INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

Packet Loss Concealment for Voice Transmission over IP Networks

Ejaz Mahfuz

Department of Electrical Engineering
McGill University
Montreal, Canada

September 2001

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Engineering.

2001/09/27

Abstract

Voice-over-IP (VoIP) uses packetized transmission of speech over the Internet (IP network). However, at the receiving end, packets are missing due to network delay, network congestion (jitter) and network errors. This packet loss degrades the quality of speech at the receiving end of a voice transmission system in an IP network. Since the voice transmission is a real-time process, the receiver cannot request for retransmission of the missing packets. Concealment algorithms, either transmitter or receiver based, are used to replace these lost packets. The packet loss concealment (PLC) techniques described in the standards ANSI TI.521 (Annex B) and ITU-T Rec. G.711 (Appendix I), have good performance, but these algorithms do not use subsequent packets for reconstruction. Furthermore, there are discontinuities between the reconstructed and the subsequent packets, especially at the transitions from voiced to unvoiced and phoneme to phoneme.

The goal of this work is to develop an improved PLC algorithm, using the subsequent packet information when available. For this, we use the Time-Scale Modification (TSM) technique based on Waveform Similarity Over-Lap Add (WSOLA) to reconstruct the dropped or lost packets. The algorithm looks ahead for subsequent packets. If these packets are not available for reconstruction, algorithm uses information from past packets. Subjective tests show that the proposed method improves the reconstructed speech quality significantly.

Sommaire

La transmission de la voix sur l'Internet (réseau IP) se fait par transmission de paquets. Au récepteur, certains paquets manquent dû aux délais, à la congestion ou aux erreurs de transfert. Cette perte de paquets dégrade la qualité de la voix au récepteur d'un système de transmission IP. Étant donné que la transmission de la voix est effectuée en temps réel, le récepteur ne peut pas requérir à la retransmission des paquets perdus à cause des délais de transferts trop importants. Des algorithmes de dissimulation des pertes (concealing) sont utilisés au niveau de l'émetteur ou au niveau du récepteur afin de combler la perte des paquets. Les techniques de dissimulation des pertes (concealing) des paquets des normes ANSI TI.521 (annexe B) et ITU-T G7.11 (Annexe I) offrent une bonne performance. Ces algorithmes ne prennent cependant pas en considération le contenu des paquets à venir lors de la reconstruction de la voix. C'est pourquoi les discontinuités se font valoir entre les paquets reçus et les paquets à venir, particulièrement lors des transitions entre la parole voisée et non voisée ainsi qu'entre les phonèmes consécutifs.

L'objectif de notre recherche est d'améliorer les algorithmes de dissimulation des pertes en prenant en considération, dans la mesure du possible, le contenu des paquets à venir. Pour ce faire, nous utilisons l'altération de l'échelle du temps (TSM) basé sur l'ajout en chevauchement des similarités des ondes (WSOLA) afin de reconstruire les paquets perdus. L'algorithme inspecte les paquets à venir. Si ces paquets ne sont pas disponibles lors de la reconstruction, l'algorithme utilise l'information contenue dans les paquets antérieurs. Les résultats des essais subjectifs formels démontrent que la méthode proposée améliore considérablement la qualité de la voix reconstruite.

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Professor Peter Kabal for his support and guidance throughout my graduate studies at McGill University. His vast knowledge, patience and valuable advice helped me to accomplish this work successfully. The financial support provided by Prof. Kabal is also gratefully appreciated.

I am thankful to all my fellow graduate students in the Telecommunications and Signal Processing Laboratory for their companionship, fruitful suggestions, proof reading and participation in listening tests for this research. Special thanks go to Tarun Agarwal, Mark Klein, Aziz Shallwani, Paxton James Smith, and M. Mahbulbul A. Khan. Moreover, I wish to acknowledge Benoît Pelletier who helped me with the French abstract.

I am obliged to my wife Tamanna and my son Daiyaan for their love and continuous support. I am also thankful to our families for their best wishes and encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The transmission of voice over packet switched networks, such as an IP (Internet Protocol) network (like the Internet), is an area of active research. Much of the past work focused on using packet switching for both voice and data in a single network. Renewed interest in packet voice, and more generally, packet audio applications has been fuelled by the availability of supporting hardware, increased bandwidth throughout the Internet and the desire to integrate data and voice services in the networks.

The motivation for transporting voice over IP networks is the potential cost saving achievable by eliminating or bypassing the circuit-switched telephony infrastructure. PC-based programs such as NeVoT (Network Voice Terminal) [1], RAT (Robust Audio Tool) [2], Free Phone [3] and MSN messenger service have demonstrated the feasibility of voice transport over the Internet. There is now a desire for wider deployment of VoIP using stand alone terminals.

In a Voice over IP (VoIP) application, the voice is digitized and packetized at the sender at regular intervals (e.g., every 10 ms) using an encoding algorithm. The voice packet is then sent over the IP network to the receiver where it is decoded and played-out to the listener.

IP networks (e.g., the Internet) are inherently best-effort networks with variable delay and loss (packets do not arrive at all to the receiver). Voice traffic can tolerate some packet loss, where lost packets are replaced by zeros. However, if the packet loss rate is greater than 5%, it is considered harmful to the voice quality [4] and a good concealment technique is required for reconstruction of the lost packets. The maximum packet loss rate

and required concealment algorithm can depend on the nature of the encoding algorithm and on the sampling rate of the voice stream. The length of a phoneme (smallest meaningful contrastive unit in the phonology of a language [5]) is typically between 80 to 100 ms [6]. When the duration of the packet loss is greater than the length of a phoneme it can change the meaning of a word.

VoIP applications use UDP (User Datagram Protocol) as the transport layer protocol. The Real Time Protocol (RTP) [7] is used to provide additional functionality, adding sequence numbers and timestamps. The Real Time Control Protocol (RTCP) [7] is also employed to return receiver statistics (e.g., the number of packets detected as lost) to the sender. RTCP packets are sent every 5 seconds by a receiver to a sender and consume very little bandwidth.

For audio quality in packet audio applications, the main concerns are delay and loss. In an IP network, delay and loss are not known in advance since they depend on the behaviour of other connections throughout the network. A variety of audio conferencing tools have been available for a few years, and they have been used to audiocast conferences. Experimental evidence suggests that, although the quality of audio delivered by the Internet tools has improved, the quality is still mediocre in many audio conferences. This is clearly a concern since the audio quality has been found to be more important than the video quality or audio/video synchronization to successfully carry out collaborative work.

Real-time voice applications have an upper bound on tolerable end-to-end delay (from transmitter–receiver or receiver–transmitter). For interactive voice applications, the maximum allowable end-to-end delay is between 250 to 500 ms [8]. In earlier studies [9] it was noted that in LANs and campus networks, where the network caused delay, delay variance and losses were relatively small; most of the end-to-end delay was accumulated in the terminals.

In terminals, delay is introduced by hardware and software. In audio hardware, voice samples pass through an A/D (Analog to Digital) converter at the sender and D/A (Digital to Analog) converter at the receiver which introduces delay. In packet audio applications, software processing delay is also introduced. Processing delay is very much dependent on the speech codec (COder and DECoder) used. Some codecs, like PCM (Pulse Code Modulation) codec have low complexity and introduce little delay, whereas others, such as the GSM (Global System for Mobile Communications) codec, require excessive computations and cause significantly more processing delay. The buffering of voice samples is necessary

both at the sending and receiving end. Buffering delays are introduced both in the audio hardware and in the packet audio software. Delays are also introduced by the operating system because it has to serve other processes that are simultaneously running in the terminal.

Variable delays and loss rates in the network have to be smoothed out in the application software in order to preserve the sound quality. Voice packets are buffered at the receiver and are played-out periodically. The algorithms used to calculate the appropriate playout time for each packet of voice are called playout algorithms.

1.1 Motivation and Scope

Today's packet switched networks are not only used for data transfer, but also for audio and video transmission. An example is the MBone (Multicast Backbone) overlay network in the internet. In the case of waveform coded audio, packet loss causes signal drop-outs (no information about the content or the character of the signal, and considered to be zero), which are very annoying for the listener at the receiver. In order to achieve high quality real-time voice transmission, an effective packet loss concealment mechanism must be employed to alleviate the problems of loss and delay in the internet.

Existing software-based loss concealment mechanisms can be classified into two categories:

- Receiver based.
- Sender (Transmitter) and Receiver based.

In simple receiver-based reconstruction schemes, lost packets are recreated by padding silence (substituting samples with zeros) or white noise [10]. More sophisticated reconstruction schemes substitute the missing signal segments by repeating a prior segment (samples received as good). The pattern matching [11] technique repeats a correctly received signal segment, of which maximal similarity with the lost segment is assumed. This is accomplished by matching a sample pattern with the series of samples received just before the gap. As entire signal segments of at least one packet duration are completely repeated, this may cause an echo. To avoid this, a Pitch Waveform Replication [11] mechanism is applied to reconstruct the signal by repeating only one pitch period throughout the missing packet.

A phase matching technique extended to pitch replication provides synchronization at both edges of the substituted signal, reducing clicking distortions caused by the described methods [12]. In the case of Pitch Waveform Replication and Phase Matching, the multiple repetition of the same small signal segment can cause tinny sounds. All of these strategies only work well for low and infrequent losses. Also the perceived quality drops significantly with an increased length in the lost segment.

Sender and receiver-based reconstruction schemes are usually more effective but more complex. A common way is for the sender to first process input streams in such a way that the receiver can better reconstruct missing data. Based on different ways of processing input data, these schemes can further be split into those that add redundant control and those that do not. There are several methods for the sender to add redundancy. These include sending duplicate packets [13], or sending past packets coded in lower bit rates along with current ones, or sending error correction bits in voice packets using forward error correction(FEC) [14] [15]. All these methods require extra bandwidth or imply long end-to-end delay. The algorithms that do not add redundancy, utilize inherent redundancies in the source voice stream. A typical method interleaves voice samples into distinct packets and reconstructs lost samples by interpolation using their surviving neighbours. The simplest form is two-way interleaving that packetizes odd and even number samples separately [4], and interpolates lost samples by simple averaging in case one of the packets is lost [10].

The described packet loss concealment techniques, as well as the techniques in the standards T1.521-1999 [12] and T1.521a-2000 [16] (described in details in chapter 5), do not consider subsequent packet information to reconstruct the missing packets.

1.2 Our Approach

The objective of our research is to build a new receiver based reconstruction algorithm. In this aspect, we have used subsequent packet information (the good packet after the lost packet, if it is available) for the reconstruction of lost packets. Subsequent packet (also referred to as 'future packet') information is very useful for reconstruction, especially at the transition between speech segments, such as voiced-to-unvoiced, unvoiced-to-voiced or phoneme-to-phoneme. If the future packet is not available, reconstruction is done based on received packets just before the lost packets. We perform the time-scale modification (changing the duration of the speech signal without changing the pitch period information)

of the speech signal using WSOLA (Waveform Similarity Overlap-Add) technique [17] to fill in the lost packets (the details are given in chapter 5). In this research, time-scale modification has been used to stretch the signal; preserving the same speaking rate to generate the missing samples.

A good time-scale modification (TSM) algorithm is one that produces "natural-sounding" after changing the duration (such as expanding the duration) of speech segments. Intelligibility, tonal quality, and speaker recognition should be preserved, and processing artifacts (pops, clicks, burbles, reverberation, etc.) should be kept to a minimum. WSOLA-based time-scale modification technique has the potentiai to meet all these requirements.

The reconstruction technique for lost voice packets in an IP network described in [18] uses WSOLA. Only the subsequent packets were used for reconstruction. A packet size of 160 samples was considered. Three consecutive future packets were used for reconstruction. The performance of the algorithm is good, except for higher packet loss rates (above 20%) and when the losses are at the signal transitions. When the packet loss rate is high (this means when there are 3 to 4 consecutive packet loss) the PLC algorithm creates artifacts, reduces intelligibility and introduces an excessive delay as the algorithm waits for the future packets to arrive. Moreover, if two consecutive packets are missing, four subsequent packets are needed to recover the lost packets. Also, the algorithm does not guarantee a better reconstruction if any of the subsequent packets are missing due to long network delay. As described in [18], it is not only necessary to keep a copy of a certain number of packets, but also to withhold these packets from the playout buffer. It has also been described in [16], that the packet loss reconstructed scheme described in the standard T1.521a-2000 (Annex B) has better performance than the reconstructed scheme described in [18].

In our approach, we have considered one future packet (if it is available to use) and three past packets for reconstruction (the packet size is 80 samples, but it can support other packet sizes). Past packets' information is used along with the future packet. The last 20 samples in the playout buffer need to be withheld from playing-out for overlap-add with the reconstructed signal. We have used a time-scale modification technique to reconstruct a signal with a pitch period close to the original signal. WSOLA can perform time-scale modification (expansion) in real-time, preserving the natural sound. WSOLA is low complexity and can even be applied to the residual signal or to the original signal. If there are any discontinuities at the boundary, WSOLA performs well to reduce them, making them less noticeable. We have applied this time-scale modification technique on

the residual signal (filtered version of the original signal using linear prediction analysis) as well as on the original signal to reconstruct the packets which are lost or delayed.

1.3 Thesis Organization

The motivation of this research is to introduce an efficient, robust and good quality (which does not create unnatural artifacts and maintains intelligibility) packet loss concealment technique for voice transmission in an IP network. In Chapter 2, the background is provided about the use of IP networks for voice transmission, and the advantages and disadvantages using IP (Internet) for voice transmission. In the next chapter (Chapter 3), the Linear Prediction analysis, synthesis and interpolation of LP coefficients are discussed. Chapter 4, provides detail about the time-scale modification of the speech signal, based on OLA (OverLap-Add), SOLA (Synchronized OverLap-Add) and WSOLA. This chapter shows how the time domain algorithm modifies the signal while preserving the pitch period information. The drawbacks of OLA and SOLA techniques for time-scale modification in real-time are also discussed. In Chapter 5, the packet loss concealment standard ANSI T1.521a-2000 (Annex B) and the proposed algorithm are discussed. Also, the subjective test results and the improvements achieved of the proposed algorithm compared to the existing PLC standard, are provided in the same chapter. Chapter 6, summarizes the work and gives suggestions about future investigations.

Chapter 2

Transmission of Speech Signals over IP Networks

In the modern telecommunications world, the recent trend is to replace circuit switched networks, such as the PSTN (Public Service Telephone Network) designed for voice transmission, with packet switched networks like the Internet. This research is 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. A new algorithm is developed to improve speech quality. This chapter gives an overview of the applications, benefits, protocols, issues, and implementation of a voice-over-IP system.

2.1 Introduction

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, such as Internet phone. These real-time applications have quite different characteristics and requirements.

The first significant characteristic of real-time applications is high delay sensitivity. Given strict end-to-end delay and interframe delay requirements for real-time transmissions, packets delayed over a certain time limit are considered lost and cannot be retransmitted by the sender. Retransmission is not a viable option for real-time voice. The current internet does not support real-time data transmission because it has no special delivery mechanism

to differentiate between real-time data and non real-time data. Hence, all real-time data frames are treated the same way as non real-time frames and will be dropped or delayed with equal chance under heavy load and congestion. The current Internet also may have large delay variations and loss. The loss rate of packets to some destinations can be as high as 50% [19] [20].

The second significant characteristic is that most real-time applications do not require data to be 100% precise, unlike services provided by Transmission Control Protocol (TCP), which ensure that all data packets are sent correctly and reliably all the time. This characteristic is very useful because the receiver can tolerate a certain level of loss or distortion of data without significant degradation in performance. TCP can be used over an IP to increase network service reliability.

The above two characteristics define the potential problems that should be considered in order to develop a high quality, real-time voice transmission system. Reliability and predictability are two major problems [21]. Reliability ensures the reliable delivery of voice packets so that packet loss is concealed from users, whereas predictability ensures the timely delivery of voice packets.

In this research, a new reconstruction method which improves the speech quality in voice transmission over IP networks has been developed to conceal the lost packets.

2.2 Voice over IP Benefits and Applications

VoIP can be achieved on any data network that uses IP, such as Internet, Intranet and Local Area Networks (LAN). Today, packet networks are growing at a much faster rate than voice networks. Of late, there has been a growing interest in transporting voice over packet networks, for the following reasons.

- Demand for low cost (VoIP has a major potential for being a low cost alternative to PSTN).
- Demand for multimedia communication.
- Demand for integration of voice and data networks (VoIP has the potential of replacing the telephone network with an integrated network, capable of supporting both voice and data service over a common infrastructure).

Voice over IP allows telephone calls over the IP network, at a cost much lower than traditional telephone networks. A wide variety of applications is possible by the transmission of voice over packet networks. Few examples of these applications are:

- *Integration*: A network configuration of an organization with many branch offices that wants to reduce costs and combine traffic to provide voice and data access to the main office. This is accomplished by using a packet network to provide standard data transmission while at the same time enhancing it to carry voice traffic along with the data [22].

- *Voice over packet network in a trunking application*: In this scenario, an organization wants to send voice traffic between two locations over the packet network and replace the Tie Trunks used to connect the PBXs (Private Branch Exchange) at the locations. This application usually requires the Interworking Function (IWF) to support a higher capacity digital channel than the branch application, such as a T1/E1 interface of 1.544 or 2.048 Mbps. The Interworking Function emulates the signaling functions of a PBX, resulting in significant savings in communication costs [22].

- *Voice over packet software interworking with Cellular Networks*: The voice data in a digital cellular network is already compressed and packetized for transmission over the air by the cellular phone. Packet networks can then transmit the compressed cellular voice packet, saving a tremendous amount of bandwidth. The IWF provides the transcoding function required to convert the cellular voice data to the format required by the PSTN [23].

Even though VoIP presents a tremendous opportunity, it does not provide the toll quality voice over the networks; hence, achieving toll quality voice over IP is a major challenge. Major issues revolve around the quality of voice calls as well as the ease of use for the end user. However, significant progress has been made in this respect to enable packet networks to provide toll quality voice.

2.3 Internet Protocol (IP)

2.3.1 Overview of IP

Internet Protocol (IP) is a connectionless protocol in which packets can take different paths between the endpoints and paths that are shared by packets from different transmissions. It permits the exchange of traffic between two host computers without any prior call setup. This enables the efficient allocation of network resources, as packets are routed on the paths with the least congestion. Header information in the packets make sure that the packets reach their intended destinations. It is possible that the datagrams (IP user data) could be lost between the two end-user's stations. For example, the IP gateway enforces a maximum queue length size; if this queue length is violated, the buffers will overflow. In this situation, some datagrams are discarded by the network.

IP hides the underlying subnetwork from the end user. In this context, it creates a virtual network to that end user. This aspect of IP is quite attractive because it allows different types of network to attach to an IP node. IP is simple to install and because of its connectionless design, is quite robust, because the datagrams can route through different network paths to the destination.

Since IP is a best-effort datagram type protocol, it has no retransmission mechanisms. It provides no error recovery for the underlying subnetworks. The user data (datagrams) may be lost, duplicated, or even arrive out of order. It is not the job of IP to deal with most of these problems. Most of these problems are passed to the next higher protocol layer, where TCP (Transport Control Protocol) controls the flow of datagrams.

These low-level characteristics of IP translate into a fairly effective means of supporting real-time voice traffic. Assuming the routers are fast, and sufficient bandwidth is available, IP does not introduce significant overhead to the support of VoIP. There are better mechanisms but no other mechanism has the universal presence of IP (and the IP address).

Fig. 2.1 shows how IP processes an incoming IP datagram [24]. The incoming packet is stored in a queue to await processing. Once processing begins, the options field is processed to determine if any options are in the header (the support for this operation varies). The datagram header is checked for any modifications that may have occurred during its journey to the IP node. Next, it is determined if the IP address is local; if so, an IP protocol ID field in the header is used to pass the bits in the data field to the next module, such as TCP, UDP (User Datagram Protocol), ICMP (Internet Control Message Protocol), etc.

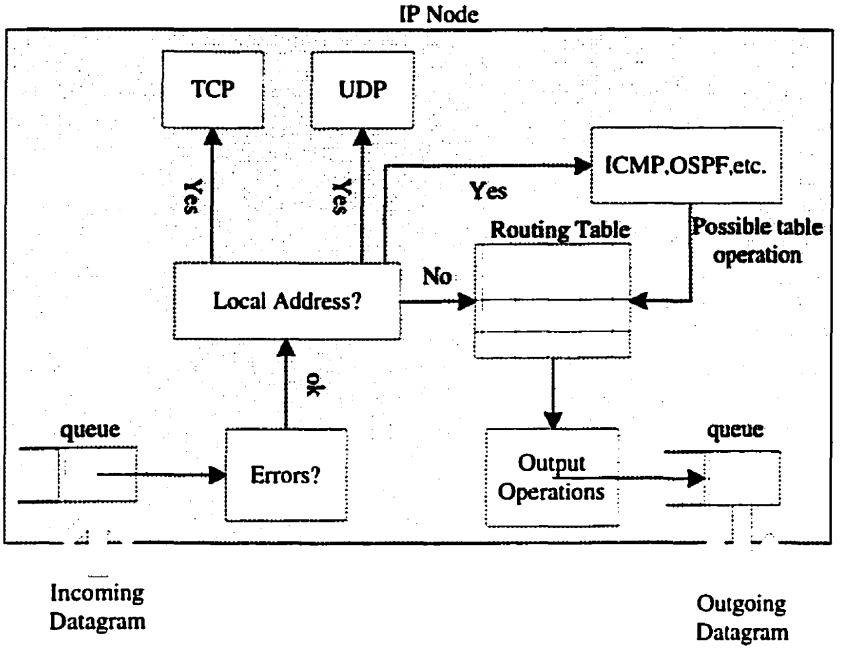


Fig. 2.1 Processing an IP datagram.

