolated LP (quantized and unquantized) filter coefficients are used. An open-loop pitch delay is estimated once per 10 ms frame and is based on the perceptually-weighted speech signal.

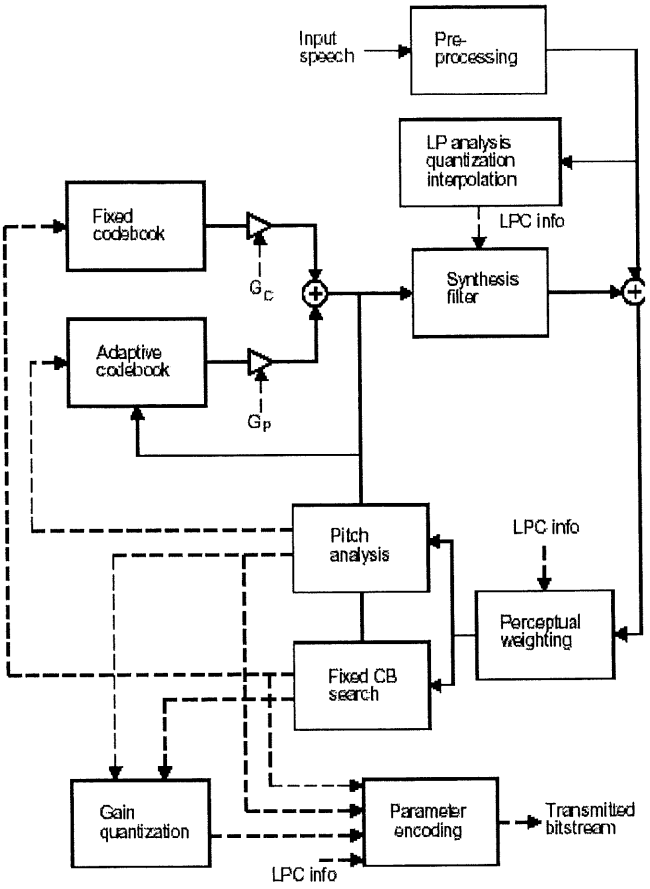


Figure 4.2 Encoding principle of the CS-ACELP encoder [13]

The following operations are repeated for each subframe:

- The target signal $x(n)$ is computed by filtering the LP residual through the weighted synthesis filter $W(z)/\hat{A}(z)$. The initial states of these filters are updated by filtering the error between LP residual and excitation.
- The impulse response $h(n)$ of the weighted synthesis filter is computed.
- The Closed-loop pitch analysis (adaptive-codebook delay and gain search) is done using the target $x(n)$ and impulse response $h(n)$, by searching around the value of the open-loop pitch delay. A fractional pitch delay with 1/3 resolution is

used. The pitch delay is encoded with 8 bits in the first subframe and differentially encoded with 5 bits in the second subframe.

- The target signal $x(n)$ is updated by subtracting the (filtered) adaptive-codebook contribution. This new target, $x'(n)$, is used in the fixed codebook search to find the optimum excitation. An algebraic codebook with 17 bits is used for the fixed-codebook excitation. The gains of the adaptive and fixed-codebook contributions are vector quantized with 7 bits, (with MA prediction applied to the fixed-codebook gain).
- Finally, the filter memories are updated using the determined excitation signal.

4.2.2 The decoder:

The main function of the decoder is to extract the parameter's indices from the received bitstream and reconstruct the output speech signal. The decoder principle is shown in Figure 3. The parameters' indices are decoded to obtain the coder parameters corresponding to a 10 ms speech frame. These parameters includes the LSP coefficients, the two fractional pitch delays, the two fixed-codebook vectors, and the two sets of adaptive and fixed-codebook gains. The LSP coefficients are interpolated and converted to LP filter coefficients for each subframe. The following steps are done for every 5 ms subframe:

- The excitation signal is constructed by adding the adaptive and fixed-codebook vectors scaled by their respective gains;
- The speech is reconstructed by filtering the excitation signal through the LP synthesis filter;
- The reconstructed speech signal is passed through a post-processing stage, which includes an adaptive postfilter based on the long-term and short-term synthesis filters, followed by a high-pass filter and scaling operation.

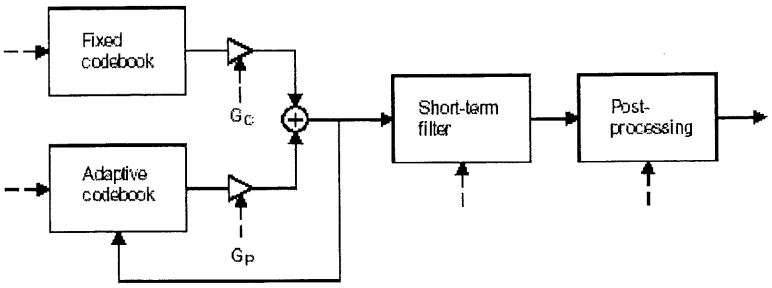


Figure 4.3 Principle of the CS-ACELP decoder [13]

4.3 Short-term prediction analysis:

The short-term analysis and synthesis filter is based on 10^{th} order Linear Prediction (LP) filters. The LP synthesis filter is defined as:

$$\frac{1}{\hat{A}(z)} = \frac{1}{1 + \sum_{i=1}^{10} \hat{a}_i z^{-i}} \tag{4-1}$$

where $\hat{a}_i$, $i = 1,...,10$, are the quantized LP coefficients.

The short-term prediction or linear prediction analysis is performed once per speech frame. This is done using the autocorrelation method with a 30 ms asymmetric window. The autocorrelation coefficients of windowed speech are computed and converted to LP coefficients every 80 samples (10 ms) using the Levinson algorithm. The LP coefficients are transformed to the LSP domain for quantization and interpolation purposes. The interpolated (quantized and unquantized) filters are converted back to the LP filter coefficients in order to construct the synthesis and weighting filters for each subframe.

4.4 Long-term prediction analysis:

The long-term prediction is done once per frame (10 ms). The search for the best adaptive-codebook delay is limited to a range around a candidate delay *Top*, which is obtained from an open-loop pitch analysis. The open-loop pitch estimation uses the weighted speech signal $sw(n)$ of Equation (2), and is determined as follows:

In the first step, three maxima of the correlation,

$$sw(n) = s(n) + \sum_{i=1}^{10} a_i \gamma_1^i s(n-i) - \sum_{i=1}^{10} a_i \gamma_2^i sw(n-i) \qquad n = 0,\ldots,39 \qquad (4\text{-}2)$$

$$R(k) = \sum_{n=0}^{79} sw(n).sw(n-k) \qquad (4\text{-}3)$$

are found in the following three ranges:

$i = 1$: 80,...,143
$i = 2$: 40,...,79
$i = 3$: 20,...,39

The retained maxima $R(t_i)$, $i = 1,...,3$, are normalized through:

$$R'(t_i) = \frac{R(t_i)}{\sqrt{\sum_n sw^2(n-t_i)}} \qquad i = 1,..,3 \qquad -4)$$

The best delay among the three normalized correlations is selected by favouring the delays with the values in the lower range. This is done by weighting the normalized correlations corresponding to the longer delays. The best open-loop delay *Top* is determined as follows:

$T_{op} = t1$
$R'(T_{op}) = R'(t_1)$
if $R'(t_2) = 0.85R'(T_{op})$
$R'(T_{op}) = R'(t_2)$
$T_{op} = t_2$
end
if $R'(t_3) = 0.85R'(T_{op})$
$R'(T_{op}) = R'(t_3)$
$T_{op} = t_3$
end

This procedure of dividing the delay range into three sections and favouring the smaller values is used to avoid choosing pitch multiples.

4.5 Innovation codebook structure:

The fixed codebook is based on an algebraic codebook structure using an interleaved single-pulse permutation design. Each codebook vector in the fixed codebook contains four non-zero pulses. Each pulse has an amplitude value of 1 or -1, and its position is given in Table 4.2.