An IP node can be configured to forward or not to forward datagrams. If the node is a forwarding node, the IP destination address in the IP datagram header is matched against a routing table to calculate the next node (next hop) that is to receive the datagram. If a match to the destination address is found in the table, the datagram is forwarded to the next node; otherwise, it is sent to a default route, or it is discarded. Fig. 2.2 is an example of a typical routing table found in a router. Individual systems differ in the contents of the routing table, but they all resemble this example. The entries in the table are:

- Destination: IP address of the destination.
- Route Mask: Mask that is used with the destination address to identify bits that are used in routing.
- Next Hop: IP address of the next hop in the route.
- If (Interface) Index (port): Physical port on the router to reach the next hop address.
- Metric: "Cost" to reach the destination address.

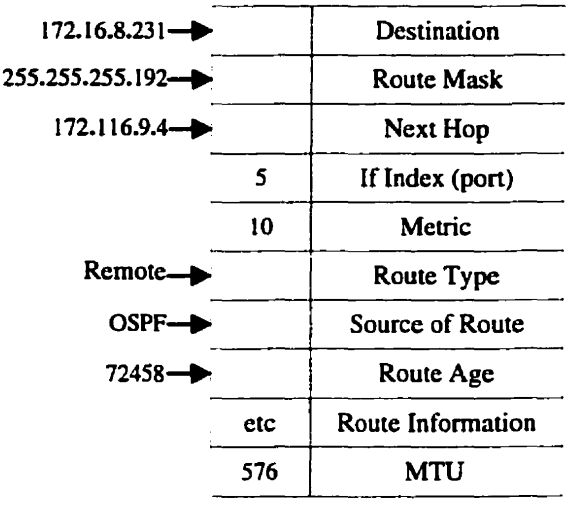


Fig. 2.2 Typical IP routing table.

- Route Type: Directly attaches to router (direct), or determines whether the node is reached through another router (remote).
- Source of Route: How the route was discovered.
- Route Age: In seconds, since the route was last updated.
- Route Information: Miscellaneous information.
- MTU: Size of payload.

2.3.2 IP Datagram

The structure of an IP datagram is depicted in Fig. 2.3. The *version* field identifies the version of an IP in use. Most protocols contain this field because some network nodes may not have the latest release available of the protocol. The current version of IP is 4.

The *header length* contains four bits which are set to a value to indicate the length of the datagram header. The length is measured in 4 octets. Typically a header without QoS (Quality of Service, discussed later in this chapter) options contains 20 octets. Therefore, the value in the header length field is usually 5.

The *type of service (TOS)* field can be used to identify several QoS functions provided for Internet applications. The size and entries to this field varies, depending on QoS provided.

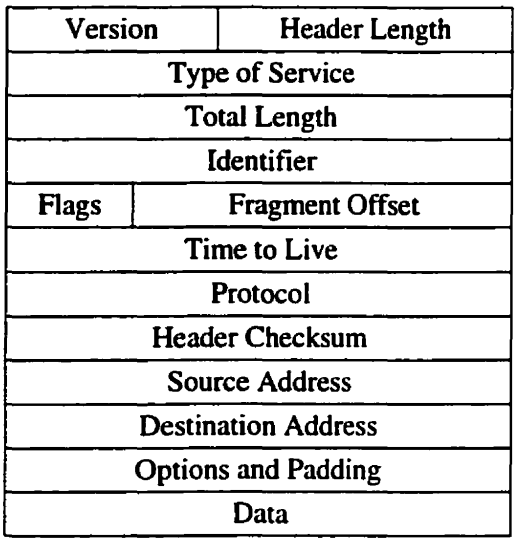


Fig. 2.3 The IP datagram.

It is quite similar to the service field that resides in the Open System Interconnection (OSI)-based CLNP (Connectionless Network Protocol) PDU (Protocol Data Unit)[1]. Transit delay, throughput, precedence, and reliability can be requested with this field.

Typically, the TOS field contains five entries consisting of 8 bits. Bits 0, 1, and 2 contain a precedence value which is used to indicate the relative importance on the datagram. Values range from 0 to 7, with 0 set to indicate a *routine precedence*. The precedence field is not used in most systems, although the value of 7 is used by some implementations to indicate a network control datagram. However, the precedence field could be used to implement flow control and congestion mechanisms in a network. This would allow gateways and host nodes to make decisions about the priority of "throwing away" datagrams in case of congestion.

The next three bits are used for other services and are described as follows: bit 3 is the *delay bit (D bit)*. When set to 1, this TOS requests a short delay through the Internet. The aspect of delay is not defined in the standard and it is up to the vendor to implement the service. The next bit is the *throughput bit (T bit)*. It is set to 1 to request for high throughput through the internet. Again, its specific implementation is not defined in the standard. The next bit used is the *reliability bit (R bit)*, which allows a user to request

[1]Fields in an IP datagram are also called Protocol Data Units

high reliability for the datagram. The last bit of interest is the *cost bit (C bit)*, which is set to request the use of a low-cost link. The last bit is not used at this time.

The *total length* field specifies the total length of the IP datagram. It is measured in octets and includes the length of the header and the data. IP subtracts the header length field from the total length field to compute the size of the data field. The maximum possible length of a datagram is $2^{16} - 1 = 65,535$ octets. Gateways that service IP datagrams are required to accept any datagram that supports the maximum size of a PDU of the attached networks. Additionally, all gateways must accommodate datagrams of 576 octets in total length.

The IP protocol uses three fields in the header to control datagram fragmentation and reassembly. These fields are the *identifier*, *flags*, and *fragmentation offset*. The *identifier* field is used to uniquely identify all fragments from an original datagram. It is used with the source address at the receiving host to identify the fragment. The *flags* field contains bits to determine if the datagram is fragmented, and if fragmented, one of the bits can be set to determine if this fragment is the last fragment of the datagram. The *fragmentation offset* field contains a value which specifies the relative position of the fragment to the original datagram. The value is initialized as 0 and is subsequently set to the proper number when an IP node fragments the data. The value is measured in units of eight octets.

The *Time-To-Live (TTL)* parameter is used to measure the time a datagram has been in the internet. It is similar to CLNP's lifetime field. Each gateway in the internet is required to check this field and discard the datagram if the TTL value equals 0. An IP node is also required to decrement this field in each datagram it processes. In actual implementations, the TTL field is a number of hops value. Therefore, when a datagram proceeds through a gateway (hop), the value in the field is decremented by a value of one.

The *protocol* field is used to identify the next level protocol above the IP that is to receive the datagram at the final host destination. It identifies the payload in the data field of the IP datagram. The internet standards group has established a numbering system to identify the most widely used upper layer protocols.

The *header checksum* is used to detect an error that may have occurred in the header. Checks are not performed on the user data stream. The current approach keeps the checksum algorithm in IP quite simple. It does not have to operate on many octets, but it does require that a higher level protocol at the receiving host must perform some type of error check on the user data if it cares about its integrity.

IP carries two addresses in the datagram. These are labelled *source* and *destination addresses* and remain the same value throughout the life of the datagram. These field contain the internet addresses.

The *option* field is used to identify several additional services. It is similar to the option part field of CLNP. The options field is not used in every datagram. The majority of implementations use this field for network management and diagnostics [25].

Fig. 2.4 illustrates the IP network protocols that are currently being used to implement VoIP.

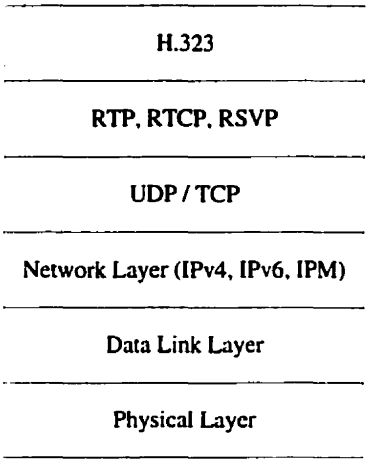


Fig. 2.4 VoIP Protocol Structure.

2.4 Issues in VoIP

2.4.1 Quality of Service (QoS)

The current Internet service model is flat, offering a classless and best-effort delivery service. The biggest problem faced by voice over packet networks is that of providing end users with the quality of service that they get in a traditional telephony network. Unlike the PSTN, where a dedicated end-to-end connection is established for a call, packet based networks use statistical multiplexing of network resources. Although sharing resources amongst multiple users leads to a cost saving (and hence the attraction of voice over packet networks), it does not guarantee the overall quality of service offered to a user. The next generation of

IP, version 6, includes support for the *flow control* of packets between one or more hosts [26]. In conjunction with a hop-by-hop resource reservation protocol such as RSVP [27], end-to-end capacity can be set aside for real-time traffic. There are multiple parameters that determine the quality of service provided by a network. This subsection describes the barriers to the operation of these schemes, including requirements for codecs, bandwidth, delays, delay jitter and the packet loss experienced in a network.

2.4.2 Factors Affecting Quality of Service

Codecs

Internet telephony services must operate in a bandwidth, delay, loss, and cost-constrained environment. This environment has been passed down to the codec (COder and DECoder) development efforts of the ITU-T (International Telecommunication Union Telecommunication standardization sector). Recently ITU codecs, G.711, G.723.1, G.729, and G.729A, [28] [29] [30] [31] have been designed to work well in the presence of these constraints. Although they were originally designed with different applications in mind, they all are candidates for enabling VoIP. Table 2.1 shows the performances of different codecs. Information about these codecs was taken from different sources (including ITU-T, IEEE and [32]).

The Mean Opinion Scores (MOS) rates the quality of sound played-out for subjective listening tests on the standard 5-point absolute category rating (MOS) scale. The opinion, or perceived level of distortion, is mapped into either the descriptive term "unsatisfactory, poor, fair, good, excellent", or the numerical rating 1–5. Note that the MOS score may vary for different test environments (e.g., for different test files, for different subjects).

Delay

The delay experienced in a packet network is classified into the following types: accumulation delay, packetization delay, network delay and propagation delay. Each of these adds to the overall delay experienced by the user. Accumulation delay is caused by a need to collect a frame of voice samples for processing by the voice coder. This delay depends upon the sampling rate and the type of voice coder used. The accumulated voice samples are next encoded into a packet, which leads to the packetization delay. Once this packet is sent through the network, it experiences transmission delay in reaching the destination. This is

Table 2.1 List of standardized speech coders.

Standard (N)	Algorithm (N)	Complexity (MIPS)	Frame Size /lookahead(ms)	Compression	Bit rate (kb/s)	MOS (N)
G.711	PCM	0	0.125/0	1	64	4.10
G.726 G.727	ADPCM	1	0.125/0	4/2.7/2/1.6	16/24/32/40	3.85
G.722	SB-ADPCM	10	0.125/1.5	1.3/1.1/1	48/56/64	3.3
G.728	LD-CELP	30	0.625/0	4	16	3.61
G.729	CS-ACELP	20	10/5	8	8	3.92
G.729A	CS-ACELP	11	10/5	8	8	3.7
G.723.1	MPC-MLQ	16	30/7.5	10.2/12.1	6.3/5.3	3.9
GSM 06.10	RPE-LTP	10	20/0	4.9	13	3.5
IS-54	VSELP	24	20/5	8	8	3.54
IS-96	QCELP	20	20/5	7.5/16/32	8.5/4/2	-
FS-1016	CELP	30	-	13.3	4.8	3.0
FS-1015	LPC10E	15	-	26.7	2.4	2.4

caused because of multiple factors, which includes the processing done by each intermediate node in the network to forward the voice packet, the capacity of the underlying physical medium, etc.

Delay Jitter

In packet-based networks, two packets sent from the same source to the same destination might take different routes through the network. This is because the packets are routed through the network independently. Hence, two packets between the same source and destination might experience different processing delays and different congestion situations in the network, resulting in a variation in the overall delay experienced by the packets. This variation in the delay experienced by the packets is measured as delay jitter. Also, this might lead to packets reaching the destination out of order.

To take care of the delay jitter, a buffering scheme is used at the destination. Packets at the destination are received and buffered. After the buffer is full to a threshold value, the packets are played in sequence and with a constant delay. However, this buffering of packets at the destination leads to an additional delay and also adds to the other three types of delays discussed above.

Packet Loss and Missing Packets

Voice packets routed through an IP network can be lost because of its best-effort nature. To provide reliable transmission of data in an IP network, a retransmission scheme is used at the transport layer, which retransmits any packets for which an acknowledgement is not received from the destination (assuming that the packet got lost). However, the same scheme cannot be applied to voice, as a retransmitted voice packet might reach the destination much later than when it is needed.

Packets arriving late due the delays described above, are discarded at the receiver. There are also packets which are lost due to the network errors and the best-effort nature of IP networks. All these discarded and lost packets are considered as *'missing packets'*, and a good reconstruction algorithm is necessary to fill in these packets.

Echo

Echo occurs as a result of transmitted signals being coupled into a return path and fed back to their respective sources. The returned signal occurs with noticeable delay.

The subjective effect of echo is also a function of delay. On short connections, the delay is small enough that the echo merely appears to the talker as natural coupling in his ear. A telephone is purposely designed to couple some speech energy (called sidetone) in the earpiece. Otherwise, the telephone seems dead to a talker. If the delay is more than about 25 milliseconds, the caller can hear a distinct echo. Hence, long-distance circuits require significant attenuation to minimize echo annoyance because of the long round-trip delay.

Echo affects the talker more than the listener. Due to echo, one can hear one's own voice in the receiver after a delay of more than about 25 ms; this can cause interruptions and break the cadence in a conversation. For example, assume that user 'A' is talking to user 'B'. The speech of user 'A' to user 'B' is called '$s(n)$'. When '$s(n)$' hits an impedance mismatch, or other echo-causing environments, it bounces back to user 'A'. User 'A' can then hear the

delay several milliseconds after user 'A' actually speaks. Since packet networks introduce higher end-to-end delay, and hence have a greater round-trip time, echo cancellation is an essential requirement for a voice over packet network [25].

Talker Overlap

If the end-to-end delay becomes greater than 250 ms, the problem of 'talker overlap' surfaces. This is experienced by one talker overlapping the other talker, and can be extremely annoying. Also, a delay of more than 250 ms feels like a half-duplex connection and cannot be claimed to be an interactive session.

2.5 VoIP Related Standards

The standardization activity of VoIP is being governed by two bodies, namely the ITU-T and the IETF. The following sub-sections elaborate on the standardization effort.

2.5.1 ITU-T Standards

The first set of standards related to VoIP was developed by ITU-T through their H.323 series. This standard, along with other standards developed by ITU-T, are detailed below.

H.323

The dominant standard for transmitting multimedia in packet switched networks is the International Telecommunication Union (ITU) recommendation H.323 [33] [34], which uses IP/UDP/RTP encapsulation for audio. The H.323 standard provides an infrastructure for audio, video and data communications over packet based networks. This standard is a part of the H.32x protocol family, that includes -besides H.323- standards like H.324 (standard for multimedia transport over SCNs [Sustainable Communities Network]) and H.320 (standard for ISDNs [Integrated Services Digital Network]), among others. The H.323 standard describes four key components for an H.323 system, namely the terminals, gateways, gatekeepers, and Multipoint Control Units (see Fig. 2.5). These components are described in the following subsections.

Terminals: A terminal is a PC or a standalone device running an H.323 protocol and multimedia applications. A terminal supports audio communications and can optionally

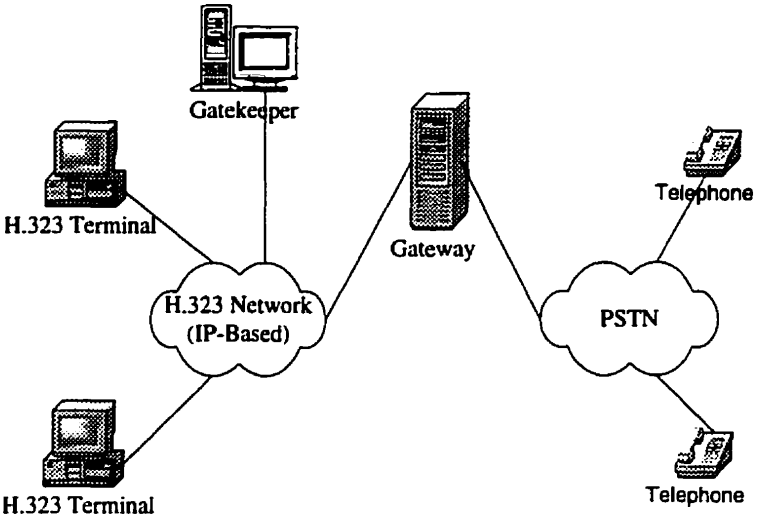


Fig. 2.5 H.323 Network Components.

support video or data communications. The primary goal of H.323 is to interwork with other multimedia terminals. Because the basic service provided by an H.323 terminal is audio communications, an H.323 terminal plays a key role in IP telephony services.

Gateways: A gateway connects two dissimilar networks. An H.323 gateway provides connectivity between an H.323 network and a non-H.323 network. This connectivity of dissimilar networks is achieved by translating protocols for call-setup and release, converting media formats between different networks, and transferring information between different networks connected by the gateway. A gateway is not required, however, for communication between two terminals on an H.323 network. Gateways perform functions like search (conversion of called party phone to IP address), connection, digitization, demodulation, compression/decompression, and demodulation.

Gatekeepers: Gatekeepers can be considered as the brains of the H.323 network. It is the focal point for all calls within the H.323 network. Although they are not mandatory, they perform important services like address translation, admission control, bandwidth management, zone-management, and call-routing services.

Multipoint Control Units (MCU): These provide support for conferences between three or more H.323 terminals. All terminals participating in the conference establish a connection with the MCU. The MCU manages conference resources, negotiates between

terminals for the purpose of determining the audio or video codec to use, and may handle the media stream.

H.323 does not provide any QoS guarantees, but does specify that a reliable transport protocol, such as TCP, can be used for transmitting control information. The voice over IP (VoIP) standard committee is proposing a subset of H.323 for audio over IP [35].

H.225

H.225 [36] is a standard, which covers narrow-band visual telephone services as defined in H.200/ AV.120 series recommendations. It specifically deals with those situations where the transmission path includes one or more packet-based networks, each of which is configured and managed to provide a non-guaranteed QoS. H.225 describes how audio, video, data and control information on a packet-based network can be managed to provide conversational services in H.323 equipment.

H.248

H.248 is same as the Megaco standard published by IETF, and is discussed in the next section.

2.5.2 IETF Standards

The Internet Engineering Task Force (or IETF), along with ITU-T, is playing a key role in VoIP related standardization efforts. The following subsections elaborate upon key areas and their standards (Request for comments(RFCs)).

Media Gateway Control Protocol (MGCP)

The Media Gateway Control Protocol or MGCP [25] implements the interface between a Media Gateway (MG) and a Media Gateway Controller. This interface is implemented as a set of transactions. The transactions are composed of a command and a mandatory response.

MGCP is concerned with several types of gateways, some of which are shown in Fig. 2.6. The trunking gateway operates between a conventional telephone network and a voice over IP network. The residential gateway operates between a traditional telephony end user and

the voice over IP network. The ATM (Asynchronous Transfer Mode) gateway operates the same way as a trunking, except that the interface is between an ATM network and a voice over IP network. The access gateway provides an analog or digital interface of a PBX into an IP over internet network.

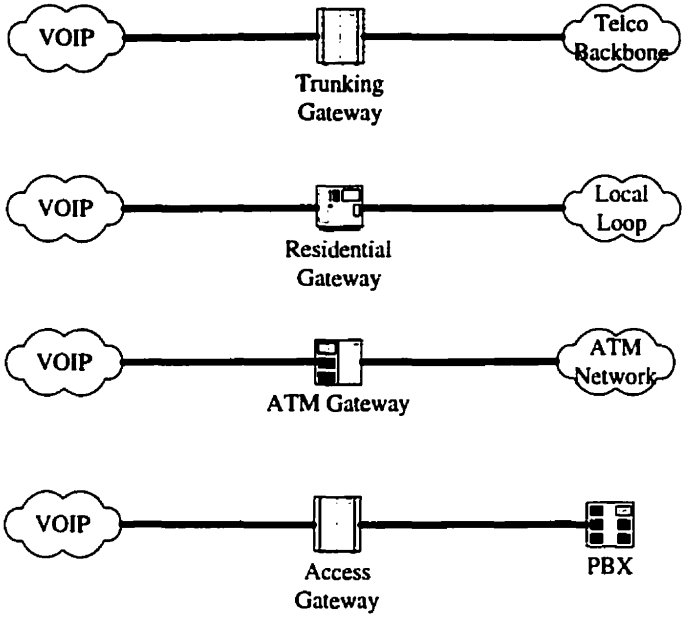


Fig. 2.6 Types of MGCP gateways.

The MGCP assumes the bulk of the intelligence for telephony call control operations and resides in an external element called the 'Call Agent'. This statement does not mean that the gateways are completely unintelligent. Rather, it means that most of the control operations are performed by the 'Call Agent'. In essence, signalling is the responsibility of a call agent. Call agents act as masters to the slave gateways, and the gateways receive commands that define their operations from the call agents. In the Fig. 2.7, one call agent can control three gateways, but the actual configurations depend upon specific installations. The figure also shows that two Call Agents are communicating with each other. The MGCP defines the operations between the call agents and the gateways, but does not define the operations between the gateways.

The MGCP also supports point-to-point or multipoint operations. MGCP protocol is detailed in RFC 2705.

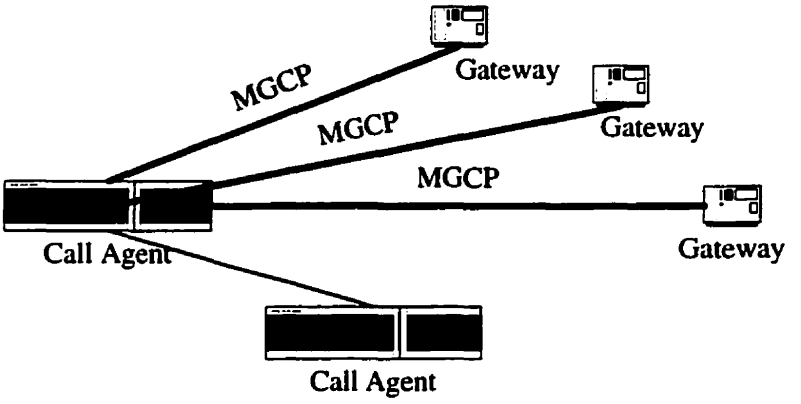


Fig. 2.7 Call agents and gateways.

Megaco

Megaco/H.248 [37] [25] is the media gateway control protocol defined by the IETF and the ITU-T and is used in a distributed switching environment. It is designed as an internal protocol within a distributed system, which appears to the external world as a single VoIP gateway. Internally, the architecture is designed such that the intelligence of call control is outside of the gateways and handled by external agents (see Fig. 2.8). Megaco thus divides the media logic and the signalling logic of a gateway across different functional components. While the Media Gateway (MG) handles the media logic part, the Media Gateway Controllers (MGCs, or Call Agents) control the Media Gateways to establish media paths through the distributed network. An MGC can control multiple MGs. In contrast, one MG can register with multiple MGCs. Communication between these two functional units (MG and MGC) is governed by the Media Gateway Control Protocol (or Megaco). Megaco is thus a master/slave protocol, where the call agents act as command initiators (or masters), and the MGs act as command responders (or slaves).

SIP

The *Session Initiation Protocol (SIP)* [25] is a major support tool for the MGCP (and other signaling systems). It operates with user agents and user agent servers. The main job of the server is to provide name-to-address resolution and user location. For example, when a user makes a call, the user agent sends an SIP message to a server. The user is unaware of

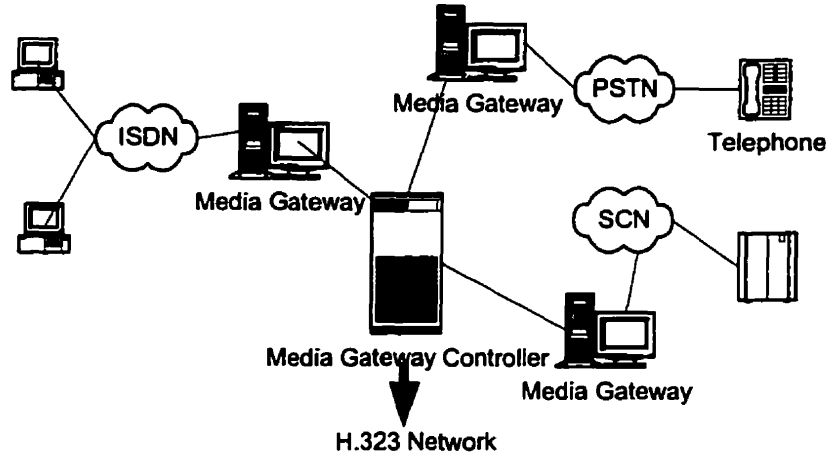


Fig. 2.8 Megaco Network Architecture.

this support operation, but will have given its agent an identifier, such as a phone number. The message is sent to a server by the agent, and at this server, the name may be resolved to an IP address, or the server may be resolved to an IP address, or the server may redirect (proxy) the message to another server.

The SIP allows more than one server to contact the user, and these forked messages are sent to multiple servers. The responses are returned to the agent in such a manner that the agent can make decisions about the best path for a call.

The SIP is a very attractive support tool for IP telephony because:

- It can operate as stateless. A stateless implementation provides good scalability, since the servers do not have to maintain information on the call state once the transaction has been processed.

- It uses much of the formats and syntax of HTTP (Hypertext Transfer Protocol), thus providing a convenient way of operating with ongoing browsers.

- The SIP message (the message body) is opaque; it can be of any syntax. Therefore, it can be described in more than one way. As examples, it may be described with Multipurpose Internet Mail Extension (MIME), or Extensible Markup Language (XML).

- It identifies a user with a URI (Uniform Resource Identifier), thus providing the user the ability to initiate a call by clicking on a web link.

SIP is used for a number of applications such as Internet telephony, call-forwarding, multimedia conferencing, terminal type negotiation, caller/callee authentication, and a host of other multimedia services. SIP is typically transported over the connectionless UDP protocol. UDP is preferred over TCP because of its lower state-management overheads, real-time characteristics, and better performance. The standards for SIP include RFC 2543 (Session initiation protocol), RFC 2327 (Session Description Protocol) and a number of Internet drafts, that are being worked upon. Fig. 2.9 shows network components and sample message flows for an SIP-based network.

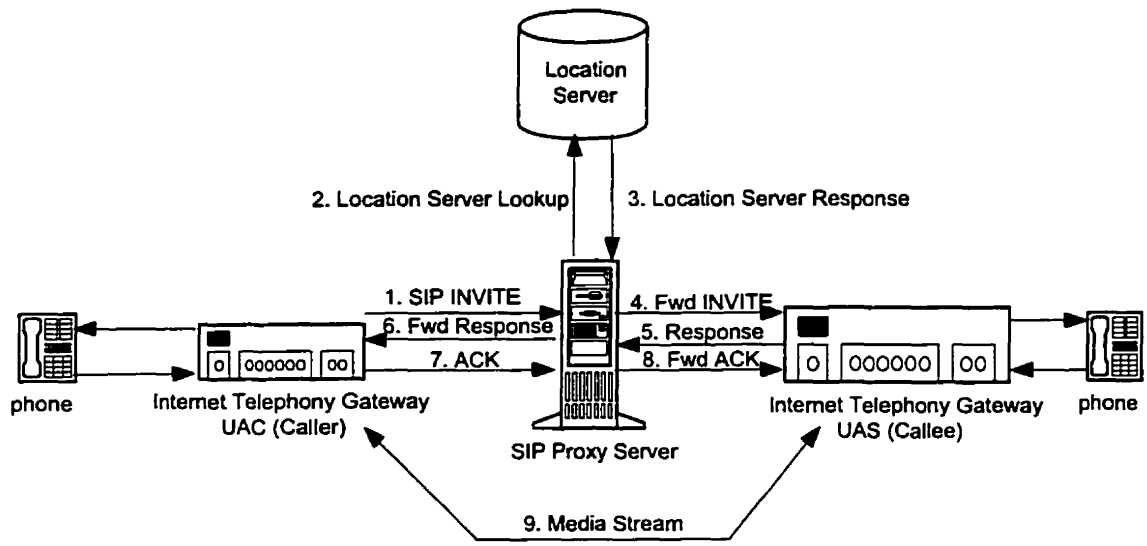


Fig. 2.9 SIP Network Components and Message Flows.

RTSP

Real-Time Streaming Protocol (RTSP) [38] supports the exchange of real-time information between the server and a user. It gives the user the ability to control a media server. A good way to view RTSP is that it provides the user with VCR-type controls, such as fast forward, stop, rewind, record, etc. In addition, a user can direct a media server as to the type of audio (or video) format the media is to use.

RTSP is an excellent tool for controlling the playback rate from a voice-mail server, and it can be used to control the content of a recording.

2.6 Other VoIP Supporting Protocols

2.6.1 IGMP and MBONE

Multicasting (sending from one party to two or more parties) is of keen interest to internet telephony users and designers because it paves the road for conference calls. The Internet has supported multicasting for a number of years with Internet Group Management Protocol (IGMP). This protocol defines the procedures for a user node (host) to join a multicast group. For example, the multicast session may be a quarterly conference of a special group.

One of the attractive features of IGMP is that it does not require a host to know in advance about all the multicasting groups in an internet. Instead, routers are knowledgeable of multicast groups and send advertisements to the hosts about their multicasting groups.

Multicasting backbone (MBONE) is another protocol that has been in operation for number of years. MBONE is the "pioneer" system for Internet audio/video conferences. Originally, MBONE was used to multicast various standards' groups meetings and its use has been expanded for activities such as viewing space shuttle launches, video shows in general, and other activities.

MBONE relies on IP multicasting operations and IGMP to convey information. In addition, the term multicasting backbone does not mean that MBONE is actually a backbone network. MBONE is an application that runs on the Internet backbone [38].

Multicast traffic runs inside the data field of the IP datagram and relies on the conventional IP header for delivery of the traffic through an internet. This concept is called multicast tunnels in the sense that multicast traffic is tunnelled through an internet by riding inside the IP datagram. Fig. 2.10 shows that multicasting traffic is destined to the hosts residing on the networks attached to routers B and C. Traffic emanates from a host attached to router A. The figure shows that the destination multicast address is 224.0.0.99. The figure also shows the unicast IP addresses of the sending host (172.16.1.3) and router B (172.16.1.1), and router C (172.16.1.2).

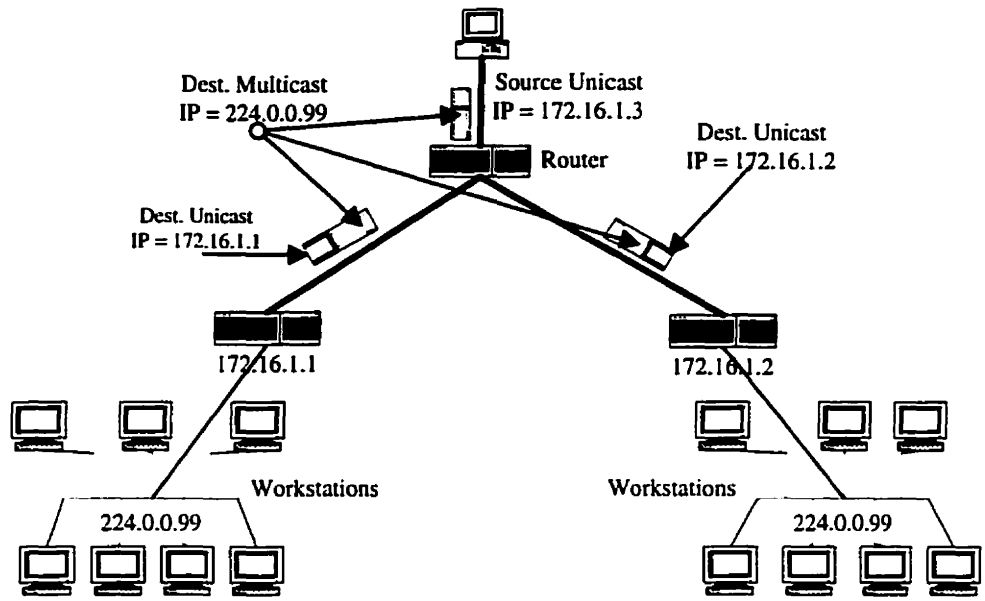


Fig. 2.10 Multicasting Tunnel.

2.6.2 RSVP

Resource Reservation Protocol [38] (RSVP) is used to reserve resources for a session on the internet. This aspect of the Internet, which is quite different to the underlying design intent of the system, was only established to support a best effort service, without regard to predefined requirements for user application.

RSVP is intended to provide guaranteed performance by reserving the necessary resources at each machine that participates in supporting the flow of traffic (such as a video or audio conference). IP does not set up paths for the traffic flow, whereas RSVP is designed to establish these paths, as well as to guarantee the bandwidth on the paths.

RSVP does not provide routing operations, but utilizes IPv4 or IPv6 as the transport mechanism in the same fashion as the Internet Control Message Protocol (ICMP) and the Internet Group Message Protocol (IGMP).

RSVP operates with unicast or multicast procedures and interworks with current and planned multicast protocols. Like IP, it relies on routing tables to determine routes for its messages. It utilizes IGMP to first join a multicast group and then executes procedures to reserve resources for the multicast group.

RSVP enables endpoints to signal the network with the kind of QoS needed for a particular application. The receiver host application must determine the QoS profile which is passed to the RSVP. After the analysis of the request for QoS, RSVP is used to send request messages to all the nodes that participate in the data flow.

2.6.3 RTP

RTP, the Real Time Protocol [7], is a generic mechanism for supporting the integration of voice, video and data. RTP headers provide the sequence number and time-stamp information needed to reassemble a real-time stream from packets. The Real-Time Protocol (RTP) is designed for the support of real-time traffic; that is, traffic that needs to be sent and received in a very short time period, including timing reconstruction, loss detection, security and content identification. Two real-time traffic examples are (a) audio conversations between two people, and (b) playing individual video frames at the receiver as they are received from the transmitter.

RTP is also an encapsulation protocol, in that the real-time traffic runs in the data field of the RTP packets, and the RTP header contains information about the type of traffic that RTP is transporting. The time-stamp field in its header can be used to synchronize the traffic play-out to the receiving application. Fig. 2.11 shows the two major features of RTP in how it supports traffic from senders to receivers. In the first figure, the RTP system is acting as a translator and in the second figure, an RTP server is performing a mixer operation. RTP is standardized in [7].

2.6.4 RTCP

After a reservation has been established through the use of RSVP, the traffic is then sent between machines with RTP. Next, the Real Time Control Protocol (RTCP) [25] [7] comes into the picture by providing the procedure for the machines to keep each other informed about, (a) the quality of services they think they are providing (if they are service providers), and/or (b) the quality of services they are receiving (if they are service clients).

RTCP provides support for real-time conferencing for large groups within an Internet, including source identification and support for gateways (like audio/video bridges), and multicast-to-unicast translators. It is possible to use the RTP without RTCP.

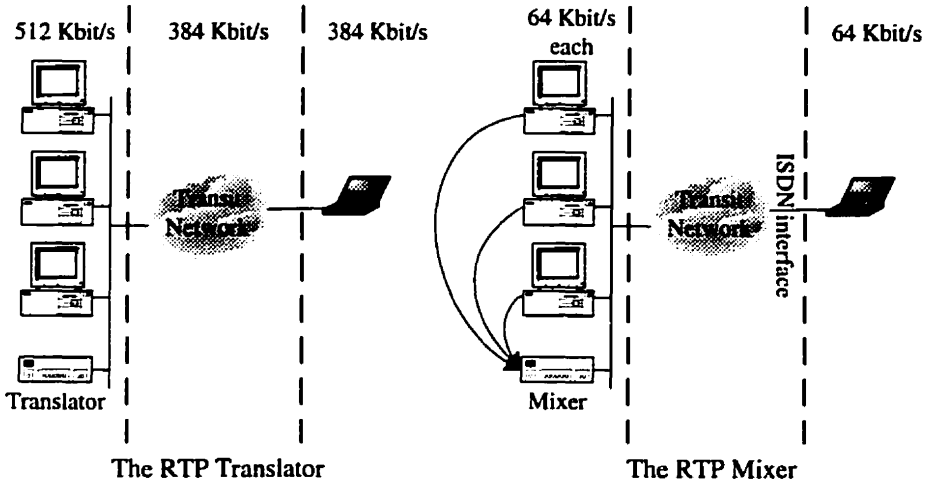


Fig. 2.11 An RTP Translator and Mixer.

2.7 Implementing VoIP in Systems

The deployment of a VoIP infrastructure for public use involves much more than simply adding compression functions to an IP network. Anyone must be able to call anyone else, regardless of location and the form of network attachment (telephone, wireless phone, PC, or other device). Fig. 2.12 illustrates one scenario for how telephony and facsimile can be implemented using an IP network. This design would also apply if other types of packet networks (such as ATM or frame relay) were being used.

Fig. 2.13 is a refinement of Fig. 2.12 that includes the placement of the VoIP gateway and the system level support functions that are integral to a high quality VoIP system. The VoIP gateway is shown here as a separate component, but it could also be integrated into the voice switch (a PBX or Central Office (CO) Switch) or into an IP Switch. Some of the functions that are required for a VoIP system include:

- Fault Management.
- Accounting/Billing.
- Configuration.
- Addressing/Directories.

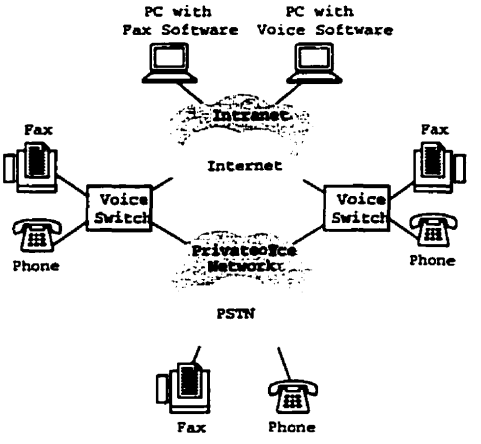


Fig. 2.12 VOIP Infrastructure.

- Authentication/Encryption.

Implementations of full-scale VoIP systems must provide all the abilities that are usually taken for granted in open systems (including the PSTN). These include:

- ***Interoperability:*** In a public networking environment different products will need to interwork if any-to-any communications is to be possible. Using common software that has been tested for conformance to all applicable standards (such as for compression), can significantly reduce the cost of product development. The interconnection of VoIP to the PSTN also involves meeting the specific standards for telephone network access.

- ***Reliability:*** The VoIP network, whether by design or through management, should be fault tolerant with only a very small likelihood of complete failure. In particular, the gateway between the Telephone and VoIP systems needs to be highly reliable.

- ***Availability:*** Sufficient capacity must be available in the VoIP system and its gateways to minimize the likelihood of call blocking and mid-call disconnects. This will be especially important when the network is shared with data traffic that may cause congestion. Mechanisms for admission control should be available for both the voice and data traffic, with prioritization policies set.

- ***Scalability:*** There is potential for extremely high growth rates in VoIP systems,

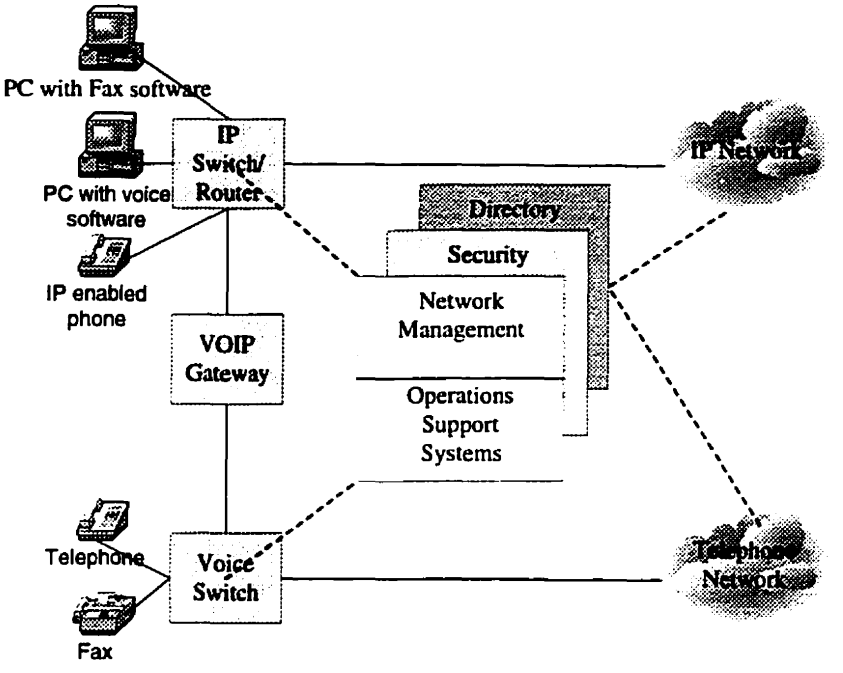


Fig. 2.13 A Combined PSTN/VOIP System.

especially if they prove to be the equal of PSTN at a much lower cost. VoIP systems must be flexible enough to grow to very large user populations, allow a mix of public and private services and adapt to local regulations. The need for large numbers of addressable points may force the use of improved Internet protocols such as IPv6.

- ***Accessibility:*** Telephone systems assume that any telephone can call any other telephone and allow the conferencing of multiple telephones across wide areas. This will be driven by functions that map between telephone numbers and other types of packet network addresses, specifically IP addresses. There must, of course, exist gateways that allow every device to be reachable.

- ***Viability:*** Many are claiming significant economic advantages to the implementation of VoIP based on flat rate prices for Internet service. There is no regulatory prohibition against the interconnection of telephone systems with IP systems.

Chapter 3

Linear Prediction of Speech

This chapter provides the background on the Linear Predictive analysis of speech signals. This research is motivated to improve the packet loss concealment strategy using the residual signal. The residual signal is obtained by filtering the original signal through an LP analysis filter. This residual signal is converted back to the original signal using an inverse LP filter (synthesis filter). The LP filter coefficients are computed from a windowed version of the original signal using the autocorrelation method. The inverse LP filter coefficients are computed by interpolating the LP coefficients of past and future packets. The spectral representation of the LP coefficients -the LSF's- have been used for the interpolation of the LP coefficients.

This chapter describes the analysis methods used to determine LP coefficients and different representations of these coefficients, which will help to understand the detailed approach to packet loss concealment provided in Chapter 5.

3.1 Acoustical Model of Speech Production

Speech is a sound wave created by vibration that is propagated in the air. A full analysis of the vocal tract should consider three-dimensional wave propagation, the variation of the vocal tract shape with time, losses due to heat conduction and viscous friction at the vocal tract walls, softness of the vocal tract walls, radiation of sound at the lips, nasal coupling and excitation of sound. While a detailed model that considers all of the above is not yet available, some models provide a good approximation in practice, as well as a good understanding of the physics involved.

The vocal cords constrict the path from the lungs to the vocal tract. As lung pressure is increased, air flows out of the lungs and through the opening between the vocal cords (glottis). At one point the vocal cords are together, thereby blocking the airflow, which builds up pressure behind them. Eventually the pressure reaches a level sufficient to force the vocal cords to open and thus allow air to flow through the glottis (this is illustrated in Fig. 3.1). Then, the pressure in the glottis falls and, if the tension in the vocal cords is properly adjusted, the reduced pressure allows the cords to come together, and the cycle is repeated. This condition of sustained oscillation occurs for voiced sounds. The closed-phase of the oscillation takes place when the glottis is closed and the volume velocity (density of air in the vocal cord) is zero. The open-phase is characterized by a non-zero volume velocity, in which the lungs and the vocal tract are coupled.