TABLE 4.2 STRUCTURE OF FIXED CODEBOOK

Pulse	Sign	Positions
i_0	s_0: ±1	m_0: 0, 5, 10, 15, 20, 25, 30, 35
i_1	s_1: ±1	m_1: 1, 6, 11, 16, 21, 26, 31, 36
i_2	s_2: ±1	m_2: 2, 7, 12, 17, 22, 27, 32, 37
i_3	s_3: ±1	m_3: 3, 8, 13, 18, 23, 28, 33, 38 4, 9, 14, 19, 24, 29, 34, 39

The codebook vector $c(n)$ is constructed by taking a zero vector of dimension 40, and putting the four unit pulses at the found locations, multiplied with their corresponding sign:

$$c(n) = s_0 d(n - m_0) + s_1 d(n - m_1) + s_2 d(n - m_2) + s_3 d(n - m_3) \qquad n = 0,...,39 \qquad (4\text{-}5)$$

where d(0) is a unit pulse. An adaptive pre-filter $P(z)$ is incorporated in the codebook and is used to filter the selected codebook vector in order to enhance harmonic components, which improve the quality of the reconstructed speech. The filter is defined as follows:

$$P(z) = 1 / (1 - \beta z^{-T}) \qquad (4\text{-}6)$$

where T is the integer component of the pitch delay of the current subframe, and ß is a pitch gain. The value of ß is made adaptive by using the quantized adaptive-codebook gain from the previous subframe, that is:

$$\beta = g_p^{(m-1)} \qquad \text{bounded by} \quad 0.2 = \beta = 0.8 \tag{4-7}$$

For delays less than 40, the codebook $c(n)$ of Equation (4-5) is modified according to:

$$c(n) = \begin{cases} c(n) \ldots\ldots\ldots\ldots\ldots\ldots\ldots & n = 0,\ldots, T-1 \\ c(n) + \beta c(n-T) \ldots\ldots & n = T,\ldots, 39 \end{cases} \tag{4-8}$$

This modification is incorporated in the fixed-codebook search by modifying the impulse response $h(n)$ according to:

$$h(n) = \begin{cases} h(n) \ldots\ldots\ldots\ldots\ldots\ldots\ldots & n = 0,\ldots, T-1 \\ h(n) + \beta h(n-T) \ldots\ldots & n = T,\ldots, 39 \end{cases} \tag{4-9}$$

4.6 Gains quantization:

Both the adaptive-codebook gain or pitch gain and the fixed-codebook gain are vector quantized using 7 bits. The gain codebook search is realized by minimizing the mean-squared weighted error between original and reconstructed speech which is given by:

$$E = \boldsymbol{x}^t\boldsymbol{x} + g_p^2\, \boldsymbol{y}^t\boldsymbol{y} + g_c^2\, \boldsymbol{z}^t\boldsymbol{z} - 2g_p\boldsymbol{x}^t\boldsymbol{y} - 2g_c\boldsymbol{x}^t\boldsymbol{z} + 2g_pg_c\boldsymbol{y}^t\boldsymbol{z} \tag{4-10}$$

where $\boldsymbol{x}$ is the target vector, $\boldsymbol{y}$ is the filtered adaptive-codebook vector, and $\boldsymbol{z}$ is the fixed codebook vector convolved with $h(n)$,

$$z(n) = \sum_{i=0}^{n} c(i)h(n-i) \qquad n = 0,\ldots,39 \tag{4-11}$$

4.6.1 Memory update:

An update of the states of the synthesis and weighting filters is needed to compute the target signal in the next subframe. The excitation signal $u(n)$ is calculated after the quantization of the two gains in the current subframe and is obtained by Equation (4-12):

$$u(n) = g_p v(n) + g_c\, c(n) \qquad n = 0,...,39 \qquad (4\text{-}12)$$

where g_p and g_c are the quantized adaptive and fixed-codebook gains, respectively, $v(n)$ is the adaptive-codebook vector (interpolated past excitation), and $c(n)$ is the fixed-codebook vector including harmonic enhancement.

4.7 Decoding and post-processing:

The excitation $u(n)$ is passed through an LP synthesis filter. The reconstructed speech for the current subframe is given by:

$$s(n) = u(n) - \sum_{i=1}^{10} \hat{a}_i s(n-i) \qquad n=0,...,39 \qquad (4\text{-}13)$$

where $\hat{a}_i$ are the interpolated LP filter coefficients for the current sub-frame. The reconstructed speech (n) is then processed by the post processor described in the next sub-clause.

4.7.1 Post-processing:

The post-processing stage consists of three functions: adaptive post-filtering, high-pass filtering and signal upscaling. The adaptive post-filter is a cascade of three filters: a long-term post-filter $H_p(z)$, a short-term post-filter $H_f(z)$ and a tilt compensation filter $H_t(z)$, followed by an adaptive gain control procedure. The post-filter coefficients are updated on a 5 ms sub-frame basis. The post-filtering process is organized as follows:

First, the reconstructed speech (n) is inverse filtered through $\hat{A}(z\ /?_n)$ to produce the residual signal (n). This signal is used to compute the delay T and gain g_t of the long-term post-filter $H_p(z)$. The signal (n) is then filtered through the long-term post-filter $H_p(z)$ and the synthesis filter $1/[g_f \hat{A}(z\ /?_d)]$.

Finally, the output signal of the synthesis filter $1/[g_f \hat{A}(z/\gamma_d)]$ is passed through the tilt compensation filter $H_t(z)$ to generate the post-filtered reconstructed speech signal $s_f(n)$. An adaptive gain control is then applied to $s_f(n)$ to match the energy of $s(n)$. The resulting signal $s_f'(n)$ is high-pass filtered and scaled to produce the output signal of the decoder.

TABLE 4.2 DESCRIPTION AND BIT ALLOCATION OF G.729 PARAMETERS

Symbol	Descriptions	Bits
L0	Switched MA predictor of LSP quantizer	1
L1	First stage vector of quantizer	7
L2	Second stage lower vector of LSP quantizer	5
L3	Second stage higher vector of LSP quantizer	5
P1	Pitch delay first subframe	8
P0	Parity bit for pitch delay	1
C1	Fixed codebook first subframe	13
S1	Signs of fixed-codebook pulses 1st subframe	4
GA1	Gain codebook (stage 1) 1st subframe	3
GB1	Gain codebook (stage 2) 1st subframe	4
P2	Pitch delay second subframe	5
C2	Fixed codebook second subframe	13
S2	Signs of fixed-codebook pulses 2nd subframe	4
GA2	Gain codebook (stage 1) 2nd subframe	3
GB2	Gain codebook (stage 2) 2nd subframe	4

4.8 Concealment procedure:

An error concealment procedure has been incorporated in the G.729 decoder to reduce the quality degradation in the reconstructed speech because of frame erasures in the bit stream. The error concealment process is activated when a frame of coder parameters has been identified as lost or corrupted. In this case, the decoder's BFI (Bad Frame Indication) parameter is set to 1; under normal circumstances BFI is always set to 0. The mechanism for detecting frame erasures is not defined in the G.729 Recommendation, and will depend on the application.

The concealment strategy has to reconstruct the current frame, based on previously received information. The method replaces the missing excitation signal with one of similar characteristics, while gradually decaying its energy. This is done by using a voicing classifier based on the long-term prediction gain, which is computed as part of the long-term post-filter analysis. The long-term post-filter finds the long-term predictor for which the prediction gain is more than 3 dB. For the error concealment process, a 10 ms frame is declared periodic if at least one 5 ms sub-frame has a long-term prediction gain of more than 3 dB. Otherwise the frame is declared non-periodic. An erased frame inherits its class from the preceding (reconstructed) speech frame. Note that the voicing classification is continuously updated based on this reconstructed speech signal.

The specific steps taken for an erased frame are:

1) Repetition of the synthesis filter parameters
2) Attenuation of adaptive and fixed-codebook gains
3) Attenuation of the memory of the gain predictor
4) Generation of the replacement excitation(Explained in the following paragraph)

4.8.1 Generation of the replacement excitation:

The excitation used depends on the periodicity classification. If the last reconstructed frame was classified as periodic, the current frame is considered to be periodic as well. In that case only the adaptive codebook is used, and the fixed codebook contribution is set to zero. The pitch delay is based on the integer part of the pitch delay in the previous frame, and is repeated for each successive frame.

If the last reconstructed frame was classified as non-periodic, the current frame is considered to be non-periodic as well, and the adaptive-codebook contribution is set to zero. The fixed-codebook contribution is generated by randomly selecting a codebook index and sign index.

4.8.2 G.729 concealment limitations:

G.729 error concealment procedure has many limitations. As mentioned earlier, concealment strategy has to reconstruct the current lost frame, based on previously received good frames. Attenuations, repetitions and zero settings of previously received parameters introduce rapid divergence. This strategy leads to error propagation due to encoder/decoder de-synchronization. In the next chapter we will implement a new technique based on the late-arrival frames. Contradictory to G.729 concealment procedure, the new technique will insure a rapid convergence and minimize error propagation.

4.9 Impact of FER on concealment quality:

Errors in the bit stream can make the whole speech frame unusable leading to an FER or a Frame-Error-Rate. FER is more closely related to audio quality. An FER of 2% will be audible. If the FER is too high, the audio codec will mute until the FER decreases. FER is an objective measurement but measuring audio distortion or Mean Opinion Scores (MOS) is more preferable.

CHAPTER 5

IMPROVING RECOVERY USING LATE-ARRIVAL FRAMES

5.1 Past work on error concealment:

The current state-of-the-art frame reconstruction for modern speech coders, whether vocoders or CELP-based systems, essentially consist in repeating the speech parameters contained in the last correctly received frame. If two or more consecutive frames are lost, increasingly strong muting is applied. This approach, which has the advantage of not introducing any extra delay, has been followed by most recent speech coding standards.

Among the past works done on error concealment is the work made by Texas Instruments (TI). The following is a description of TI work:

5.1.1 Improved frame erasure concealment for CELP-based coders:

The work presents two new techniques for concealing frame erasures for CELP-based speech coders. Two main approaches were followed: *interpolative*, where both past and future information are used to reconstruct the missing data, and repetition-based, where no future information is required. Key features of the *repetition-based* approach include improved muting, pitch delay jittering, and LPC bandwidth expansion. The interpolative approach can be employed in Voice over IP scenarios at no extra cost in terms of delay. Applied to the ITU-T G.729 ACELP 8 kb/s speech coding standard, both interpolative and repetition-based techniques outperform standard concealment in informal listening tests.

5.1.2 Repetition-based concealment:

There are three key features in this method. The first feature is a new muting algorithm which mutes the excitation signal directly with a muting factor to decay the signal gradually, instead of attenuating the codebook gains in the previous frame as is done in the G.729 standard frame erasure concealment. The second feature is a pitch delay jittering for a bursty frame erasure. The random jitter is added to the repeated pitch delay only when a consecutive frame erasure occurs. The third feature is LPC bandwidth expansion for bursty frame erasures. As is the case in the pitch delay jittering, the LPC bandwidth in the previous frame is expanded only when a consecutive frame erasure occurs. The proposed method is designed not only to reconstruct speech in bad frames but also to recover speech smoothly after the frame erasure.

5.1.3 Interpolative concealment:

If future speech data is, or can be made, available, then an interpolative approach to frame erasure concealment becomes possible. This should intuitively produce better concealment than the simpler repetition-based approach, at the expense of extra delay.

Interpolation-based concealment for CELP coders has hardly been investigated. The reason for such relative neglect is probably the extra delay entailed by the approach, which is not acceptable in applications like wireless where delay is tightly controlled.
The emergence of a new, important application, however, Voice over IP networks, makes interpolative concealment attractive. In VoIP systems, in fact, one or more future frames are, at least most of the time, available at the decoder, stored in the so-called playout buffer. Such buffer, introduced to smooth out the effects of delay jitter, is an essential component of all VoIP receivers. Interpolative concealment can exploit the delay introduced by the playout buffer to improve performance under frame erasures at no extra cost in terms of delay.

Packets arriving from the network are first processed by the network module. Statistics are collected, packets ordered and transferred to the playout buffer. If near the time of

playback the packet has not yet arrived, it is declared lost and the frame erasure concealment module reconstruct it using both past and future frames.

Another past work made by TI. Here's a description of what was done.

5.2 Efficient CELP-based diversity schemes for VoIP:

Diversity schemes include information about packet *n* in future packets or send information about packet n via separate paths. If packet n is lost, it is reconstructed from information included in future packets or information received via separate paths.

The work presents CELP-based diversity schemes for voice over packet applications. The diversity schemes reduce the impact of packet losses while being efficient in terms of both bandwidth requirement and computational complexity. With our diversity schemes, transmission schemes that allocate bandwidth resources among diversity stages during congestion give significantly better performance than schemes that use no diversity during congestion, for the same bandwidth usage.

Time diversity schemes include information about packet n in future packets, and if packet n is lost it is reconstructed from information included in future packets. Path diversity schemes send information about packet *n* via different paths and if packet *n* is lost, it is reconstructed from information received via different paths. Diversity schemes can be realized utilizing redundancy schemes or multiple description schemes.

5.2.1 Redundancy schemes:

Redundancy schemes piggyback a version or function of *n-th* packet on future packets. They can be categorized into media independent forward-error-correction (FEC) schemes, and media specific redundancy schemes. The existing media-specific redundancy schemes employ a separate encoding scheme for generating the redundancy version of *n-th* packet to future packets. These existing approaches either increase the computational complexity or substantially increase the bandwidth requirement.

5.2.2 Multiple description schemes:

These schemes break the input signal into multiple descriptions such that each of the descriptions has less than the full information intended for transmission, and the combination of all the descriptions give the full information. In this paper we introduce CELP-based, multiple description data partitioning schemes. The schemes are efficient in terms of both computational complexity and bandwidth requirement.

CELP is an analysis-by-synthesis speech coding method, and operates on frames of speech samples. The output speech is synthesized by applying an excitation signal is the sum of adaptive codebook contribution and fixed codebook contribution. Each of the contributions has its own gain value.

5.3 Late-arrival versus missing frames:

Prediction-based speech coders, such as G.729, are very sensitive to frame losses and the error propagation caused by these losses. This sensitivity is due to inter frame dependencies in the predictor internal states. The internal states of an encoder and its corresponding decoder include information about past samples required for long-term and short-term predication, as well as memory information for predictive quantizers.

In voice over IP networks, one or several speech frames are encoded and grouped into a single packet. Packets travelling over an IP network, such as the Internet, are subject to variable delays, i.e. jitter. This jitter is due to queuing at routers along the transmission path. At the receiver side, a jitter buffer or a playout buffer is used to wait for all packets arriving within an acceptable time limit. Jitter buffer helps controlling the effect of variable delays. However, some packets may still arrive too late to be decoded. Missing or late packets are usually considered as "lost", and a concealment procedure has to be applied to replace the missing audio samples.

When all frames are received correctly, i.e. no bit errors or lost packets, the encoder and decoder predictor states are identical. The speech signal generated by the decoder is said to be "correct", i.e. identical to the local synthesis at the encoder side. When one or more

frames are lost, the decoder has to apply a concealment procedure in order to generate the missing audio samples. This procedure produces some distortion, even if in many instances the concealed speech retains much of the missing speech structure. Moreover, it does not update correctly the internal state of the decoder. Therefore, due to the highly predictive nature of modern coders, errors introduced in the concealed frame also propagate in the following ones even if the decoder receives the corresponding packets correctly.

5.4 G.729 embedded error concealment:

An error concealment procedure has been incorporated in the decoder to reduce the degradation in the reconstructed speech because of frame losses in the bitstream. This error concealment process is functional when the frame of coder parameters (corresponding to a 10 ms frame) has been identified as being lost. The mechanism for detecting frame loss is not defined in the Recommendation of G.729, and will depend on the application.

Under normal conditions, i.e. when there are neither transmission errors nor lost frames, the internal states of the coder and that of the decoder changes simultaneously. In other words, they are identical for every frame. The coder and the decoder share the same past, thus the optimal speech parameters chosen by the coder are also optimal at the decoder level.