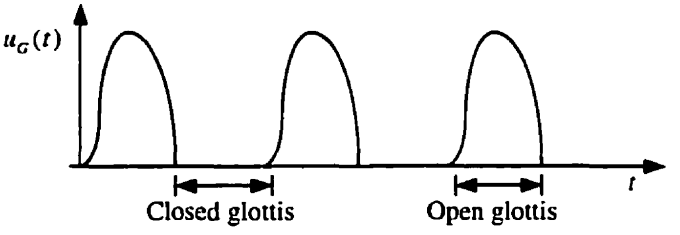


Fig. 3.1 Glottal excitation: volume velocity is zero during the closed phase, during which the vocal cords are closed.

3.2 Human Speech Properties

Speech can generally modelled as a baseband signal, limited to a bandwidth of 7–8 kHz [5]. The spectral characteristics of the speech wave are time-varying, since the physical system which produces speech (vocal tract) changes over time. As a result, speech can be divided into sound segments that possess similar acoustic properties over short periods of time, i.e. the vowel 'o' in the word 'shop'. Nonetheless, speech is not quite a string of discrete well-formed sounds, but rather a series of steady-state sounds with intermediate transitions. The preceding and/or succeeding sound in a string can affect whether a target is reached completely, how long it is held and other finer details of the sound. This interplay is generally called *coarticulation.*

If we look at the spectrum of a steady-state segment, we can observe the different frequency components. Frequency components depend on the actual shape, size and position of several cavities that are formed in the vocal tract. Each vocal tract shape is characterized by a set of resonant frequencies, and peaks due to resonances are known as formants. Three-to-five formants can be found in the 4 kHz frequency band. Formants usually appear as peaks in the power spectrum.

Fig. 3.2 shows the time domain representation of an unvoiced sound (low energy) and a voiced sound (high energy). Fig. 3.3 shows the spectral representation of voiced sound and an unvoiced sound. The formants are very prominent in the spectral representation of the voiced sound whereas the spectrum of the unvoiced sound is flatter.

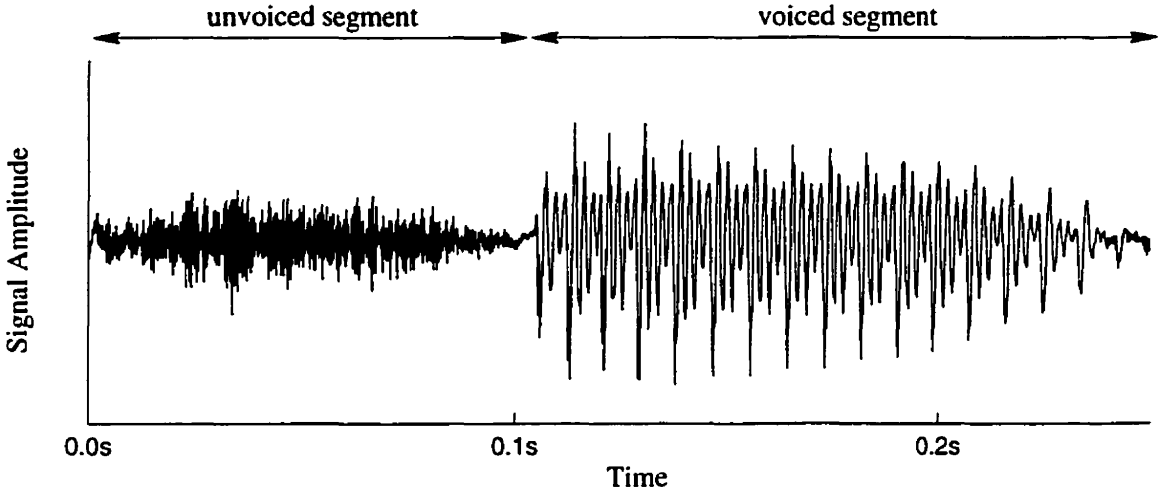
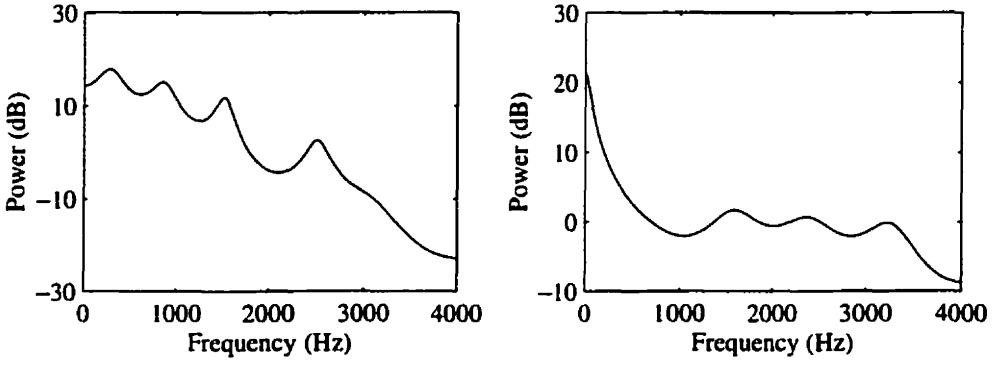


Fig. 3.2 Time domain representation of a voiced to unvoiced speech segment.

A special category of segments that speech is usually divided into are the so-called phonemes. More specifically, phonemes are the basic theoretical units for describing how speech conveys linguistic meaning. Each phoneme can be considered to be a code that consists of a unique set of articulatory gestures. From an acoustical point of view, the phoneme represents a class of sounds that convey the same meaning. Voiced segments are described by the parameters called fundamental period T_0, which is actually the time between successive vocal fold openings, while the rate of vibration is called the fundamental frequency $F_0 = 1/T_0$. The term pitch basically implies the fundamental frequency F_0 that



(a) Power spectrum of a voiced speech signal

(b) Power spectrum of an unvoiced speech signal

Fig. 3.3 Short-time power spectra of voiced (a) and unvoiced (b) sound.

can typically take the values from 50 to 200 Hz and changes with time.

3.3 Linear Prediction Model

Vocal tract analysis has shown, from the practical point of view, that it can be well modelled by an *autoregressive (AR) model*, also called *all pole filter* to describe the vocal tract. In an AR model the nasal cavity is neglected. Hence, the vocal tract is reduced to the pharynx- and mouth cavity. An AR model can be described by:

$$s(n) = b_0 u(n) - \sum_{k=1}^{m} c_k s(n-k). \tag{3.1}$$

Here, $s(n)$ corresponds to speech output and $u(n)$ to the sound excitation signal generated by the vocal cord. It is a recursive filter, but only takes the present sound excitation $u(n)$ into account instead of the last m values of $u(n)$; this is sufficient for speech modelling. Nonetheless, even Eq. (3.1) is quite difficult to realize, as the real model parameters c_k are unknown. Therefore, one attempt to get an estimate for $s(n)$ by a linear equation:

$$\hat{s}(n) = \sum_{k=1}^{p} a_k s(n-k). \tag{3.2}$$

The Eq. (3.2) conforms with a non-recursive filter (FIR). The set of parameters a_k can be optimized by several methods so as to match the real speech sequence $s(n)$ as closely as possible. The residual sequence is defined as the difference occurring between the original signal $s(n)$ and the estimation value (see Fig. 3.4). The LP residual signal is also called the prediction error signal. With the assumption that the order $p = m$ and the factor $b_0 = 1$, the prediction error $r(n)$ is given as:

$$r(n) = s(n) - \hat{s}(n) \tag{3.3}$$

$$= s(n) - \sum_{k=1}^{p} a_k s(n-k). \tag{3.4}$$

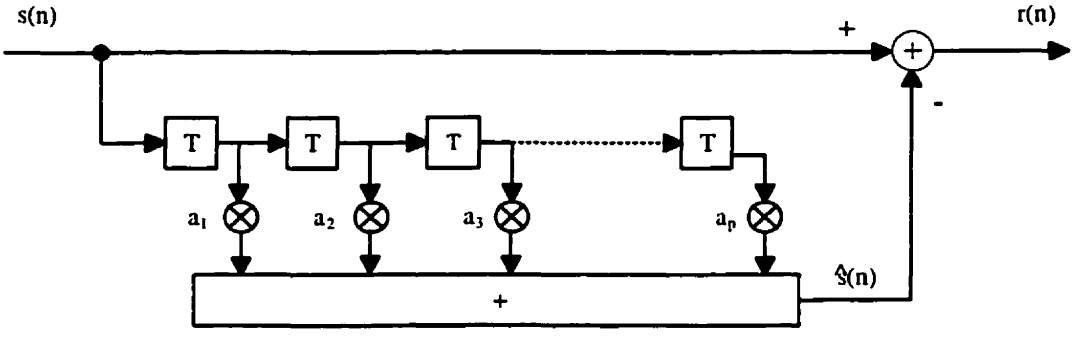


Fig. 3.4 Linear Prediction with non-recursive filter.

Taking the z-transform on both sides of the given Eq. (3.4), gives

$$R(z) = A(z)S(z), \tag{3.5}$$

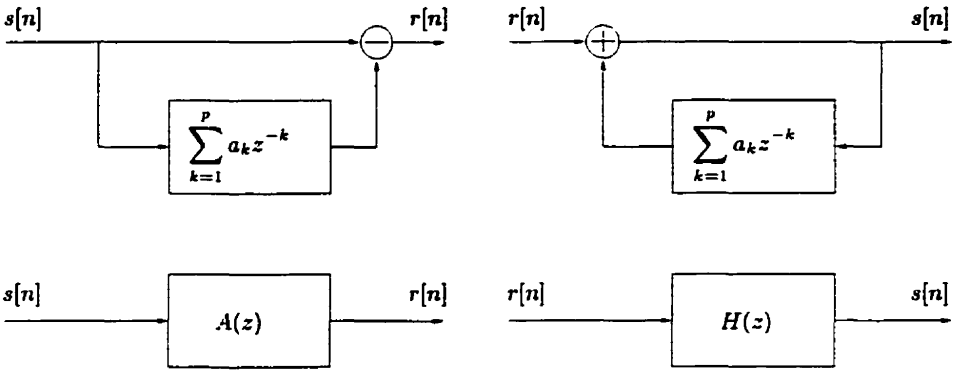
where $R(z)$ is the z-transform of the LP residual signal, and

$$A(z) = 1 - \sum_{k=1}^{p} a_k z^{-k}. \tag{3.6}$$

The filter $A(z)$ is known as the *LP analysis filter*. The all-pole *LP synthesis filter* $H(z)$, given as,

$$H(z) = \frac{1}{A(z)}, \tag{3.7}$$

models the short-term spectral envelope of the speech signal. A diagram of the analysis and the synthesis stage is shown in Fig. 3.5.



(a) Formant Prediction. (b) Formant Synthesis stage.

Fig. 3.5 Block diagrams of format (a) analysis and (b) synthesis stages.

The order of the analysis and the synthesis model is selected, based on a compromise between spectral accuracy, computation time, memory and transmission bit rate. Generally a pair of poles is allowed for each formant present in the speech spectrum, plus an additional 2 to 4 poles to approximate possible zeros. For 8 kHz sampled speech, the order typically ranges from 8 to 16.

3.4 Estimation of Linear Prediction Coefficients

A speech signal is not stationary and its statistics are not explicitly known. Thus, the predictor must therefore be adapted to changing signal characteristics. Typically, during time intervals up to 20 ms, speech signals are considered to be stationary. Windowing the sampled signal is the first step in linear prediction parameter estimation. There are two widely used methods for estimating the LP coefficients:

- Autocorrelation.
- Covariance.

Both methods use prediction coefficients $\{a_k\}$ in such a way that the energy of the residual signal or error signal is minimized. As we have used the autocorrelation method in this research to compute LP coefficients, we will only discuss this method.

3.4.1 Windowing

We know that phonemes have an average duration of 80 ms. So, while analyzing speech signals it is assumed that the properties of the signal do not change within the short interval of time (15 ms, used in this research); this allows for short term analysis of a signal. The signal is divided into successive segments and analysis is done on these segments. For this purpose, a signal segment $s(n)$ is multiplied by a fixed length of window $w(n)$, called an *analysis window*, to extract the parameters. The right shape of the window is very important, because it allows different samples to be weighted differently. The simplest window is a rectangular window:

$$w(n) = \begin{cases} 1 & \text{for } 0 \leq n \leq N_w - 1 \\ 0 & \text{otherwise.} \end{cases} \tag{3.8}$$

It has an abrupt discontinuity at the edge in the time domain. As a result there are large side lobes and undesirable ringing effects [39] in the frequency domain representation of the rectangular window. To avoid large oscillations, a window without discontinuities in the time domain should be used. This corresponds to low side lobes of the windows in the frequency domain. The use of the Hamming window is very common in speech analysis. It is actually a raised cosine function:

$$w(n) = \begin{cases} 0.54 - 0.46\cos(\frac{2\pi n}{N_w - 1}) & \text{for } 0 \leq n \leq N_w - 1 \\ 0 & \text{otherwise.} \end{cases} \tag{3.9}$$

There are also other types of tapered windows, such as Hanning, Blackman, Kaiser and Barlett windows. A window can also be hybrid, which means that the two halves of the window are of different sizes. As an example, G.729 coder uses an asymmetric window [30].

3.4.2 Autocorrelation Method

In the autocorrelation method, the speech signal $s(n)$ is first multiplied by an analysis window $w(n)$ of finite length to obtain a windowed speech segment $s_w(n)$.

$$s_w(n) = w(n)s(n). \tag{3.10}$$

After multiplying the speech signal with the analysis window, the autocorrelation of the windowed speech signal segment is computed. The autocorrelation function of the windowed signal $s_w(n)$ is

$$R(i) = \sum_{n=i}^{N-1} s_w(n)s_w(n-i), \qquad 0 \leq i \leq p. \tag{3.11}$$

The autocorrelation function is an even function where $R(i) = R(-i)$.

To solve for the LP filter coefficients, the energy of the prediction residual within the finite interval $0 \leq n \leq N$ defined by the analysis window $w(n)$ must be minimized:

$$E = \sum_{n=-\infty}^{\infty} r^2(n) = \sum_{n=-\infty}^{\infty} \left(s_w(n) - \sum_{k=1}^{p} a_k s_w(n-k) \right)^2. \tag{3.12}$$

By setting the partial derivatives of the energy with respect to the filter coefficients to be zero,

$$\frac{\partial E}{\partial a_k} = 0 \qquad 1 \leq k \leq p, \tag{3.13}$$

we obtain p linear equations in p unknown coefficients a_k:

$$\sum_{k=1}^{p} a_k \sum_{n=-\infty}^{\infty} s_w(n-i)s_w(n-k) = \sum_{n=-\infty}^{\infty} s_w(n-i)s_w(n), \qquad 1 \leq i \leq p. \tag{3.14}$$

By substituting the values from Eq. (3.11) in Eq. (3.14), we get

$$\sum_{k=1}^{p} R(|i-k|)a_k = R(i), \qquad 1 \leq i \leq p. \tag{3.15}$$

The set of linear equations can be represented in the following matrix form:

$$\begin{bmatrix} R(0) & R(1) & \cdots & R(p-1) \\ R(1) & R(0) & \cdots & R(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ R(p-1) & R(p-2) & \cdots & R(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} R(1) \\ R(2) \\ \vdots \\ R(p) \end{bmatrix}. \tag{3.16}$$

The above equation can be expressed as

$$\boldsymbol{Ra} = \boldsymbol{r}, \tag{3.17}$$

where $\boldsymbol{R}$ is a $p \times p$ Toeplitz matrix which contains values of the autocorrelation sequence for $s(n)$, $\boldsymbol{a}$ is a $p \times 1$ vector of prediction coefficients, and $\boldsymbol{r}$ is a $p \times 1$ vector of autocorrelation values. Since $\boldsymbol{R}$ is a Toeplitz matrix, $A(z)$ is minimum phase [40] (see Eq. (3.6)) and can be solved by the Levinson-Durbin [41] algorithm. At the synthesis filter $H(z) = 1/A(z)$, the zeros of $A(z)$ become the poles of $H(z)$. Thus, the minimum phase of $A(z)$ guarantees the stability of $H(z)$.

3.4.3 Bandwidth Expansion and Lag Window

LP analysis cannot accurately estimate the spectral envelope for high-pitch voiced sounds, as it may generate synthesis filters with artificially sharp spectral peaks. To avoid this problem, bandwidth expansion [42] [43] may be employed. This has the effect of expanding the bandwidth of the formant peaks in the frequency response.

In the process of bandwidth expansion, the roots of the all-pole filter are scaled by an expansion factor γ, which has the following form:

$$H'(z) = \frac{1}{A'(z)} = \frac{1}{A(\gamma z)}, \tag{3.18}$$

where the expanded prediction coefficients are

$$a'_k = a_k \gamma^k \qquad 1 < k < p. \tag{3.19}$$

The *bandwidth expansion factor* γ for f_b Hz is computed as

$$\gamma = e^{\frac{-f_b \pi}{f_s}}. \tag{3.20}$$

For instance, $\gamma = 0.996$ approximately yields a 10 Hz bandwidth expansion in the analysis of speech sampled at 8 kHz. For speech analysis, bandwidth expansion of 10 to 25 Hz are often performed. The problem can be solved in another way. In this procedure the autocorrelations are multiplied by a *lag window* (usually a Gaussian shape). This is equivalent to convolving the power spectrum with a Gaussian shape, and this widens the peaks of the spectrum.

3.5 Representation of LP Spectral Parameters

In the proposed PLC (Packet Loss Concealment) algorithm, interpolated LP coefficients of the past and future frame are used in the inverse LP filter. Straightforward interpolation of LP coefficients is not done because small changes in the coefficient causes a large change in the power spectrum, and may result in an unstable LP synthesis filter. Therefore, other parametric representations such as: line spectral frequencies (LSF), autocorrelation coefficients (AC), reflection coefficients (RC), log area ratios (LAR), cepstral coefficients(CC), etc., of the LP coefficients are used for this purpose. All of these parametric representations have effectively one-to-one correspondence with the LP coefficients, and preserve the information of LP coefficients.

For the purpose of this research, autocorrelation functions have been used to detect the voiced and unvoiced segments and LSFs for interpolation. In this section, we will discuss the AC and LSF representations of the LP coefficients and LSF interpolation.

3.5.1 Autocorrelation Function

The autocorrelation function $R(n)$ is an alternate way of representing direct form predictor coefficients. To compute the filter coefficients using this method, first we need to calculate the sample correlation function. No extra effort is needed to obtain those parameters. The most important and interesting property about the autocorrelation function is that the sample correlation function of two consecutive frames of a signal is almost equal to the average of the sample correlation function of the two frames. The model that is achieved

by averaging the autocorrelation function is close to that obtained by considering two consecutive frames as one frame [44]. This feature makes it attractive for interpolation of the LP coefficients. Autocorrelation functions can be normalized using the frame energy $R(0)$, and be called normalized autocorrelation functions. Even after normalizing the values of autocorrelation function, it is still used as usual. The autocorrelations which are not normalized are called energy weighted autocorrelation coefficients (EAC).

3.5.2 Line Spectral Frequency

Line Spectral Frequencies (LSF), also known as Line Spectral Pairs (LSP's) provide an equivalent representation of the predictor coefficients that is very popular in speech processing. This involves mapping the p zeros of $A(z)$ onto the unit circle through two z-transforms $P(z)$ and $Q(z)$ of $(p+1)$th order:

$$P(z) = A(z) + z^{-(p+1)}A(z^{-1}), \tag{3.21}$$

$$Q(z) = A(z) - z^{-(p+1)}A(z^{-1}), \tag{3.22}$$

it directly follows that:

$$A(z) = \frac{1}{2}[P(z) + Q(z)]. \tag{3.23}$$

The roots of the polynomials $P(z)$ and $Q(z)$ are called the LSF's. According to [45] and [46] the polynomials $P(z)$ and $Q(z)$ have the following properties:

- All zeros of $P(z)$ and $Q(z)$ lie on the unit circle.
- Zeros of $P(z)$ and $Q(z)$ are interlaced with each other; i.e.,the LSF's are in ascending order.
- The minimum phase property of $A(z)$ can be preserved, if the first two properties are intact after quantization or interpolation and $\mid a_p \mid < 1$.

The roots of $P(z)$ and $Q(z)$ occur in complex-conjugate pairs and hence there are 'p' LSF's lying between 0 and π. The process produces two fixed zeros at $\omega = 0$ and $\omega = 1$ which can be ignored.

There are several ways to calculate the LSF's; one that is used by Soong and Juang [46] applies a discrete cosine transformation [47] to the coefficients of the polynomials $P(z)$ and $Q(z)$.

Kabal and Ramachandran [48] use an expansion on the mth order *Chebyshev* polynomial in x:

$$T_m(x) = \cos(m\omega), \tag{3.24}$$

where $x = \cos\omega$ maps the upper semi-circle in the z-plane to the real-valued interval $[-1, 1]$. The roots of the expanded polynomials are determined iteratively by looking at the sign changes in the range $[-1, 1]$. The LSF's correspond to the polynomial roots using the transformation $\omega = \cos^{-1}(x)$.

3.6 Interpolation of Linear Prediction Coefficients using LSF's

If the interpolation is implemented directly in the LP coefficients domain, the filter using these interpolated LP coefficients is not guaranteed to be stable. Therefore, the linear predictive coefficients are converted into a different parametric representation, which have one-to-one correspondence with the linear predictive coefficients. Interpolation in that corresponding domain and converting the coefficients to the LP domain keeps the filter stable. With the proper choice of interpolation technique, the undesired transients due to large change in the LP based models at adjacent frames can be avoided in the reconstructed or synthesized speech signal.