The concealment strategy has to reconstruct the current frame, based on previously received information. The method replaces the missing excitation signal with one of similar characteristics, while gradually decaying its energy. This is done by using a voicing classifier based on the long-term prediction gain, which is computed as part of the long-term post-filter analysis. The long-term post-filter finds the long-term predictor for which the prediction gain is more than 3 dB. For the error concealment process, a 10 ms frame is declared periodic if at least one 5 ms subframe has a long-term prediction gain of more than 3 dB. Otherwise the frame is declared non-periodic. An erased frame

inherits its class from the preceding (reconstructed) speech frame. Note that the voicing classification is continuously updated based on this reconstructed speech signal.

The specific steps taken for a lost frame are:

1) Repetition of the synthesis filter parameters;

2) Attenuation of adaptive and fixed-codebook gains;

3) Attenuation of the memory of the gain predictor;

4) Generation of the replacement excitation.

5.5 Principles of the proposed algorithm:

The usual procedure followed in case of a late-arrival frame is simply to reject it. This loss of speech information leads to a degradation in the speech signal and causes a very slow convergence towards the original signal.

The proposed algorithm takes advantage of the late-arrival frames and uses their information content to accelerate the convergence of the decoder during the recovery phase.

This algorithm is based on [3][2], which was implemented on AMR-WB (Adaptive Multi-Rate - Wideband) speech coder. In our case, the same algorithm will be implemented on G.729 speech coder, which is a narrowband coder. G.729 coder is widely used in PSTN and VoIP applications, while as AMR-WB is new coder that is still not deployed.

5.5.1 Algorithm description:

Figure 5.1 simplifies the recovery algorithm using the late-arrival frame. Three cases are considered.

1- No frame loss:
Binary frames are received and decoded normally. The speech signal generated by the decoder is "correct".

2- Frame *n* is lost:
Binary frames are received and decoded normally up to frame *n-1*. Binary frame *n* is lost. It is replaced by the internal G.729 concealment procedure. This means

certain divergence was introduced between the "correct" speech signal and the decoded speech signal. This divergence propagates from the recovery phase (frame $n+1$ and up)

3- Frame n arrives late:

At the moment of decoding binary frame $n+1$, frame n arrives late. Two cases arise:

Case A:

Rejecting binary frame n and ignoring its speech information content. Thus the internal decoder state produced by the G.729 concealment procedure in kept unchanged (i).

Case B:

Restoring the internal decoder states as they were before the concealment procedure of frame n. Decoding frame n as normal (ii) without keeping the output speech signal. Extracting speech parameters that are needed to correct the internal states of the decoder. Then, frame $n+1$ *is* normally decoded, this time using the good internal states (iii). In reality, it is important to make some smoothing between frame n and frame $n+1$ to eliminate any discontinuity.

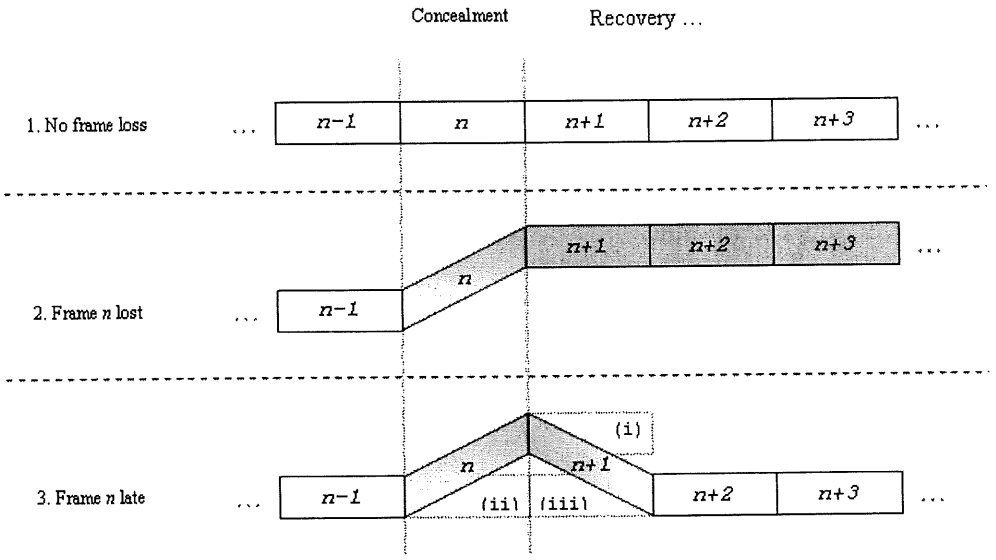


Figure 5.1 Sequence of frame decoding [3]

5.6 Application to G.729:

5.6.1 Internal state structure:

All variables defining the internal states of the coder and decoder (static variable, filter memory, counters, etc.) are regrouped in one or many structures. In our case, the coder state structure is Coder_State and that of the decoder is called Decoder_State. This makes the use of multi-instances of the codec more feasible.

5.6.2 Codec interface:

The interface with the coder/decoder us realized via a couple of functions. Usually, it is done by the following functions:

- Initialization:
  Memory allocation and initialization of internal state structures
  (Coder_State and Decoder_State)

- Frame coding:
  This function converts the speech frame (input file) into a binary frame (output file). It uses and constantly updates the internal states of the coder.

  Also sometime included in this function is the bit-rate and the codec mode.

- Frame decoding:
  This function converts a binary frame (input frame) into a decoder speech frame (output frame). It uses and constantly updates the internal states of the decoder.

  In this function we also find the BFI (Bad Frame Indicator) parameter. When BFI is equal to 0, frame decoding is normally executed. If BFI is equal to 1, then the the decoder concealment procedure is executed.

  To be able to take into consideration the late-arrival frames, a new parameter called UPD (UPDATE) is added to the decoding function input arguments.

- Codec exit

  At this final stage, the codec frees allocated memories by the internal state structures Coder_State and Decoder_State.

5.6.3 Call sequence of decoder function:

When frame n is lost, the call sequence of the decoder function (without taking into consideration any late-arrival) is as follows:

Decoder	($Frame_{n-1}$,	$Speech_{n-1}$,	BFI=0)	
Decoder	(- ,	$Speech_n$,	BFI=1)	### Concealment
Decoder	($Frame_{n+1}$,	$Speech_{n+1}$,	BFI=0)	### Début recovery
Decoder	($Frame_{n+2}$,	$Speech_{n+2}$,	BFI=0)	

When frame *n* arrives late but can still play a role in the decoding of frame *n+1*, the call sequence of the decoder function (modified by the introduction of the new argument UPD) is as follows:

Decoder	($Frame_{n-1}$,	$Speech_{n-1}$,	BFI=0, UPD = 0)	
Decoder	(- ,	$Speech_n$,	BFI=1, UPD = 0)	### Concealment
Decoder	($Frame_n$,	- ,	BFI=0, UPD = 1)	### Update
Decoder	($Frame_{n+1}$,	$Speech_{n+1}$,	BFI=0, UPD = 0)	### Début recovery
Decoder	($Frame_{n+2}$,	$Speech_{n+2}$,	BFI=0, UPD = 0)	

This call sequence is simple and transparent to the user. When the frame arrives late, the user makes a call to the decoder function specifying that the binary frame is only to be used for updating the internal states (no speech signal generated!).

5.6.4 Modifications implemented in G.729:

- New field is added to the internal state structure of the decoder (in file decld8cp.h):
  - Field for storing the excitation signal

- Two structures created for storing the internal states Good (internal states before any lost frame) and Conc (internal states during lost frame concealment).

Functions were written for manipulating these structures (Initialize, Copy, Mix, and others found in file Update.c and Update.h).

- In the decoder calling function (in file g729.c):
  - Add flag UPD to the parameters of the decoder calling function. The value of UPD depends on whether the update scenario is set for use after a bad frame (BFI=1).

- In file codeccp.c, which is the interface between file g729.c and decld8cp.c, do the following:
  - Initialize the Good and Conc internal states
  - Add flag UPD to the decoder function
  - If a frame is intended for Update only then:
    - Decode the frame without producing an audio output. Just save the internal states.
  - If the frame comes after an update:
    - Add flag SaveMix to the decoder calling function. If SaveMix=1, then save the excitation signal. If SaveMix=2, then Mix the Good and Conc Excitation signals.
    - Calculate the excitation signal following the concealment. Set flag SaveMix=1 to save the excitation. No audio output produced.
    - Mix the Good and Conc internal states (keep the "Good" synthesis filter memory to avoid any discontinuity).
    - Calculate the excitation with the new mixed internal states. Call the decoder function with flag SaveMix=2 to mix the excitation signals and to perform a synthesis of it. This time an audio signal is outputted.

- In the decoder calling function (in file decld8cp.c):
  - Add flag SaveMix to the parameters of the decoder function
  - Implement the different actions with respect to the value of flag SaveMix.

5.7 Signal examples:

In the following we show some signal examples in Figure 5.2, 5.3 and 5.4. Line 1 of each Figure shows the output signal of the original decoder without any frame loss. Line 2 shows the output signal of the original decoder when the 10^{th} frame is lost. Line 3 shows the output signal of the modified decoder when an update is applied following the concealment. Line 4 shows the difference between the output signal of the original decoder and that with the lost 10^{th} frame.

We can notice the error propagation spanning across several frames and causing a high distortion level. Finally, line 5 shows the difference between the output signal of the original decoder and that of the modified decoder. It is clear that the update following the concealment is very efficient and that the error does not propagate further than the recovery frame.