The LSF's are interlaced with each other for a given LP analysis order. Kim and Lee [49] called this property the intra-model interlacing theorem. By preserving the intra-model interlacing theorem of the interpolated LSF's it is possible to have a stable interpolated LSF synthesis filter. In [50], Kabal and Islam showed that interpolation using LSF has the best performance compared to than any other representations of LP coefficients.

Chapter 4

Time-Scale Modification of Speech

In this research, the idea of time-scale modification of a speech signal is used not to change the speaking rate of a signal, but to reconstruct the signal segment which is lost or delayed. The main idea is to generate a signal which has a pitch period close to the original signal using the pitch period information of the past and future frames. The WSOLA time-scale modification (TSM) technique is capable of generating an output signal with the same pitch period from the signal provided to the algorithm. This technique also minimizes discontinuities at the boundaries between good packets and reconstructed packets. It is possible to use the WSOLA time-scale modification technique on the residual signal in the same way it is used on the original signal. This chapter gives the details of overlap-add (OLA), synchronize overlap-add (SOLA) and waveform similarity overlap-add (WSOLA) time-scale modification techniques.

4.1 Definition of Time-Scale Modification

Time-scale modification of speech refers to processing performed on speech signals that changes the perceived rate of articulation without affecting the pitch or intelligibility of the speech. Such modification can be categorized into two classes: time-scale compression (or speed-up) which increases the rate of articulation; and time-scale expansion (or slow-down) which decreases the rate of articulation.

Traditional uses of time-scale modification allow for either faster listening of messages recorded on answering machines, voice mail systems, and other information services, or synchronizing speech with the typing speed from dictation. Alternatively, the goal of slow-

down (time-scale expansion) in most cases is to decrease the rate of articulation to aid in comprehension or dictation of rapidly spoken speech segments with important information, such as an address or phone number.

For voice-over-IP, time-scale modification has been used for adaptive playout scheduling by scaling individual voice packets which modifies the rate of the playout voice signal in packet voice communication [51]. It has also been used for packet loss concealment or delay concealment for internet voice applications [52] [53]. Changing the playback speed of MPEG-compressed audio -without decompressing the audio file first- can be achieved by using time-scale modification [54].

The problem with time-scaling a speech signal $x_a(t)$ (Fig. 4.1(a)) of original duration Δt lies with the corresponding frequency distortion. The duality between time scaling and frequency scaling (depicted in Fig. 4.1) becomes clear by considering the signal $y_a(t)$ (Fig. 4.1(b)) that corresponds to an original signal $x_a(t)$ played at a speed α (in Fig. 4.1, α = 0.5) times higher than the recording speed. Thus, an original time span Δt is played in $\frac{\Delta t}{\alpha}$ and $y_a(t) = x_a(\alpha t)$. From the definition of the Fourier transformation for analog signals, uniform scaling in one domain corresponds to reverse scaling in the transformed domain:

$$y_a(t) = x_a(\alpha t) \leftrightarrow Y_a(\Omega) = \frac{1}{|\alpha|} X_a\left(\frac{\Omega}{\alpha}\right). \tag{4.1}$$

The intended time scaling clearly does not correspond to mathematical time scaling. Rather it requires a scaling of the *perceived* timing attributes, such as speaking rate, without affecting perceived frequency attributes, such as pitch. Because of the mathematical duality between time domain and frequency domain representations, one can consider two equivalent formulations for this problem:

- Modify the time domain representation of signal $x_a(t)$ without altering its perceived frequency attributes.
- Modify the frequency domain representation of signal $y_a(t)$, which is $Y_a(\Omega)$, without altering its perceived time structure (such as pitch period).

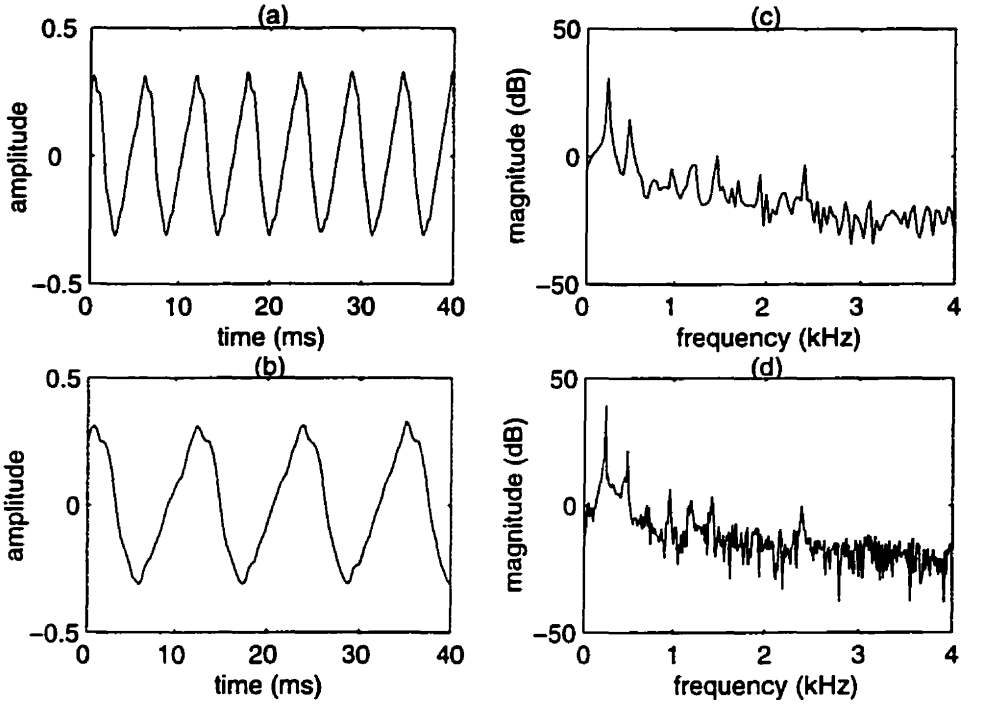


Fig. 4.1 Illustration of the duality between the time domain and frequency domain. The upper row shows a 40 ms voiced speech segment and its spectrum; the second row illustrates that when the signal is played at half speed (by changing the sampling rate) it is stretched twofold in the time domain and compressed in the frequency domain.

4.2 The Time Scaling Function

Formally, time-scale modifications are specified by defining $n \rightarrow n' = \tau(n)$ between the original time-scale and the modified time-scale. This mapping defines the so-called *time-scaling* or *time-warping* function. It specifies that the sounds which occur at time n in the original signal should occur at time n' in the time-scaled signal. Sometimes, a time varying time-modification rate $\beta(t)$, where $\beta(t) > 0$, is specified, from which the time-scaling function can be derived as

$$n \rightarrow n' = \tau(n) = \frac{1}{T} \int_0^{nT} \beta(u) du, \tag{4.2}$$

where T is the sampling period. At the time instants where $\beta(t) > 1$, the time scaling corresponds to slowing down the original, while at time instants where $0 < \beta(t) < 1$, the time scaling corresponds to speeding-up the original.

4.3 Time-Scale Expansion and Application

During slow down, the length of the modified signal must be increased relative to the original signal, resulting in a segment of longer duration. Such modification would slow down the perceived rate of articulation. Ideally, the expansion employed should insert additional pitch periods distributed evenly throughout the entire segment. However, this proves to be difficult, as the local pitch period varies across phonemes and may be difficult to gauge during non-periodic portions of the speech signal such as fricatives.

In this research, we will use time-scale expansion to develop a new packet loss concealment algorithm.

4.4 Existing Time-Scale Modification Techniques

Several algorithms have been developed to achieve time-scale modifications based on the inherent structure of the speech signal. Time-domain techniques rely on the periodic nature of speech, while analysis/synthesis techniques exploit redundancies in the signal to reduce the speech waveform to a limited set of time varying parameters.

4.4.1 Time-Domain Algorithms - TDHS (Time-Domain Harmonic Scaling)

Time-domain techniques operate by inserting or deleting segments of speech signal, which can result in discontinuities in the transition between inserted or deleted segments. Several attempts have been made to minimize the effects of inter-segment transitions in the final signal by improving the splicing technique or windowing adjoining segments [55] [56]. These techniques improve quality at the expense of increasing complexity.

The TDHS algorithm [55] employs multiple correlations of signal segments to determine local pitch periods along intervals of the input signal. A triangular windowing function is aligned with the pitch periods and the resulting segments are added such that pitch periods are inserted or deleted to create a time-scale modified signal. The algorithm requires exact pitch determination to operate successfully, thus pitch variations in the input signal must

be tracked accurately. In general, the pitch period is constant only over very short intervals and may drastically vary between phonemes. Consequently, the length of the triangular window as well as the length of inserted or deleted segments must vary. Numerous methods for determining pitch in the time-domain have been discussed in [30] [16] [29].

TDHS provides good quality in the class of low complexity time-domain algorithms. It uses pitch-synchronous windowing intervals and variable length windowing functions to reduce inter-segment discontinuities in the output signal. There are a couple of alternatives to this method, such as Synchronized Overlap-Add (SOLA), which was originally proposed by Roucos and Wilgus [56], and Waveform Similarity Overlap-Add (WSOLA), proposed by Verhelst and Roelands [17]. These techniques have low complexity and operate in the time-domain, but do not rely on pitch tracking. As these methods use fixed window lengths and fixed windowing intervals, they have advantages for real-time implementation.

In this research, WSOLA has been used to perform the time scale modification of the speech signal. The algorithm operates by finding the closest match of each segment from the waveform previously stored and doing overlap-add with the matched signal of about 50%.

4.5 Short-Time Fourier Transform and Overlap-Add Synthesis

4.5.1 The Short-Time Fourier Transform

The Fourier transform, defined as

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n}, \tag{4.3}$$

is a frequently used frequency domain representation. If we consider speech as a signal with slowly evolving characteristics (i.e., as a quasi-stationary signal), a short-time analysis strategy can be applied together with Fourier transformations to obtain the Short-Time Fourier Transform (STFT) as the desired time-frequency representation [57]. The short-time Fourier transform of a signal $x(n)$ is defined by segmenting the signal using the windowing function $w(n)$

$$x_w(n, m) = w(n)x(n + m), \tag{4.4}$$

and taking its Fourier transform

$$X(\omega, m) = \sum_{n=-\infty}^{\infty} x(n+m)w(n)e^{-j\omega n}. \tag{4.5}$$

Conceptual disadvantages of this approach is that the analysis precision is limited by the windowing operation and non-stationarity; a practical advantage is that short-time analysis works with consecutive, possibly overlapping, signal segments and is easily amenable to on-line processing.

4.5.2 The Overlap-Add Synthesis Method

By modifying $X(\omega, n)$ to achieve time scaling the result may no longer represent a STFT in that a signal which has the modified transform $\hat{Y}(\omega, n)$ as its STFT may not exist. Nevertheless, $\hat{Y}(\omega, n)$ would contain the information which best characterizes the signal modification, such that a special synthesis formula is required which leads to the correct result if $\hat{Y}(\omega, n)$ is a STFT and a reasonable result. One such synthesis method uses overlap-addition (OLA). As introduced by Griffin and Lim [58], this method constructs $y(n)$ such that its STFT $Y(\omega, n)$ is maximally close to $\hat{Y}(\omega, n)$ in the least squares sense, i.e., such that the total squared error

$$E = \sum_k \frac{1}{2\pi} \int_{-\pi}^{+\pi} \left| \hat{Y}(\omega, k) - Y(\omega, k) \right|^2 d\omega, \tag{4.6}$$

is minimized over all signals $y(n)$ (the sum is over all time instants k for which $\hat{Y}(\omega, k)$ is defined). From Parseval's theorem, Eq. (4.6) can be written as

$$E = \sum_k \sum_{m=-\infty}^{+\infty} (\hat{y}_w(m, k) - y(m+k)w(m))^2, \tag{4.7}$$

where $\hat{y}_w(m, k)$ is the inverse Fourier transform of $\hat{Y}(\omega, k)$. The signal $y(n)$ which minimizes 'E' is obtained by solving

$$\frac{\partial E}{\partial y(n)} = -2 \sum_k (\hat{y}_w(n-k, k) - y(n)w(n-k))w(n-k) = 0. \tag{4.8}$$

Thus,

$$y(n) = \frac{\sum_k w(n-k)\hat{y}_w(n-k,k)}{\sum_k w^2(n-k)}, \tag{4.9}$$

where

$$\hat{y}_w(n-k,k) = \frac{1}{2\pi}\int_{-\pi}^{+\pi} \hat{Y}(\omega,k)e^{j\omega(n-k)}d\omega, \tag{4.10}$$

is the inverse Fourier transform of $\hat{Y}(\omega,k)$ delayed by k samples. The OLA synthesis formula reconstructs the original signal if $X(\omega,m)$ is a valid STFT, or constructs a signal whose STFT is maximally close to $X(\omega,m)$ in the least squares sense. Furthermore, note that the denominator in Eq. (4.9) is required only to compensate for a possible non-uniform weighting of samples in the windowing procedure. The synthesis operation can be simplified if the windowing function and the synthesis time instants k can be chosen such that

$$\sum_k w^2(n-k) = 1. \tag{4.11}$$

A common choice in speech processing that satisfies this simplifying condition is a Hanning window with 50% overlap between successive segments; some other possibilities are listed in [58].

4.6 Time-Scaling Techniques

4.6.1 Overlap-Add Time Scaling

The OLA synthesis gives a close realization of the time-scale modification in the time domain. By adopting a short-time analysis strategy for constructing $X(\omega,m)$ and by using the OLA criteria for synthesizing a signal $y(n)$ from the modified representation $\hat{Y}(\omega,m) = M_{xy}[X(\omega,m)]$, we will always obtain modification algorithms that can be operated in the time domain if the modification operator $M_{xy}[.]$ works on the time index m only (where m is the analysis time instant $t_a(u)$ and operator $M_{xy}[.]$ is associated with the time warping function and equal to $\tau^{-1}(m)$):

$$\hat{Y}(\omega,m) = X(\omega, M_{xy}[m]) \qquad \text{modification,} \tag{4.12}$$

$$\hat{y}_w(n,m) = x_w(n, M_{xy}[m]) \qquad \text{inverse FT,} \tag{4.13}$$

$$y(n) = \frac{\sum_m w(n-m)x_w(n-m, M_{xy}[m])}{\sum_m w^2(n-m)} \qquad \text{OLA synthesis.} \tag{4.14}$$

It is clear from the Eq. (4.14) that modification is obtained by excising segments $x_w(n, M_{xy}[m])$ from the input signal using the window and repositioning them along the time axis before constructing the output signal by the weighted overlap-addition of the segments. However, Fig. 4.2(b) shows that the periodicity of the time-scale modified signal is changed from the original sinal depicted in fig. 4.2(a) if we apply the above formula to see the time warping $\tau(m)$ of a signal. So, poor results are generally obtained when using $\hat{Y}(\omega, m) = X(\omega, \tau^{-1}(m))$.

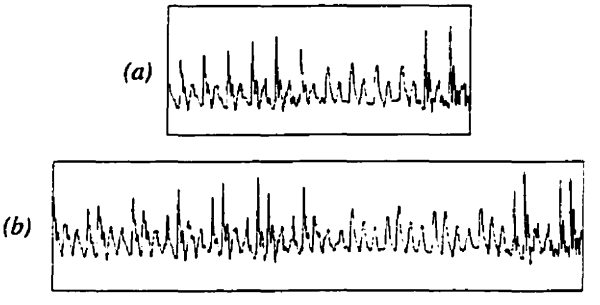


Fig. 4.2 OLA synthesis from the time-scaled STFT does not succeed in replicating the quasi-periodic structure of the signal (a) in its output (b) (from [17]).

4.6.2 The Synchronized OverLap-Add

The Synchronized Overlap-Add (SOLA) algorithm was developed by Roucos and Wilgus [56]. They sought to accomplish time-scale modification by providing the algorithm with an initial guess closer to the desired signal. The SOLA algorithm modifies the time-scale of a signal in two steps, analysis and synthesis. The analysis step consists of windowing the input signal for every S_a (Shift analysis) samples as depicted in Fig. 4.3. The synthesis step consists of overlap-adding the windows (L_w is a window length, which is fixed and a multiple

of pitch period) from the analysis step for every S_s (Shift synthesis) samples (rate-modified unshifted signal in Fig. 4.3). Each new window is aligned to maximize the correlation with the sum of previous windows before being added. This reduces discontinuities arising from the different interframe intervals used during analysis and synthesis. The resulting time-scale modified signal is free of clicks, and pops. Fig. 4.3 shows an example of the time-scale expansion of a signal using the SOLA algorithm.

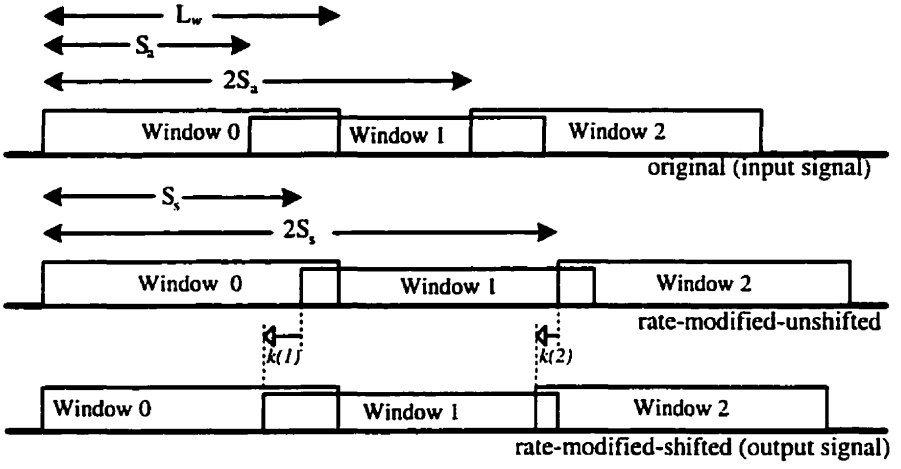


Fig. 4.3 Overview of time-scale modification(expansion) using SOLA.

In the "Synchronized Overlap-Add" algorithm, windows are added synchronously with the local period. The time-scale modified signal, $y(n)$, obtained using the "Synchronized Overlap-Add" of windowed segments, $x_w(n) = w(n)x(n)$ (where $x(n)$ is the input signal and $w(n)$ is the window function), is given by:

1. Initializing the signals $y_w(n)$ and $r(n)$:

$$\left.\begin{aligned} y_w(n) &= x_w(n) \\ r(n) &= w(n) \end{aligned}\right\}, \qquad \text{for } n = 0 \ldots L_w - 1. \tag{4.15}$$

2. Updating $y_w(n)$ and $r(n)$ by each new frame of the input signal, $x_w(n)$, as follows:

$$y_w(mS_s - k(m) + j) = \begin{cases} y_w(mS_s - k(m) + j) + x_w(mS_a + j) & \text{for } 0 \le j \le L_m - 1 \\ x_w(mS_a + j) & \text{for } L_m \le j \le L_w - 1, \end{cases} \tag{4.16}$$

where L_m is the number of overlapping points between the new window $x_w(mS_a + j)$ and the existing sequence $y_w(mS_s - k(m) + j)$ for the current frame m.

$$r(mS_s - k(m) + j) = \begin{cases} r(mS_s - k(m) + j) + w(mS_a + j) & \text{for } 0 \le j \le L_m - 1 \\ w(mS_a + j) & \text{for } L_m \le j \le L_w - 1, \end{cases} \tag{4.17}$$

$$k(m) = \max R_{xy}^m(k), \tag{4.18}$$

$$R_{xy}^m(k) = \frac{\sum_{j=0}^{L_m-1} y_w(mS_s - k + j)x_w(mS_a + j)}{\sqrt{\left[\sum_{j=0}^{L_m-1} y_w^2(mS_s - k + j)\right]\left[\sum_{j=0}^{L_m-1} x_w^2(mS_a + j)\right]}}. \tag{4.19}$$

3. Normalizing $y_w(n)$ by the buffer of appropriately shifted windowing functions $r(n)$ to obtain the final output $y(n)$:

$$y(j) = \frac{y_w(j)}{r(j)}, \qquad \text{for all } j. \tag{4.20}$$

As outlined in the above equations, $k(m) > 0$ corresponds to a shift backwards along the time-axis of the mth frame that maximizes the *normalized cross-correlation* $R_{xy}^m(k)$ between the mth window and rate-modified shifted signal composed of windows $0, \ldots, (m-1)$. L_w is the number of data points in each window frame $x_w(mS_a + j)$.

Maximizing the *cross-correlation* insures the current window is added and averaged with the most similar region of the reconstructed signal as it exists at that point. The shifting operation insures that the largest amplitude periodicity of the signal will be preserved in the rate-modified signal. This signal is to be called the *rate-modified shifted signal* to distinguish it from the *rate-modified unshifted signal* which is obtained simply by overlap-adding (see Fig. 4.3).

It is known that the straightforward OLA synthesis from the time-scaled and downsam-

pled STFT $\hat{Y}(\omega, kS) = X(\omega, \tau^{-1}(kS))$ results in a signal

$$y_1(n) = \frac{\sum_k w^2(n - kS)x(n - kS + \tau^{-1}(kS))}{\sum_k w^2(n - kS)}, \tag{4.21}$$

that is heavily distorted, as illustrated in Fig 4.2. In Eq. (4.21), 'S' is a downsampling factor introduced to reduce the amount of information that needs to be processed.

To avoid pitch period discontinuities or phase jumps at waveform-segment joins, [56] proposed to realign each input segment to the already formed portion of the output signal before performing the OLA operation. Thus, synchronized OLA algorithm produces the time-scale modified signal

$$y(n) = \frac{\sum_k v(n - kS + \Delta_k)x(n - kS + \tau^{-1}(kS) + \Delta_k)}{\sum_k v(n - kS + \Delta_k)}, \tag{4.22}$$

in a left-to-right fashion with a windowing function $v(n)$, and with a shift factor $\Delta_k \in [-\Delta_{\max} \ldots \Delta_{\max}]$ that is chosen such as to maximize the cross-correlation coefficient between the current segment $v(n-kS+\Delta_k)x(n+\tau^{-1}(kS)-kS+\Delta_k)$ and the already formed portion of the output signal

$$y(n; k-1) = \frac{\sum_{l=-\infty}^{k-1} v(n - lS + \Delta_l)x(n + \tau^{-1}(lS) - lS + \Delta_l)}{\sum_{l=-\infty}^{k-1} v(n - lS + \Delta_l)}. \tag{4.23}$$

SOLA is computationally efficient since it requires no iterations and can be operated in the time domain. The time domain operation implies that the corresponding STFT modification affects the time axis only. In case of SOLA, we have

$$\hat{Y}(\omega, kS - \Delta_k) = X(\omega, \tau^{-1}(kS)). \tag{4.24}$$

The shift parameters Δ_k thus implies a tolerance on the time warp function: in order to ensure a synchronized overlap-addition of segments, the desired time warp function $\tau(n)$ will not be realized exactly. A deviation on the order of a pitch period is allowed.

4.6.3 Waveform Similarity Overlap-Add

This part of the section describes the mathematical formulation and algorithm of the Waveform Similarity Overlap-Add (WSOLA) technique. It has been found that for this algorithm the time-scale modified version of speech quality is very high and robust against background noises, including competing voices [17]. We have used the WSOLA technique to conceal the lost packets in VoIP (as described in the next chapter).

Efficient Time-Scaling of the Speech Signal

The problem with the time-scale modification of the speech signal $x(n)$ lies in realizing the specific time-warp function $\tau(n)$ in such a way as to affect the apparent speaking rate only, preserving other perceived aspects such as timbre, voice-quality, and pitch. The OLA synthesis is close to realizing time-scale modifications using time domain operations only.

In order to construct an efficient high-quality time-scaling algorithm based on OLA, a tolerance Δ_k on the precise time-warp function that will be realized, is needed to allow a synchronized overlap-addition of original input segments to be performed in the time domain. This tolerance can be used, as in SOLA, to realize segment synchronization during synthesis $\hat{Y}(\omega, kS-\Delta_k) = X(\omega, \tau^{-1}(kS))$. However, as the Δ_k's are not known before hand, the denominator in the OLA Eq. (4.22) could not be made constant in that case. Further reduction of computational costs would be possible by using fixed synthesis time instants $S_k = kS$ and a window $v(n)$ such that $\sum_k (v(n) - kS) = 1$. Proper synchronization must then be ensured during the segmentation

$$\hat{Y}(\omega, kS) = X\big(\omega, \tau^{-1}(kS) + \Delta_k\big). \tag{4.25}$$

Thus it would seem that the most efficient realization of OLA time scaling would use the simplified synthesis equation

$$y(n) = \sum_k v\big(n - kS\big)x\big(n + \tau^{-1}(kS) - kS + \Delta_k\big). \tag{4.26}$$

where Δ_k are chosen such as to ensure sufficient signal continuity at waveform segment boundaries according to some criterion. WSOLA [17] proposes a synchronization strategy inspired by a time scaling criterion.

4.6.4 A WSOLA criterion for time scaling

We consider that a time-scaled version of an original signal should be perceived to consist of the same acoustic events as the original signal, but with these events being produced according to a modified timing stricture. In WSOLA, we assume that this can be achieved by constructing a synthetic waveform $y(n)$ that maintains maximal local similarity to the original waveform $x(m)$ in all neighbourhoods of related sample indices $m = \tau^{-1}(n)$. Using the symbol '$\rightleftharpoons$' to denote "the maximal similarity" and using the window $w(n)$ to select such neighbourhoods, we require

$$y(n+m)w(n) \rightleftharpoons x\big(n+\tau^{-1}(m)+\Delta_m\big)w(n) \qquad \text{for all } m, \tag{4.27}$$

or equivalently

$$\hat{Y}(\omega,m) \rightleftharpoons X\big(\omega,\tau^{-1}(m)+\Delta_m\big) \qquad \text{for all } m. \tag{4.28}$$

Comparing Eq. (4.27)and Eq. (4.28) with Eq. (4.25) we find an alternative interpretation for the timing tolerance parameters Δ_k as we see that the waveform similarity criterion and the synchronization problem are closely related. As illustrated in Fig. 4.4, the Δ_m in Eq. (4.27), Eq. (4.28) was introduced because, in order to obtain a meaningful formulation of the waveform similarity criterion, two signals need to considered as identical if they only differ by a small time offset[1]. Referring to Fig. 4.4, we need to express that the waveshapes of segments from the quasi-periodic signal in the middle of the figure are similar at all time instants. Such similarities go unnoticed in the upper pair of segments because they are located at different positions in their respective pitch cycles. By introducing a tolerance Δ_m on the time instance around which segment waveforms are to be compared, the quasi-stationarity of the signal can easily be detected from the lower pair of segments that was synchronized by letting $\Delta_m = \Delta$. We can thus conclude that using the requirement of waveform similarity between the input and output signal as a criterion for time scaling, readily implies that a synchronization of input and output segments will have to take place.

As Eq. (4.25) can be viewed as a downsampled version of Eq. (4.28), and it has been

[1]It can be noted that waveform similarity was used to approximate some similarity. Because two signals that differ only by some time offset sound the same, we also need to declare their waveforms to be similar.

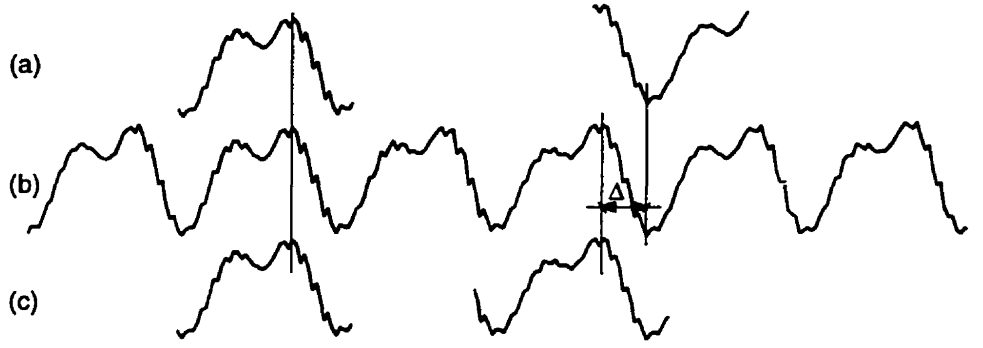


Fig. 4.4 Alternate interpretation of timing tolerance parameters Δ. Signal (b) is the search segment from original signal, and (a) and (c) are the segments to find maximum cross-correlation with.

proposed to select the parameters Δ_k such that the resulting time scaled signal

$$y(n) = \sum_k v\big(n - kS\big) \times x\big(n + \tau^{-1}(kS) - kS + \Delta_k\big), \tag{4.29}$$

maintains maximal similarity to the original waveform x_m in corresponding neighbourhoods of related sample indices $m = \tau^{-1}(n)$.

4.6.5 The WSOLA algorithm

Based on the idea of WSOLA, a variety of practical implementations can be constructed. A common version of WSOLA uses a 20 ms hanning window with 50% overlap ($S = f_s/100$, with f_s the sampling frequency in Hz) to construct the signal of Eq. (4.29) in a left-to-right manner as illustrated in Fig. 4.5.

Assume the segment labelled (A) in Fig. 4.5 was the previous segment that was excised from the input signal and overlap-added to the output at time instant $S_{k-1} = (k-1)S$. This means, synthesis (output) segment (a) is equal to input segment(A). At the next synthesis position $S_k = kS$ we need to choose a synthesis segment (b) that is to be excised from the input around a time instant $\tau^{-1}(S_k) + \Delta_k$, where $\Delta_k \in [-\Delta_{max} \ldots \Delta_{max}]$. The values in Δ_k are to be chosen such that the resulting portion of $y(n)$, $n = S_{(k-1)} \ldots S_k$ will be similar to a corresponding portion of the input. Segment (C) overlap-adds with segment (A) to reconstruct a portion of the original signal $x(n)$. The segment (C) would

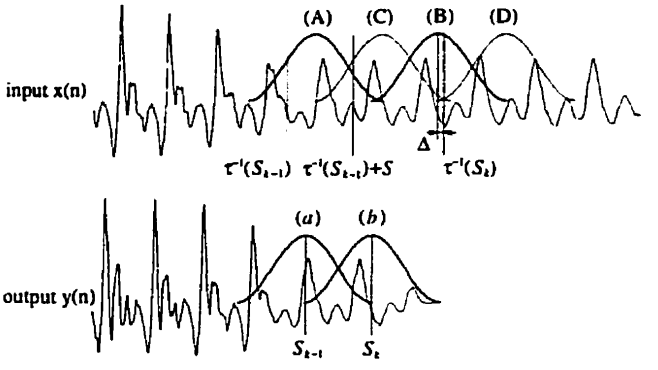


Fig. 4.5 Illustration of WSOLA time scaling (from [17]).

also overlap-add with segment (a) to reconstruct that same portion of the original in the output signal $y(n)$ (remembering segment (a) is equal to segment (A)). If segment (C) does not lie in the timing tolerance region $[-\Delta_{max} \ldots \Delta_{max}]$ around $\tau^{-1}(S_k)$, we cannot accept this segment (segment (C)) as a legitimate candidate for synthesis segment (b). We can use this segment (segment (C)) as a template to select segment (b) such that it resembles segment (C) as closely as possible, which is located within the prescribed tolerance interval around $\tau^{-1}(S_k)$ in the input signal. The position of this best segment (B) can be found by maximizing a similarity measure between the sample sequence underlying segment (C) and the input signal. After overlap-addition of synthesis segment (b) (which is equal to input segment (B)) to the output, we can proceed to the next synthesis time instant using segment (D) as our next template [59].

Fig. 4.6 illustrates in more detail how the position of a best segment 'm' is determined by finding the value $\delta = \Delta_m$ that lies within a tolerance region $[-\Delta_{max} \ldots \Delta_{max}]$ around $\tau^{-1}(mS)$ and maximizes the chosen similarity measure $c(m, \delta)$ with respect to the signal portion that would form a natural continuation for the previously chosen segment $m-1$. If N represents the window length, some examples of similarity measures that can be applied successfully are:

- cross-correlation coefficient

$$c_c(m, \delta) = \sum_{n=0}^{N-1} x\big(n + \tau^{-1}((m-1)S) + \Delta_{m-1} + S\big) \times x\big(n + \tau^{-1}(mS) + \delta\big), \quad (4.30)$$

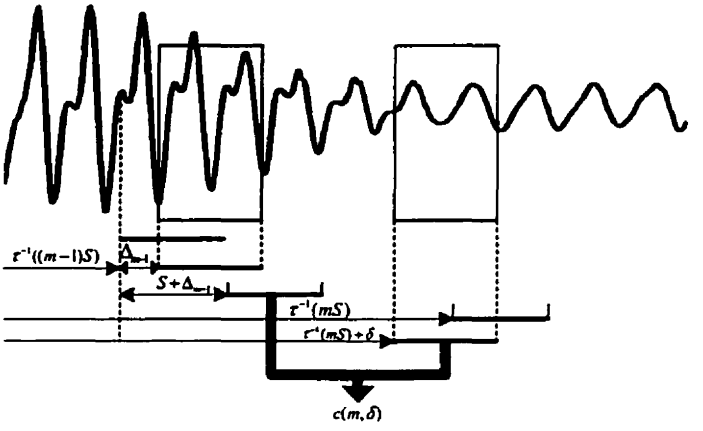


Fig. 4.6 Illustration of similarity-based signal segmentation in WSOLA.

- normalized cross-correlation coefficient

$$c_n(m,\delta) = \frac{c_c(m,\delta)}{\left(\sum_{n=0}^{N-1} x^2(n+\tau^{-1}(mS)+\delta)\right)^{1/2}}, \tag{4.31}$$

- cross-AMDF coefficient

$$c_A(m,\delta) = \sum_{n=0}^{N-1} \left|x\big(n+\tau^{-1}((m-1)S)+\Delta_{m-1}+S\big) - x\big(n+\tau^{-1}(mS)+\delta\big)\right|. \tag{4.32}$$

WSOLA proposed the criterion of waveform similarity as a substitute for the time-scaling criterion which required that in all corresponding time instance, the original and the time-scale signal should sound similar. Clearly, waveform similarity can only be a valid substitute for sound similarity if the similarity can be made sufficiently close. For time-scaling speech signals, which are largely made up from long stretches of quasi-periodic waveforms and noise like waveshapes, a strategy like WSOLA is indeed able to produce close waveform similarity. In that case the precise similarity measure selected does not matter too much in that any reasonable distance measure (Like cross-correlation in Eq. (4.30), cross-AMDF in Eq. (4.32), etc.) will do.

As illustrated in Fig. 4.7 and Fig. 4.8, the original and the WSOLA time-scaled wave-

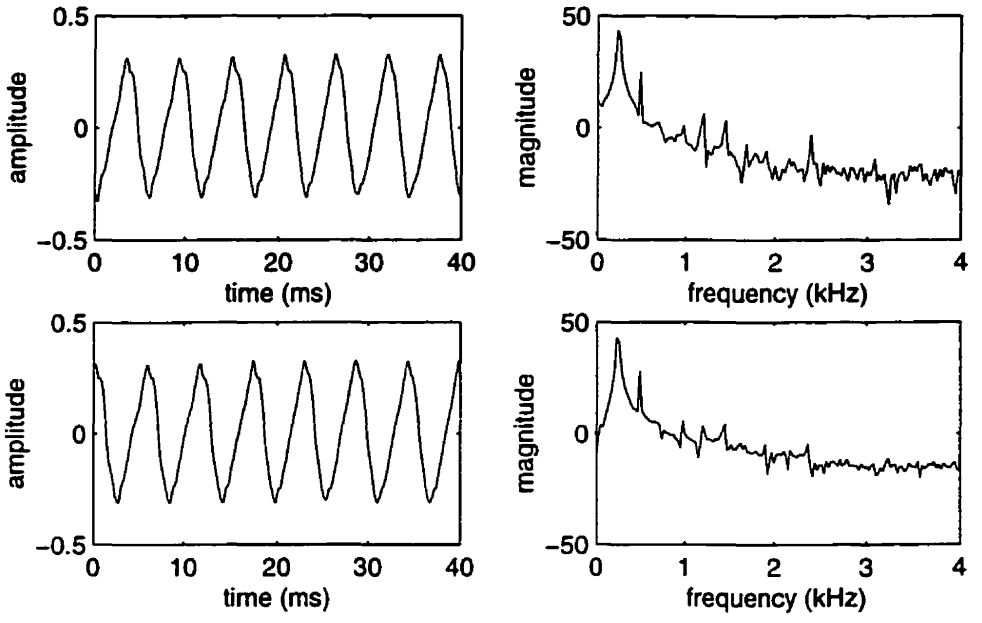


Fig. 4.7 Frequency domain effects of WSOLA time scaling. The upper row shows a 40 ms voiced speech frame and its spectrum; the second row illustrates that when this signal is played at half speed using WSOLA, no frequency scaling occurs.

forms do indeed show a very close similarity. Also, many variations of the basic technique can be constructed by varying the window function, the similarity measure, the portion of the original $x(n)$ that is to serve as a reference for natural signal continuity across OLA segment boundaries, etc. As many such variants all provide a similar high quality [17], this design flexibility can be used to further optimize the algorithms implementation for a given target system. Table 4.1 summarizes a qualitative comparison between the SOLA and WSOLA time-scale modification method.

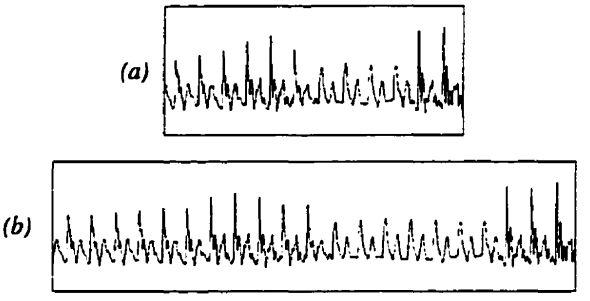


Fig. 4.8 Illustration of an original speech fragment (*a*) and the corresponding WSOLA output waveform when slowed down to 60% speed (*b*) (from [32]).

Table 4.1 Comparison between the SOLA and WSOLA for on-line time-scale modification of the speech signal [17].

	SOLA	WSOLA
Synchronizing method	output similarity	input similarity
Effective window length	fixed (> 4×pitch)	fixed
Algorithm and comp -utational efficiency	high	very high
Robustness	high	high
Speech quality	high	high
Pitch modification	no	no

Chapter 5

Concealment of Missing Packets

5.1 Introduction

In an audio transmission network system the input signal is encoded and packetized at a transmitter, sent over a network, and decoded at the receiver. Packet Loss Concealment (PLC) algorithms -also known as frame erasure concealment algorithms- are used to conceal lost packets due to transmission errors. The objective of PLC is to generate a synthetic speech signal to cover missing data (erasures) in a received bit stream. Many of the standard CELP-based speech coders, such as ITU-T Recommendations G.723.1 [29], G.728 [60] and G.729 [30], have in-built PLC algorithms. The methods for packet loss concealment described in this chapter are applicable to packetized speech transmission systems that use ITU-T Recommendation G.711 [28] as the coding mechanism.

Unlike CELP-based coders, G.711 has no model of speech production. Hence, the concealment algorithm for G.711 is entirely receiver-based. G.711 has the advantage that the signal returns to the original signal at the first sample in the first good packet after an erasure. With CELP-based coders, the decoder's state variables take time to recover after an erasure. Thus, PLC in G.711 has the ability to recover rapidly after an erasure is over.

The WSOLA time-scale modification technique is deployed to conceal these lost packets for a voice over IP system. As described earlier, WSOLA is a time-scale modification technique that can be applied to the original signal. In this research we have used WSOLA to time-scale the residual signal. The reason behind using the residual signal is that the LP analysis decouples the vocal tract and the pitch period information of an original signal. Thus, the pitch period information is very prominent in the residual signal. Time-scale

modification of the residual signal using WSOLA strongly preserves the pitch period information of the original signal. After passing the residual signal through a synthesis filter to add vocal tract information, the output signal will have the exact pitch period of the signal that has been provided in the time-scale modification. Also, the synthesis signal smooths the discontinuity that affects the perceived quality of the reconstructed signal. The proposed algorithm considers the future packet information for the reconstruction of lost packets.

This chapter starts with the description of the existing PLC technique described in the standard T1.521a (Annex B). Explanation of the new PLC algorithm will be followed by a comparison between these two algorithms using informal subjective testing.

5.2 ANSI T1.521a-2000 (Annex B) Standard for Packet Loss Concealment

The PLC technique described in this standard uses the linear predictive model of speech production to estimate the vocal tract and excitation information from the previously received packets to reconstruct the signal contained in the missing packet; it works with packet sizes of 5–30 ms. The default sampling rate is 8 kHz, but this algorithm can support other sampling rates.

5.2.1 Description of the Algorithm

The algorithm estimates the spectral characteristics of a missing speech segment, and then synthesizes a high-quality approximation of the missing segment using the LPC speech production model [16].

The LP-based PLC algorithm is implemented entirely at the receiver side of a transmission channel. Speech is generated by passing an excitation signal through an inverse LP filter. In this model, a speech signal is composed of two components

- LP analysis parameters that model the vocal tract information.
- A residual signal that contains the excitation information.

The basic operation of the algorithm is to estimate these two components for the missing speech segment, based on the LP analysis of the previously received speech frames.

State variables used in this algorithm

- *Speech buffer:* The most recent 30 ms of speech is kept in the speech buffer. The first 25 ms of this buffer contains samples that have already been played-out and the last 5 ms contains samples that have been received but not yet played.
- *Overlap buffer:* This buffer contains a 5 ms extension of the generated signal. It is used for overlap-and-add with the first good packet that comes after the lost segment.
- *LP coefficients:* The LP coefficients that are calculated at the first lost packet using samples from the speech buffer are re-used if the consecutive packets are also lost.
- *Excitation buffer:* The excitation signal for a lost packet is saved in the excitation buffer. It is used if the following packet is also lost.
- *Pitch period:* The pitch period estimated for the first packet of a lost speech segment is stored and used with consecutive lost packets.
- *Previous loss indicator:* A binary flag that shows whether the previous packet was lost or not.
- *Scale:* Current value of the envelope that scales the signal to be played.

First Lost Packet

The majority of the computations in this algorithm are done at the first packet of a lost segment. Fig. 5.1 shows the block diagram of these computations. Whenever the current packet is missing a binary indicator, which indicates the status of the previous packet, (previous loss indicator) is set to '1' and executes the blocks shown in Fig. 5.1.

LP Analysis :

The last 15 ms of the speech buffer is windowed by an asymmetric Hamming window. An LP order of 20 is used. If the signal energy is too low, LP coefficients are not computed and an all zero vector of LP coefficients is passed to the LP filter and to the inverse LP filter. Otherwise, bandwidth expansion is applied by windowing the autocorrelation function with an exponential lag window. The computed LP coefficients are also saved and used at consecutive lost packets.