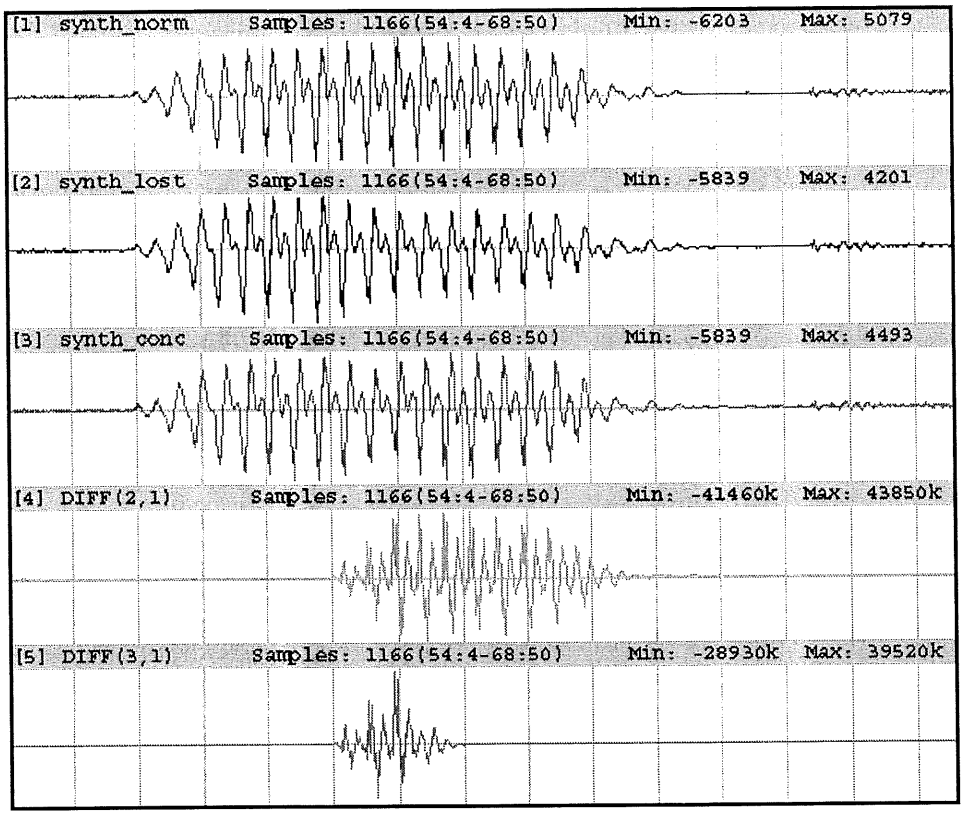


Figure 5.2 Signal examples

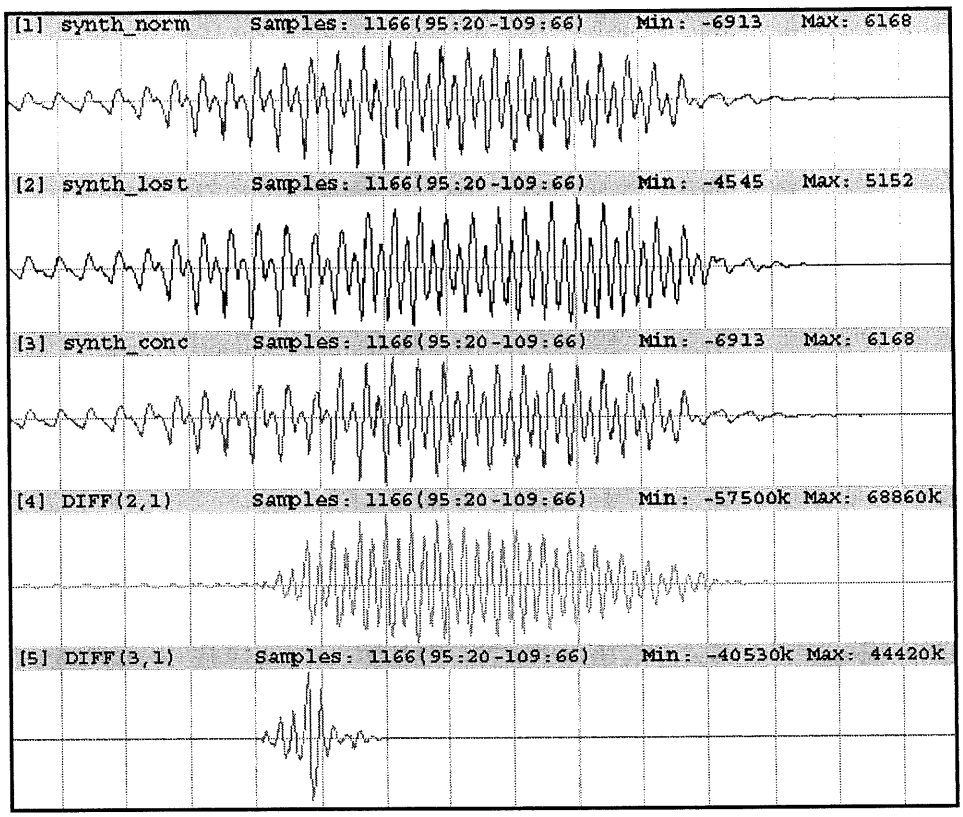


Figure 5.3 Signal examples

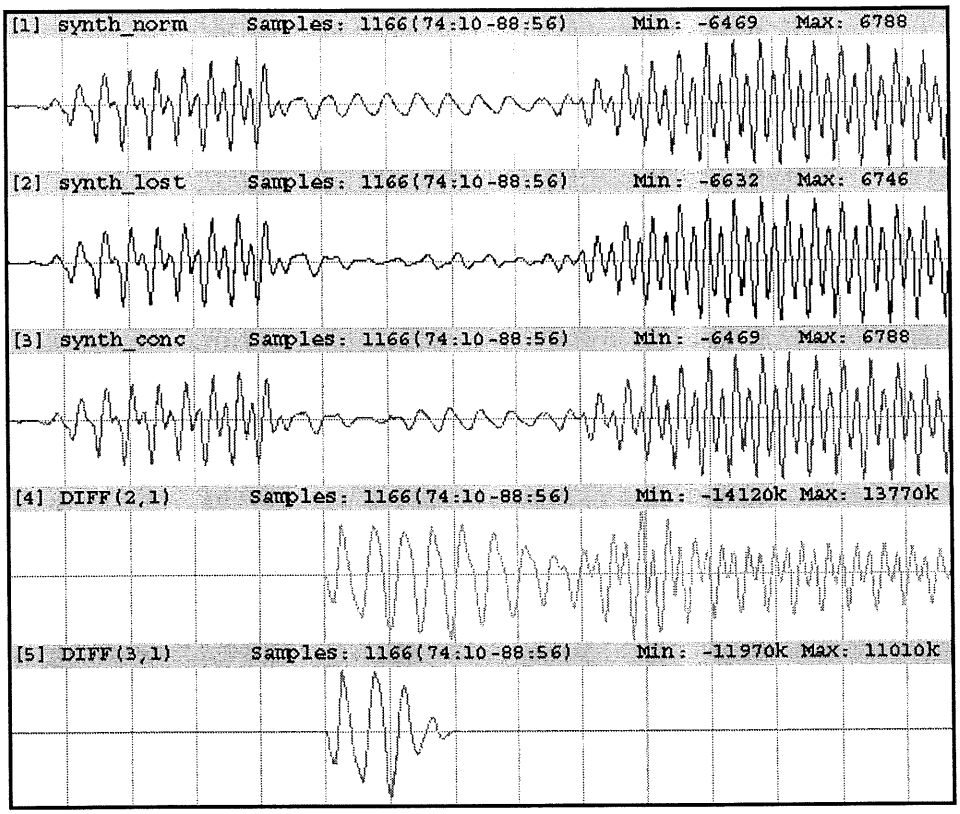


Figure 5.4 Signal examples

CHAPTER 6

OBJECTIVE AND SUBJECTIVE RESULTS

6.1 Introduction:

The final part of our work consists of presenting the results of the listening tests. These tests were conducted at the University of Sherbrooke Audio Test Lab. The tests were based on six speech files, three of which are for a male speaker and three for a female. We have introduced four test conditions for every speech file. In the following, we will discuss the objective and subjects results and their implications.

6.2 Objective results:

6.2.1 Effect of lost and late frames:

Traditionally, late frames were considered as lost. These frames carried audio in an encoded binary format. The loss of a binary audio frame resulted in a loss of important information that was essential for decoding current and subsequent frames. Consequently, this issue lead to audio artifacts that can be long lasting and annoying.

When one or more frames are lost, the decoder loses information necessary for decoding the audio signal. The lost frame is usually replaced by another generated frame. This new frame is the result of an extrapolation of the correctly received information in the last good frame. This procedure is called concealment of lost frames. It causes a divergence between the decoder generated signal and that sent by the coder. This divergence translated into bad audio quality.

Our work exploited the late-arrival packets and used their content to update the internal decoder states. This helped minimizing the error propagation to a great deal, while maintaining a better quality.

6.2.2 Error propagation after concealment:

As we see in Figure 6.1, signal 4, the divergence from the original signal caused by a lost frame resulted in an error propagation that spanned into several frames. In Figure 6.1, signal 5, the error propagation was reduced to one frame following the concealment by means of our embedded update technique. This improved very much the robustness of the G.729 decoder to possible frame losses and produced a better quality than standard G.729 concealment technique.

Figure 6.1, signal 4, also shows that the error propagated to seven frames after the concealment. While on the other hand, when using our proposed update technique, the error propagated to only one frame after the concealment (see Figure 6.1, signal 5). This is made true by using the information carried in the frame that arrived late. The late frame helped correcting the internal decoder states that diverted from the normal expected values. The result is relatively quick and almost perfect. This is due to the fact that the recovery to a normal signal and the error propagation lasted only one frame after the concealment.

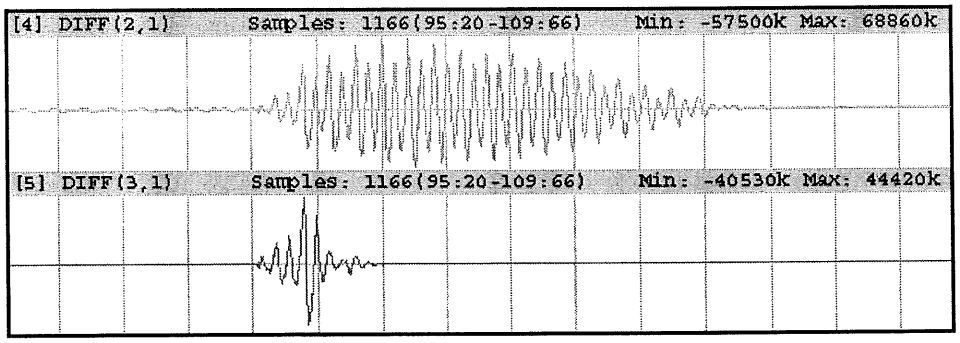


Figure 6.1 Error propagation after concealment. Signal 4: G.729 standard concealment. Signal 5: proposed new update technique

6.3 Subjective results:

6.3.1 AB test description:

As mentioned earlier, the AB tests were conducted on three male and three female speech files. Four conditions were applied to each speech file. We tested the same condition with the G.729 standard (STD) concealment and the G.729 standard concealment with the embedded update (UPD) technique. In other words, the test was a pairwise comparison between STD and UPD. The listener had the chance to choose between STD, UPD or Equal in case the audio quality was the same.

The bit-rate of the audio files was fixed at 8 kbit/s and the DTX was disabled. We considered one frame loss on a periodic basis and varying according to condition. The losses were synchronized on both decoders. The speech files, conditions and presentation orders were randomized. The AB test was conducted by 11 listeners.

The four test conditions were set as follows:

- Condition 1:
  - STD: One lost frame on the 10^{th} frame
  - UPD: One late frame on the 10^{th} frame
- Condition 2:
  - STD: One lost frame on the 15^{th} frame
  - UPD: One late frame on the 15^{th} frame
- Condition 3:
  - STD: One lost frame on the 20^{th} frame
  - UPD: One late frame on the 20^{th} frame
- Condition 4:
  - STD: One lost frame on the 25^{th} frame
  - UPD: One late frame on the 25^{th} frame

6.3.2 AB test showing improvement:

The percentage of preference for each condition is given in table 6.1 for the male speech files and in table 6.2 for the female speech files. The average of the four conditions in also listed in the table.

The UPD technique received an average of 75% of votes for the male speaker, while an average of 79% as the result for the female speaker. On the other hand, the STD technique received an average of 9% of votes for the male speaker and 6% for the female speaker. The listeners also gave an average of 16% and 15% of votes as Equal to the male and female speakers respectively.

The results clearly favour the update (UPD) technique to the standard (STD) technique. The listeners' votes show a strong preference toward the modified decoder with the embedded update after concealment.

It should be noted that the results of the AB test should be considered as general and not highly precise. A more sophisticated test like MOS (Mean Opinion Score) would yield more accurate results.

TABLE 6.1 MALE SPEECH FILES AB TEST RESULTS

Male	Condition 1	Condition 2	Condition 3	Condition 4	Average
UPD	65%	73%	79%	82%	75%
STD	11%	12%	9%	6%	9%
Equal	24%	15%	12%	12%	16%

TABLE 6.2 FEMALE SPEECH FILES AB TEST RESULTS

Female	Condition 1	Condition 2	Condition 3	Condition 4	Average
UPD	73%	77%	80%	85%	79%
STD	9%	6%	6%	3%	6%
Equal	18%	17%	14%	12%	15%

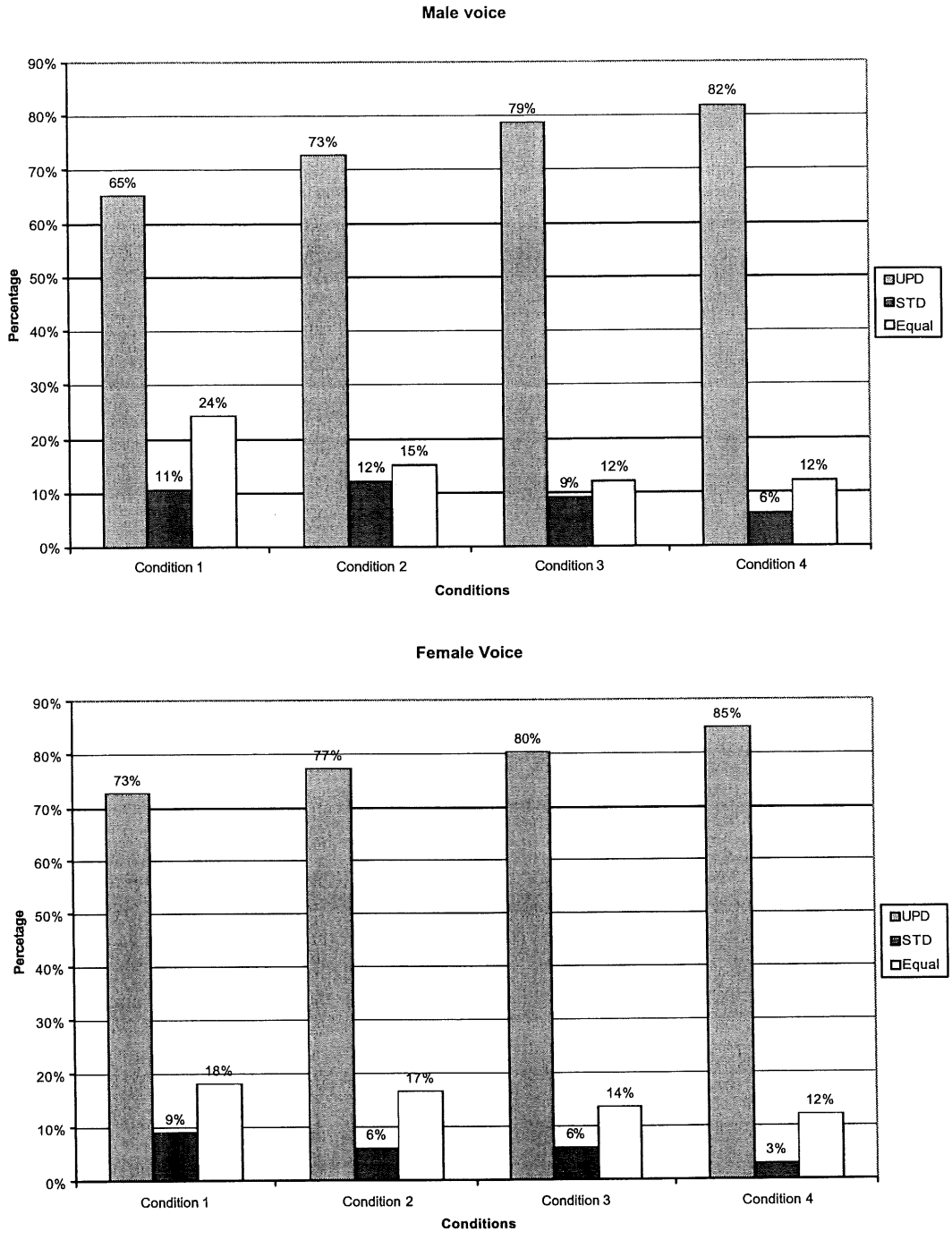


Figure 6.2 Male and Female AB Test results charts

CONCLUSION

VoIP applications are part of our everyday life. Whether Internet chatting applications (such as Netmeeting, Yahoo Messenger, etc.), or long-distance services (such as prepaid cards) provided by international telephony carriers. The Internet architecture was not designed for speech, audio and video applications. Thus arises need to improve the QoS. This can be realized by introducing new concealment and recovery techniques that would further enhance VoIP applications.

We tested a new technique to improve the recovery of a speech decoder after the reception of a late frame. Instead of considering a late frame as lost, we implemented a new technique to use the important information carried in the late frame to update the internal states of the decoder. The result was impressive and would be highly recommended for use in VoIP applications.

This technique was implemented on a G.729 coder, but it would also be applicable to any other predication-based speech, audio or video codec. Making use of late frames increase the robustness of the decoder against unpredictable jitter variations without affecting the overall end-to-end delay. Moreover, making use of late frames allows the receiver to operate with a shorter playout delay without degrading the speech quality.

This work can be further enhanced by conducting feasibility studies and tests to make the decoder robust in case of loosing several consecutive frames. This will introduce a certain complexity on the memory requirements and the processing power of the decoder.

REFERENCES

[1] GIBSON J. D., *Multimedia Communications,* Academic Press, USA, 318 p., 2001

[2] GOURNAY P., ROUSSEAU F., LEFEBVRE R., *Article: Improved packet loss recovery using late frames for prediction-based speech coders,* Speech and Audio Research Group, University of Sherbrooke, Sherbrooke, Canada, 4 p., April 2003

[3] GOURNAY P., *Document : Mise à jour des états internes d'un décodeur de parole de type CELP lors de l'arrivée tardive d'une trame,* Groupe de recherche sur la parole et l'audio, Université de Sherbrooke, Sherbrooke, Canada, 10 p., Mars 2002

[4] IETF NETWORK GROUP, *User Datagram Protocol,* RFC 768, 1980

[5] IETF NETWORK GROUP, *Internet Protocol,* RFC 791, 1981

[6] IETF NETWORK GROUP, *Transmission Control Protocol,* RFC 793, 1981

[7] IETF NETWORK GROUP, *IPv6 specifications*, RFC 1883, 1996

[8] IETF NETWORK GROUP, *Transport Protocol for Real-Time Applications*, RFC 1889, 1996

[9] IETF NETWORK GROUP, *RSVP functional specification,* RFC 2205, 1997

[10] IETF NETWORK GROUP, Robust Header Compression, Internet Draft, 2002

[11] IETF NETWORK GROUP, *Media Gateway Control Protocol,* RFC 3435, 2003

[12] IN COLLABORATION, *Whitepaper: Understanding Voice over Packet,* Spectrum Signal Processing Inc., Vancouver, B.C., Canada, 21 p., 2000

[13] ITU-T SG 15, *Recommendation: Coding of speech at 8 kb/s using CS-CELP,* 39 p., March 1996

[14] KLEIJN W. B., PALIWAL K.K., *Speech coding and synthesis,* Elsevier, 739 p., 1995

[15] MARKET DEV. & EDUCATION COMMITTEE, *Whitepaper: A Discussion of Voice over Frame Relay,* Frame Relay Forum, 12 p., August 2000

[16] PAN P. AND SCHULZRINNE H., *Document: YESSIR, a simple reservation mechanism for the Internet,*11 p., 1998

[17] TANENBAUM A. S., *Computer Networks,* Prentice Hall, USA, 813 p., 1996

[18] XENER SYSTEMS, *Web page: Difference among H.323, SIP, MGCP, and MEGACO/H.248, http://www.xener.com/customer/faq02.html*