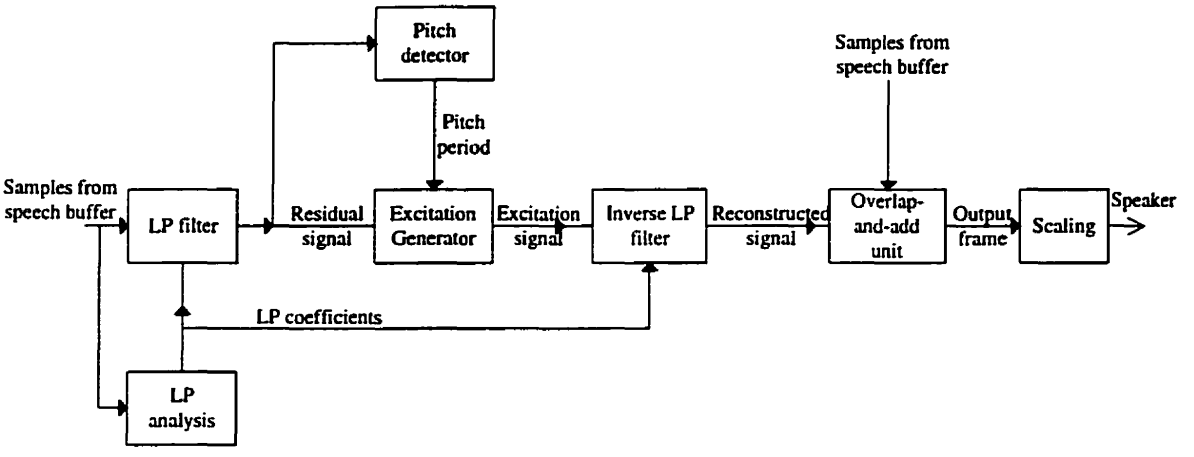


Fig. 5.1 Block Diagram for the LP-Based PLC algorithm.

LP Filter :

The entire speech buffer is filtered by the LP filter to extract the vocal tract information. Then the residual signal is passed through the pitch detector and the excitation generator.

Pitch Detector :

The pitch period of the previous speech frames is estimated by searching for the peak locations in the normalized autocorrelation function of the residual signal. Pitch periods ranging from 2.5 ms to 15 ms are searched at a resolution of 0.125 ms. Samples of the pitch period are passed to the excitation generator. These are also stored and are used in the case of a consecutive packet loss.

Excitation Generator :

The residual signal and the computed pitch period of the previous speech frames are used to generate an excitation signal for the lost packet and two 5 ms segments just before and after the lost packet. As illustrated in Fig. 5.2, the 5 ms segment, just before the last P (P is the pitch period) samples, is copied from the residual signal to the beginning of the new excitation signal. Then, the last P samples of the residual signal are appended to the excitation signal as many times as necessary to fill the remaining portion. The size of the

entire excitation signal is of 120 samples (15 ms). It is also stored for future use, in case the next packet is lost.

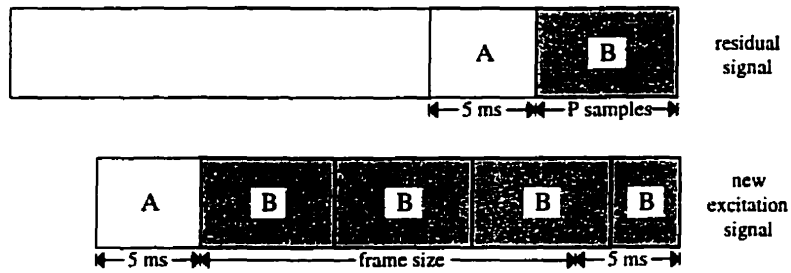


Fig. 5.2 Generating the new excitation signal from the residual signal.

Inverse LP Filter :

The excitation signal is filtered by the inverse LP filter to add the vocal tract information. The output signal is 120 samples long (for the packet size of 80 samples). The first 80 samples are used to replace the lost segment and last 40 samples are used for overlap-and-add with the next segment.

Overlap-and-Add :

The overlap-and-add unit uses the reconstructed signal and the samples from the speech buffer to generate the output frame. The two 5 ms segments at the beginning and the end of the reconstructed signal (output signal from the inverse LP filter block) are used for overlap-and-add operations. The 5 ms (40 samples) segment at the beginning of the reconstructed signal (size of 120 samples) and the 5 ms (40 samples) segment at the end of the speech buffer (size of 240 samples) are weighted by a triangular window and summed. This resulting (windowed and summed) signal replaces the 5 ms (40 samples) segment at the end of the speech buffer.

The 5 ms (40 samples) segment at the end of the reconstructed signal is copied into the overlap buffer. It is used for overlap-and-add with the next packet if the next packet is not lost.

Scaling (lost packets) :

The output frame is scaled down before it is played-out to the speaker, by multiplying each sample by the current value of the scale. The scale is set to 1.0 when the algorithm is initialized. Starting from its value at the beginning of the current output frame, it is decreased at each sample with a slope of 0.054 per 10 ms packet, and multiplied by that sample. This process continues up to the last sample of the output frame or for 20 ms - whichever is first. After 20 ms, the slope is increased to 0.222 per 10 ms packet. The scale is zero after 60 ms of consecutive packet loss. The output reconstructed frame is completely muted after the scale is zero. The value of the scale at the end of the output frame is retained, and used at the following packets.

Consecutive Packet Loss

If the current frame is lost and the previous frame indicator is also set to '1', then we have consecutive packet loss. In this case, a new excitation signal is generated using the previously computed excitation signal and pitch period information for the first lost packet. The excitation signal generator block, the inverse LP filter, overlap-and-add block and scaling block perform the same operations as before. LP analysis, LP filter and pitch detector blocks operation are skipped.

Excitation Generator (Consecutive Packet Losses) :

The last 10 ms segment from the excitation buffer is first copied to the beginning of the new excitation signal. Then, the last P samples (pitch period computed in the first lost packet) of the excitation buffer is appended to the new excitation signal as many times as necessary to fill the remaining portion. This generated excitation signal is then passed through the inverse LP filter. It also replaces the old excitation in the excitation buffer.

First Good Packet after the Erasure

If the current packet is not lost and the previous loss indicator is set ('1'), then the current packet is the first good packet after packet loss and the current packet is modified by an overlap-and-add before playing out as explained below. The output frame is constructed

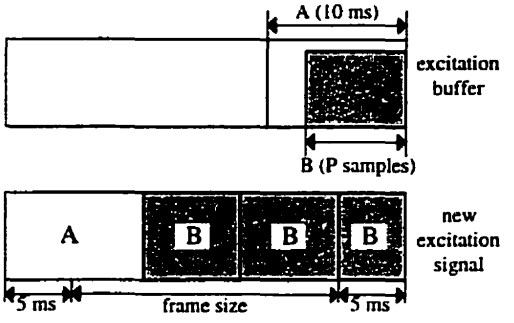


Fig. 5.3 Generating the new excitation signal for consecutive packet loss.

from the last 5 ms segment in the speech buffer and the entire current packet except the last 5 ms segment. The new output frame is scaled and then played-out.

Overlap-and-Add :

The 5 ms segment at the beginning of the current packet and the samples in the overlap buffer are weighted by a triangular window and summed. The resulting signal replaces the 5 ms segment at the beginning of the current packet.

Scaling :

If the value of the scale is less than 1.0, then the output frame is scaled up before being played. The current value of the scale is retained for the duration of the overlap window, and each sample of the output frame is multiplied by the scale. The remaining samples of the output frame are multiplied by the scale and the scale is increased with a slope of 0.498 per 10 ms packet. This continues till the end of the packet or until the scale reaches to unity.

5.2.2 Performance Evaluation

The quality of the reconstructed signal using the described algorithm is high and robust under harsh acoustic conditions. Formal subjective tests were done to evaluate this technique using packet loss up to 5% and packet sizes up to 40 ms, for clean and noisy speech [61]. The results showed a significant improvement over other existing concealment techniques.

Fig 5.4 shows the MOS (Mean Opinion Score) for the LP-based PLC algorithm compared to PLC techniques of other coders, e.g., G.729, G.723.1 against the G.711, without PLC.

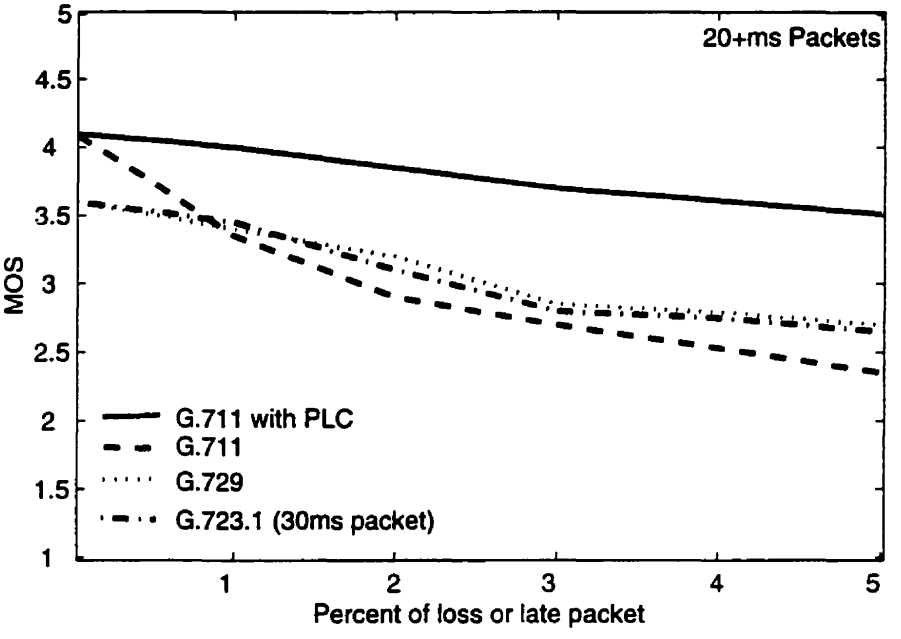


Fig. 5.4 MOS for LP-based PLC algorithm (adapted from [61]).

5.3 Development of a New Algorithm for Packet Loss Concealment

The packet loss concealment technique described above (ANSI T1.521a-2000 (Annex B)) does not consider future packets for the reconstruction of lost packets. At times, the reconstructed signal does not possess a smooth variation at signal transitions from voiced-to-unvoiced or phoneme-to-phoneme. The new technique developed in this research uses future packet information (if available) for the reconstruction of lost packets. If the future packet is not available, the proposed algorithm uses only previous packets. The algorithm uses the Time-Scale Modification (TSM) technique based on WSOLA to generate the best matched signal for the lost segment.

5.3.1 Selection of Variables and Parameters

The packet size considered in the description of the new algorithm is 80 samples (correspond to 10 ms). But, the algorithm can support packet sizes of 5–30 ms with the modification of parameters. It also supports other sampling rates. The state variables and buffer needed for the algorithm are:

- *History buffer:* The most recent 320 samples (40 ms) of speech are kept in the history buffer. The first 300 samples (37.5 ms) of this buffer contain samples that have already been played-out. The last 20 samples (2.5 ms) contain samples that have been received but not yet played.
- *TSM buffer:* This buffer contains 240 samples (30 ms) of speech and is used for the time-scale modification procedure.
- *Future buffer:* This buffer contains 80 samples (10 ms) from the future packet and is used if the future packet is available.
- *Past overlap buffer:* This buffer contains 20 samples (2.5 ms) of signal values prior to the 80 samples of reconstructed signal from the time-scale modified signal.
- *Future overlap buffer:* This buffer contains 20 samples (2.5 ms) of the signal taken after 80 samples of the reconstructed signal from the time-scale modified signal.
- *Future loss indicator:* A binary flag that shows whether the future packet is available or not.
- *Current loss indicator:* A binary flag that shows whether the current packet is lost or not.
- *Previous loss indicator:* A binary flag that indicates whether the previous packet was lost or not.
- *Hanning window:* Symmetric Hanning window used for time-scale modification by WSOLA. The size of this window is 240 samples.

The last 120 samples from the TSM buffer are selected as a modifying signal, also called modifying segment (signal that will be used for time-scale modification). With a packet

size of 80 samples, the algorithm must produce at least 200 samples. The length of the search region to find the maximum correlation of the modifying signal is 100 samples before the modifying segment. This search region is between the 1st and the 100th (corresponds to 2.5–15 ms) sample of the TSM buffer. Fig. 5.5 gives an example of selecting a modifying segment and a search region for time-scale modification.

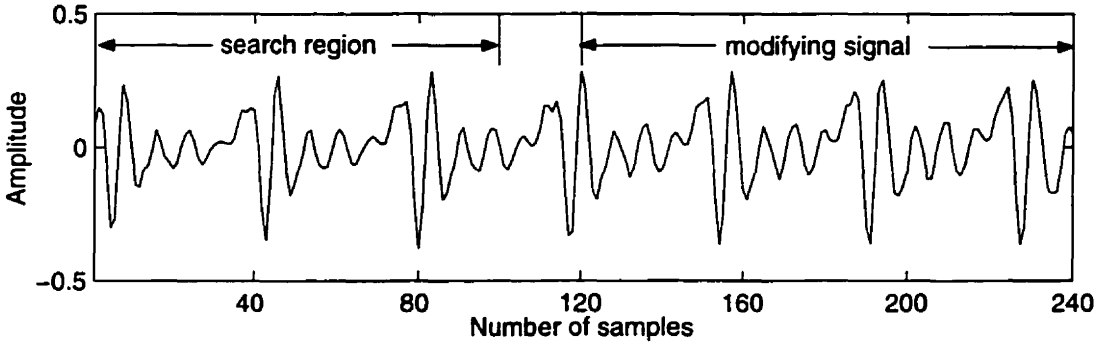


Fig. 5.5 Selection of search segment and modifying segment from the TSM buffer.

5.3.2 Description of the Algorithm

For convenience it has been assumed, in the simulation, that if packet A_n and packet A_{n+1} are lost and the future packet A_{n+2} is not lost, A_n would be reconstructed from the knowledge of packets A_{n-1}, A_{n-2} and A_{n-3}, as depicted in Fig. 5.6. Reconstruction of the packet A_{n+1} is based on packets A_n, A_{n-1} and A_{n+2}. The indicator value '0' indicates a packet has not been lost and '1' indicates a packet has been lost.

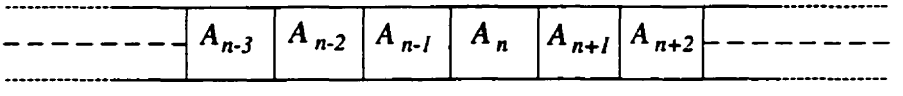


Fig. 5.6 Selection of good packets information for the reconstruction of lost packets. Clear blocks represent the good packets and the shaded blocks represent the lost packets.

5.3.3 Good Packets

If the current loss indicator is '0' (which means the packet is not lost) the decoder decodes the packet and sends it to the output audio port. Before sending the packet to the output ports:

- The content of the history buffer is shifted 80 samples to the left to create space for the current packet.
- The Current packet (80 samples) is copied at the end of the history buffer (see Fig. 5.7).

No other modifications to any buffers are performed.

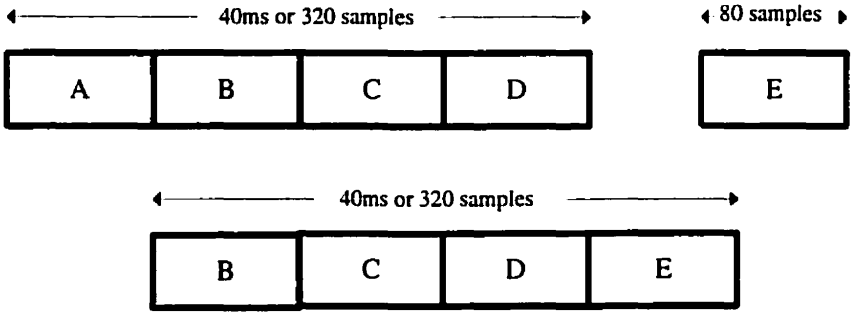


Fig. 5.7 Update of history buffer.

5.3.4 Lost Packets

For the lost packet instance, two different situations can arise.

1. The future packet is not available (future loss indicator is '1').
2. The future packet is available (future loss indicator is '0').

Case 1

When future samples are not available for reconstruction, the algorithm works on the past samples stored in the history buffer (320 samples). Fig. 5.8 shows the block diagram of the reconstruction method using past samples.

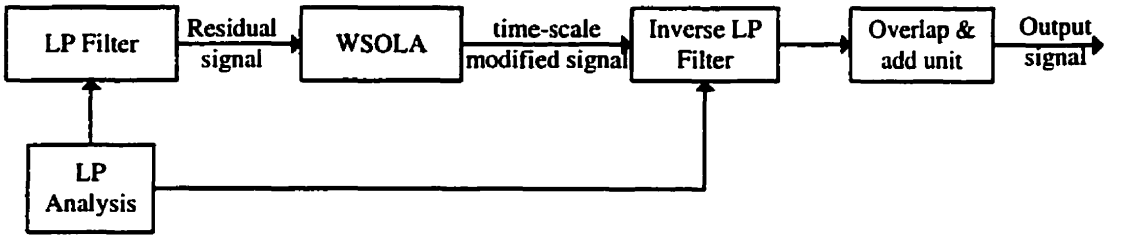


Fig. 5.8 Reconstruction of lost packet using past packets.

LP Analysis and LP Filter :

The operation of this block starts by copying the last 240 samples from the history buffer into the TSM buffer. The LP analysis block computes the LP coefficients of the previous speech frame using 120 samples in the second half of the TSM buffer. An LP order of 10 is used. First, the 120 samples (15 ms) of the TSM buffer are windowed by a symmetric Hamming window then compute the autocorrelation function. The autocorrelation function is checked for stability purposes and voiced-unvoiced detection. If the check fails, or the signal energy is too low, then all zero vectors of LP coefficients are passed to the LP filter and to the inverse LP filter. Otherwise, white noise correction and 60 Hz bandwidth expansion are applied by an exponential lag window. The entire TSM buffer is filtered through an LP filter, and the TSM residual signal is passed to the WSOLA block for time-scale modification. Fig. 5.9 shows the TSM residual signal output of the LP filter block.

Time-Scale Modification (WSOLA) :

The block 'WSOLA' performs the time-scale modification (time-scale expansion) of the last 120 samples from the TSM residual signal and generates the reconstructed signal and the past overlap signal. The description below explains the performed operations.

This block starts its operation by finding the maximum cross correlation of the last 120 samples (which was defined as the modifying signal) within the search region of the first 100 samples from the TSM residual signal. As depicted in Fig. 5.10(a), if the maximum correlation is found at the point $x1$, the next 120 samples of the TSM residual signal from the point $x1$ is multiplied by the second half of the Hanning window (120 samples) and

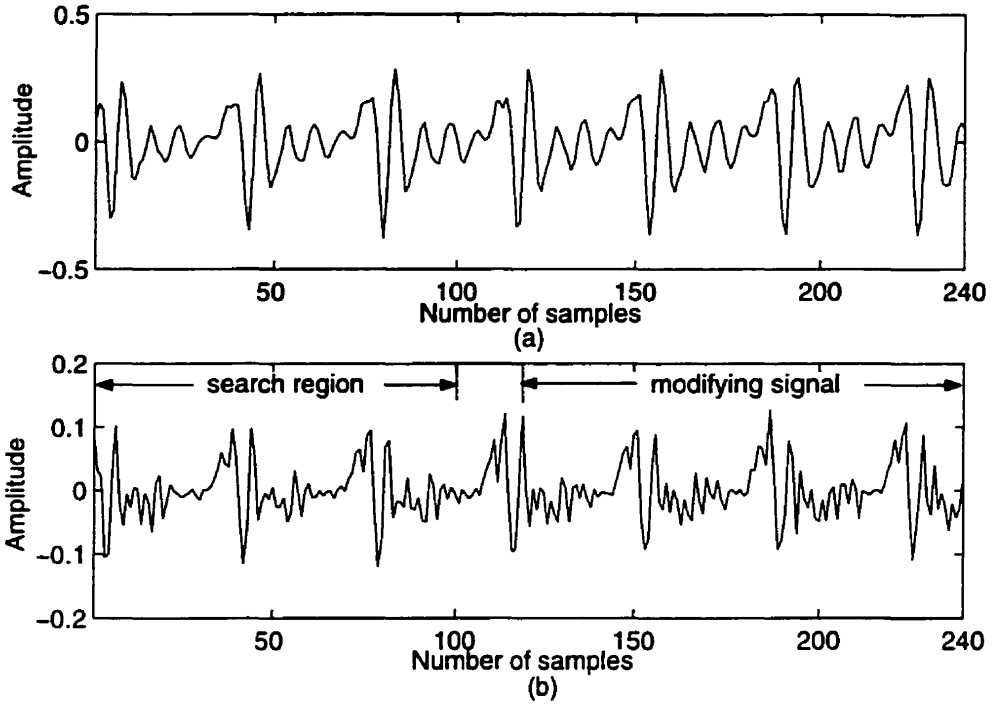


Fig. 5.9 Speech signal from TSM buffer (a) and its corresponding residual signal (b).

added with the modifying signal which is multiplied by the first half of the Hanning window (120 samples). These 120 samples and the samples after the point $x1 + 120$ of the TSM residual signal are concatenated and copied to the TSM output buffer. Fig. 5.10 gives the graphical explanation of the time-scale modification performed.