ANNEX A

ACRONYMS AND ABBREVIATIONS

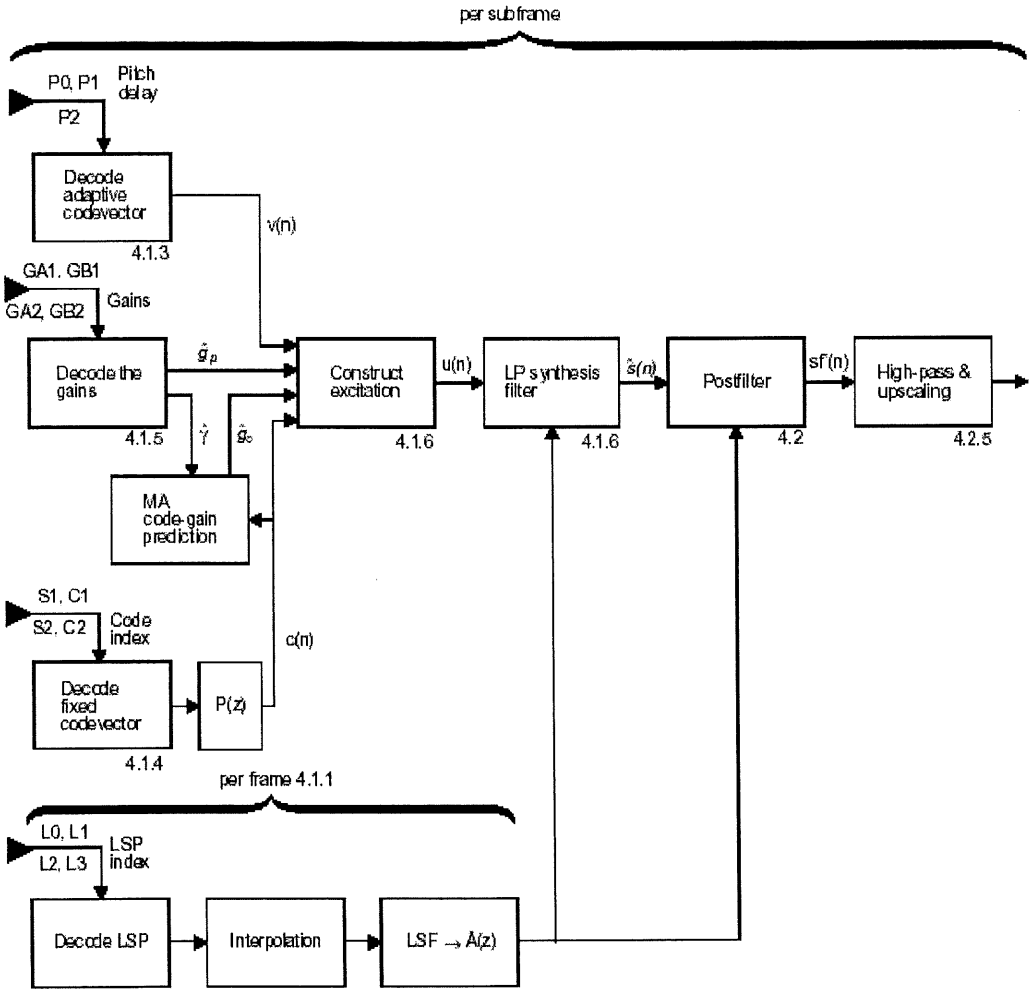
ADPCM	Adaptive Differential Pulse Code Modulation
ATM	Asynchronous Transfer Mode
ACELP	Adaptive Code-Excited Linear Prediction
AMR-WB	Adaptive Multi-Rate - Wideband
BFI	Bad Frame Indicator
CELP	Code-Excited Linear Prediction
CODEC	Coder/Decoder
CRC	Cyclic Redundancy Check
CRTP	Compressed RTP
CS-CELP	Conjugate-Structure – Algebraic Code-Excited Linear Prediction
DSP	Digital Signal Processing
DNS	Domain Name Server
EFR	Enhanced Full-Rate
ETSI	European Telecommunications Standard Institute
FEC	Forward Error Correction
FER	Frame Error Rate
FR	Frame Relay
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
IP	Internet Protocol
IP-NG	Internet Protocol – Next Generation
IHL	Internet Header Length
IPDC	Internet Protocol Device Control
ISDN	Integrated Services Digital Network
ISO	International Standards Organisation

ITU-T	International Telecommunications Union – Telecommunications
LAN	Local Area Network
LLC	Logical Link Control
LP	Linear Predication
LPC	Linear Predication Coding
LSP	Line Spectrum Pairs
MAC	Media Access Control
MG	Media Gateway
MGCP	Media Gateway Control Protocol
MEGACO	Media Gateway Control
MMUSIC	Multiparty Multimedia Session Control
MOS	Mean Opinion Score
NNTP	Network News Transport Protocol
OSI	Open System Inter-connection
PCM	Pulse Code Modulation
PSTN	Public-Switched Telephone Network
QoS	Quality of Service
ROHC	Robust Header Compression
RSVP	Resource ReSerVation Protocol
RTP	Real-time Transport Protocol
RTCP	Real-time Control Protocol
RTSP	Real-time Streaming Protocol
SG	Study Group
SGCP	Simple Gateway Control Protocol
SMTP	Simple Mail Transport Protocol
SIP	Session Initiation Protocol
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TI	Texas Instruments
TIA	Telecommunications Industry Association
TIPHON	Telecommunications and Internet Protocol Harmonization Over Networks

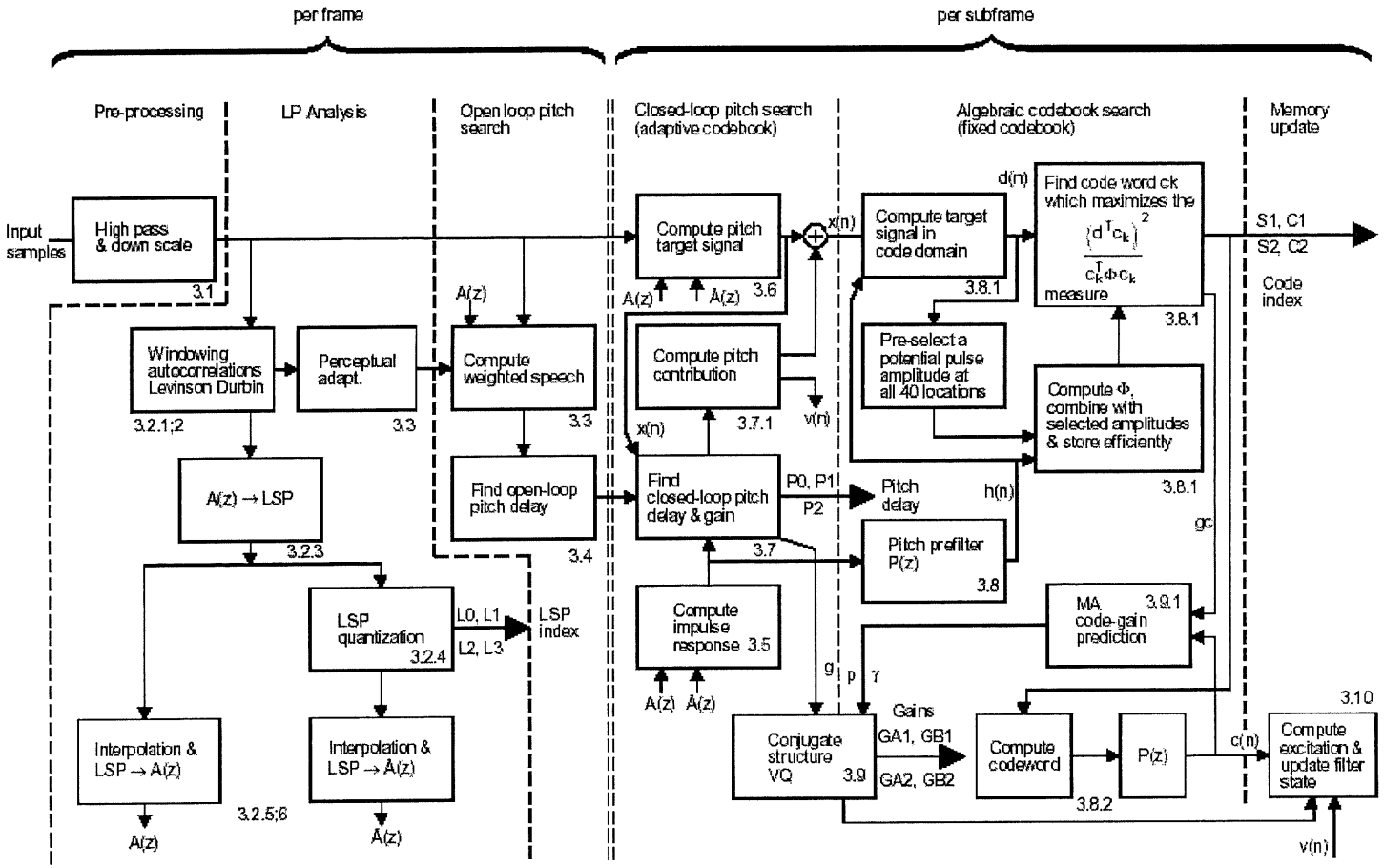
TTL	Time to Live
VAD	Voice Activity Detection
VDC	Venture Development Corporation
VoIP	Voice over Internet Protocol
VQ	Vector Quantization
UDP	User Datagram Protocol
WAN	Wide Area Network
YESSIR	Yet Another Sender Session Internet Reservations

ANNEX B

DECODER/ENCODER FLOWCHART



Signal flow at the CS-ACELP decoder



Signal flow at the CS-ACELP encoder