If the total length of the TSM output signal is greater than or equal to 200 samples, no additional iterations are performed, and search for the segment from this TSM output signal will replace the lost segment. If the length of the TSM output signal is not 200 samples, the modifying signal values are replaced with the new values from the last 120 samples from the TSM output buffer (assuming the last 120 samples start at point $y1$ (see Fig. 5.10(b)) and find the maximum cross-correlation within the same search region as before. If the maximum correlation point now is at $x2$ (as shown in Fig. 5.10(a)), the 120 samples next to the point $x2$ are multiplied by the second half of the Hanning window and added with the windowed version (multiplied with the first half of the Hanning window)

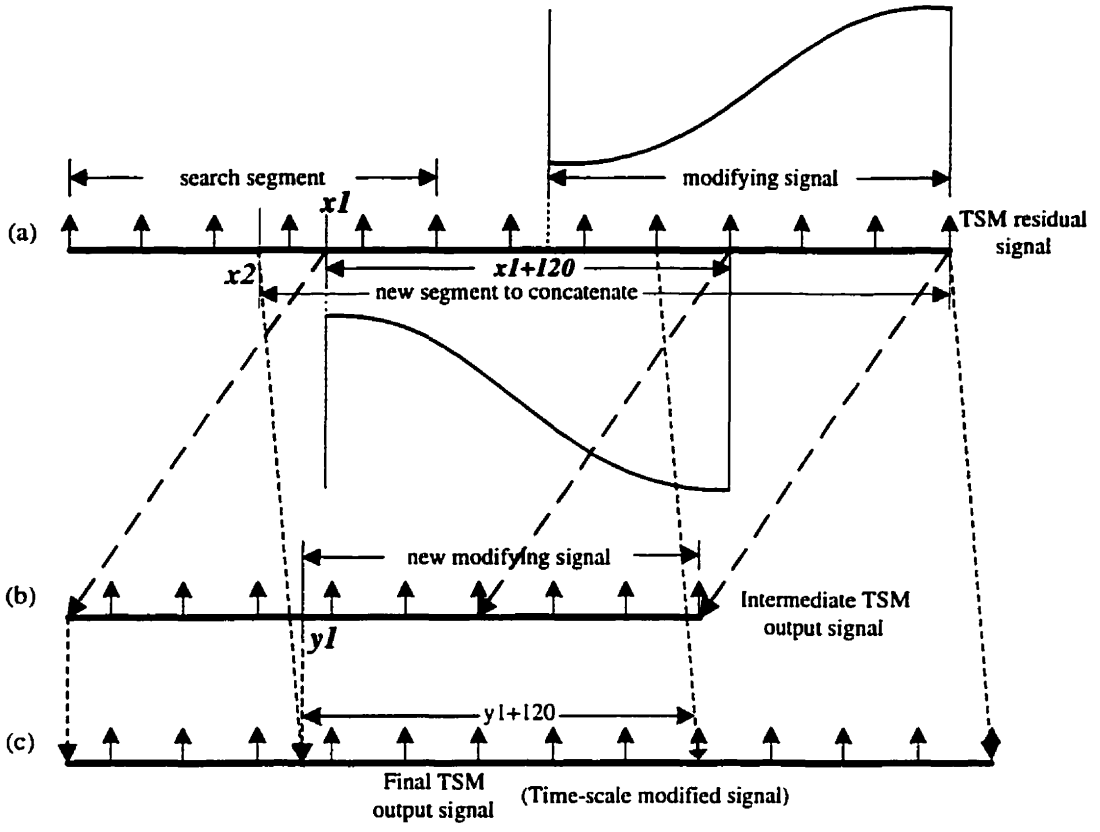


Fig. 5.10 The WSOLA algorithm for time-scale modification. (a) The TSM residual signal from the LP filter block, (b) The TSM output signal achieved in the first iteration (total number of samples are less than 200), and (c) final TSM output signal (Total number of samples are more than 200).

of the new modifying signal as before. These values are concatenated in the TSM output signal at the point $y1$, and the signal values after the point $x2 + 120$ are concatenated at the TSM output signal after the point $y1 + 120$ (in Fig. 5.10(c)).

The length of the TSM output signal is always greater than or equal to 200 samples. Discarding the first 100 samples, the next 100 samples from this TSM output signal are selected as the signal output of this block. The output signal of the 'WSOLA' block is passed to the inverse LP filter. Fig. 5.11 shows the search segment, the modifying signal and the time-scale modified version of the modifying signal. Fig. 5.11(c) shows the 100

samples next to the first 100 samples, output of this block.

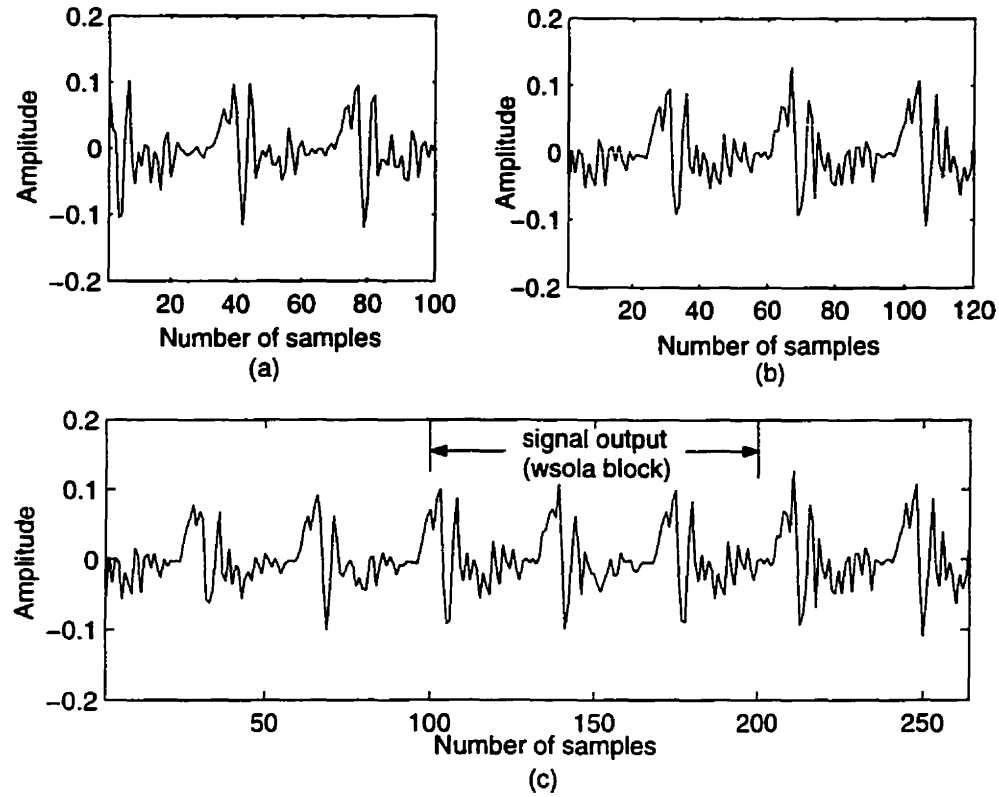


Fig. 5.11 Time scale modification of residual signal using 'WSOLA'. (a) is the search region (100 samples), (b) modifying signal (120 samples), (c) modified signal (264 samples).

Inverse LP Filter :

The output signal is filtered through the inverse LP filter to add the vocal tract information and obtain the time-domain signal from the residual signal. The filter coefficients for this inverse LP filter are taken from the LP analysis filter. The first 20 samples of the output signal, from this block, are copied to the past overlap buffer and passed to the next block for overlap-and-add operation. The last 80 samples replace the lost segment. There are no values for the 'future overlap buffer', as the future packet has also been lost. Fig. 5.12 shows the signal output of the LP synthesis filter block that will replace the lost segment.

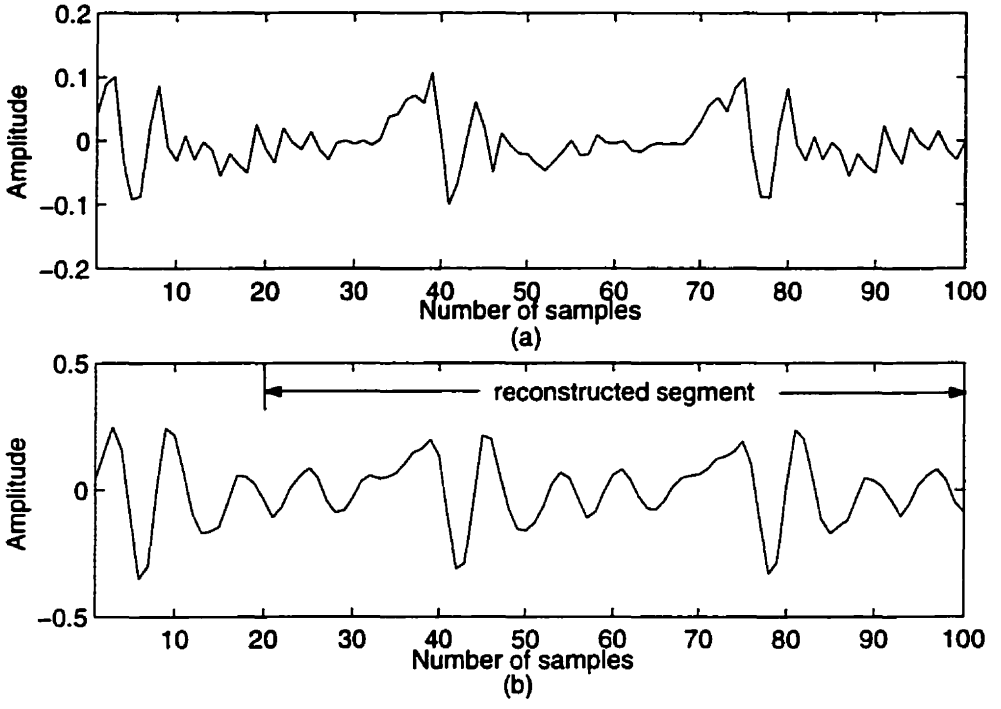


Fig. 5.12 (a) Residual reconstructed signal, (b) output of the inverse LP filter. First 20 samples are used for overlap-and-add with past packet and the last 80 samples replace the lost segment (reconstructed segment).

Overlap-and-Add Unit :

The overlap-and-add unit uses samples from the past overlap buffer and the history buffer. 20 samples (2.5 ms) of the past overlap buffer and the last 20 samples (2.5 ms) of the history buffer are weighted by a triangular window and summed. The added signal replaces the last 20 samples (2.5 ms) of the history buffer. For a packet size of 80 samples (10 ms), the entire history buffer is shifted 80 samples to the left and the last 80 samples from the output signal of inverse LP filter block are copied to the history buffer.

Scaling :

The output frame (80 samples) is scaled down before it is played-out, as explained earlier in the standard ANSI T1.521a-2000 (Annex B) (see Subsection 5.2.1, "*scaling (lost packets)*").

Case 2

If the future packet is available, the proposed concealment algorithm uses future packet information from the future buffer and past packets information from the history buffer for the reconstruction of lost packets. The block diagram for this case is depicted in Fig. 5.13 and the algorithmic description is given below.

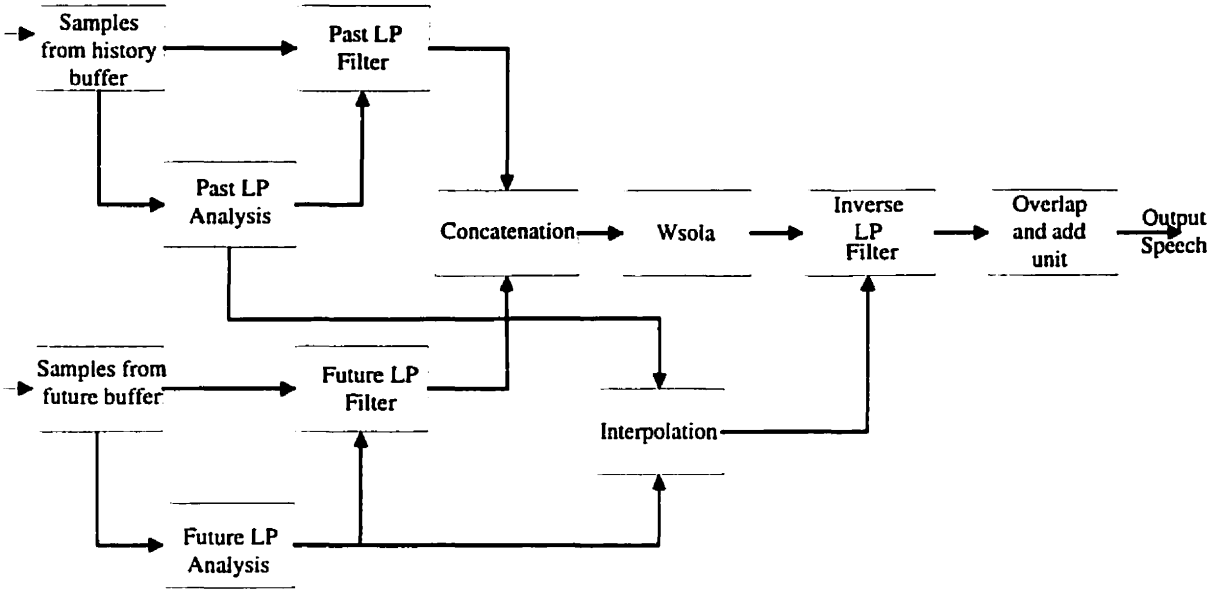


Fig. 5.13 Block diagram for reconstruction algorithm using past and future packets.

LP Analysis and LP Filter :

The past LP analysis block computes the LP coefficients from the history buffer in the same manner as described in *case 1*, and uses them in the past LP filter shown in Fig. 5.13. The future LP analysis block computes LP coefficients from the future buffer which is of 80 samples (10 ms frame) and uses them in the future LP filter. The stability check and bandwidth expansion is done according to the explanation given in *case 1*. The LP order considered in this case is also 10. Autocorrelation function values are also been used for voiced and unvoiced detection. The future frame (80 samples) and the history buffer (320

samples) are passed through the future LP filter and past LP filter to obtain the 'future residual' and 'history residual' signal as depicted in Fig 5.14.

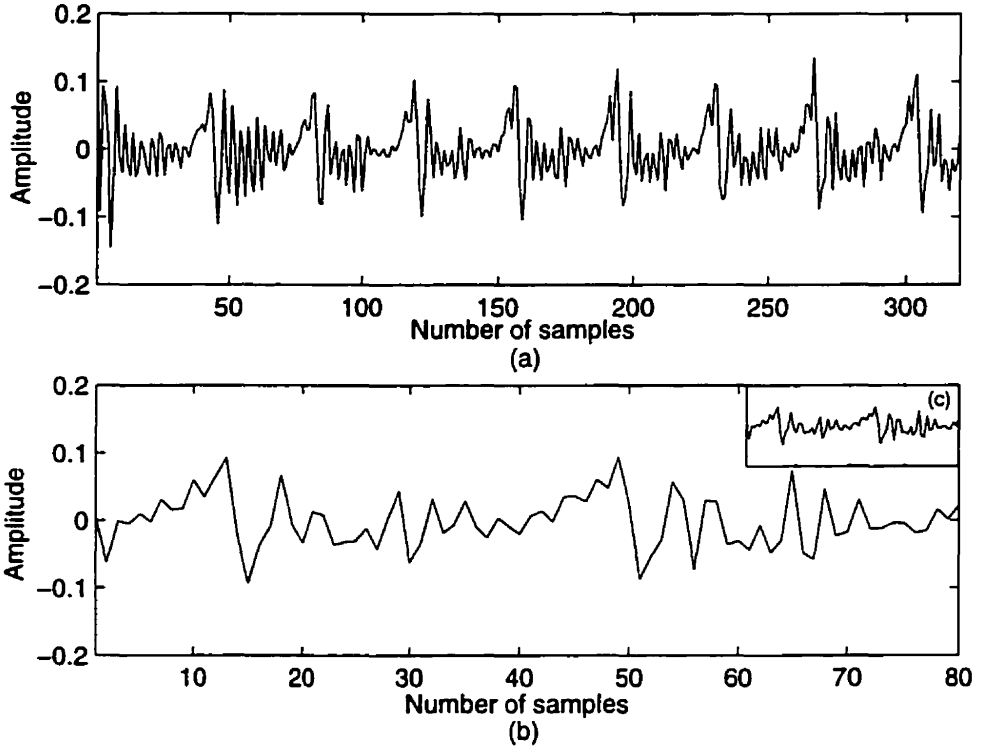


Fig. 5.14 (a) History residual signal, (b) expanded future residual signal, and (c) future residual with the same time scale in (a).

Concatenation of Past and Future Residuals :

To concatenate the future residual signal with the history residual signal, the cross correlation is first computed between the last 160 samples segment from the history residual signal and the future residual signal, then find for the maximum correlation point. Samples next to the maximum correlation point in the history residual signal are replaced with the future residual signal values. In Fig. 5.15, $x(n)$ represents the future residual signal and $y(n)$ represent the history residual signal. If the maximum correlation point is found at point 'a', the values next to this point are discarded from signal $y(n)$ and signal $x(n)$ is

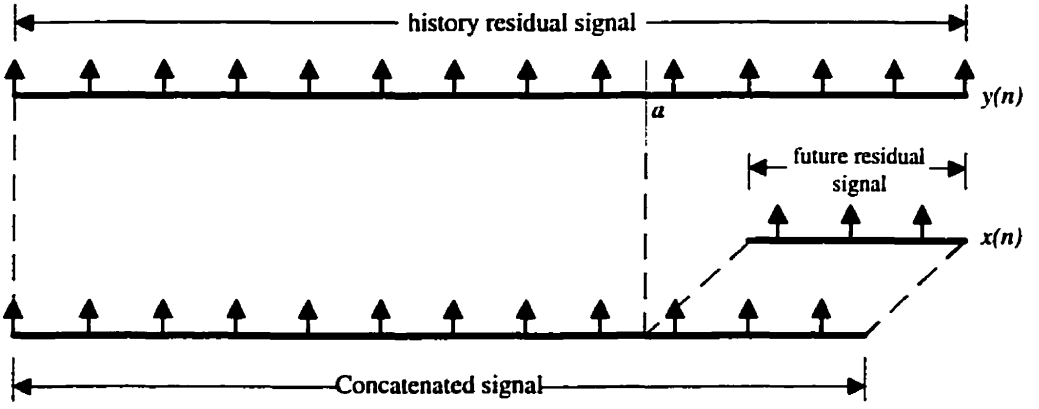


Fig. 5.15 Example of concatenating history residual and future residual.

concatenated with signal $y(n)$ at point 'a'. If either signal is detected as being unvoiced, the future residual signal is concatenated at the end of the history residual signal.

The signal length would vary within the range of 240–400 samples. For the length of 'T' samples of the concatenated signal, last 240 samples are copied to the TSM buffer, and sent to the next block (WSOLA) for time-scale modification. Fig. 5.16 shows the concatenated output signal which is 240 samples long.

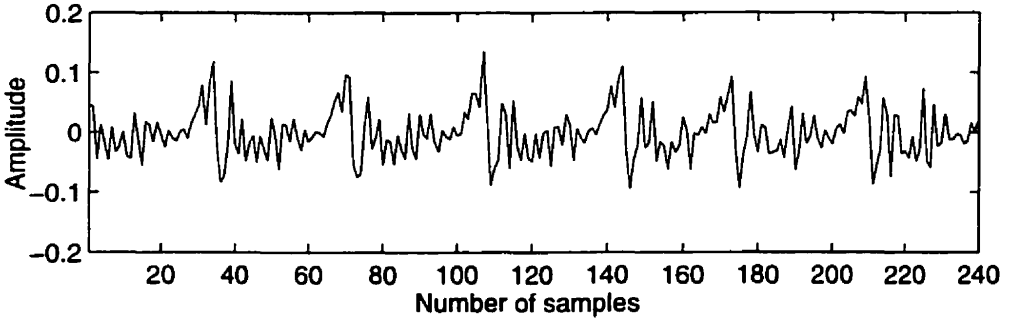


Fig. 5.16 Output signal of concatenation block.

Time-Scale Modification :

Time-scale modification (time-scale expansion) is performed in the same way as described in *case 1.* The TSM output signal is always greater than or equal to 200 samples. At this

time, the first 20 samples are discarded and the next 120 samples are selected as the signal output of the 'WSOLA' block and sent it to the inverse LP filter block to generate samples which will replace the lost samples. Fig. 5.17 shows the time-scale modified signal while using the future packet for reconstruction.

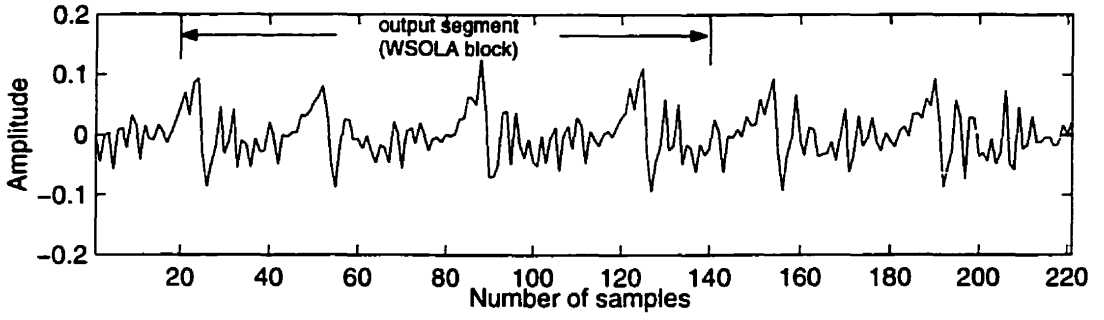


Fig. 5.17 Time scale modified (residual) signal using the future packet.

Inverse LP Filter :

The coefficients for the inverse LP filter are computed by interpolating the LP coefficients of the past and the future LP filter. We have used the LSF spectral representation of the LP coefficients to perform the linear interpolation. The output from the 'WSOLA' block, which is 120 samples long, is passed through the inverse LP filter block to produce the original output signal. The first 20 samples are copied to the past overlap buffer and the last 20 samples are copied to the future overlap buffer. These past and future overlap buffer values are passed to the next block for overlap-and-add operation. Meanwhile, the middle 80 samples replace the lost segment. Fig. 5.18 shows the signal output of the inverse LP filter block, obtained from the time-scale modified signal.

Overlap-and-Add Unit :

The overlap-and-add unit uses samples from the reconstructed signal and samples from the history buffer and future buffer. 20 samples (2.5 ms) from the past overlap buffer and the last 20 samples (2.5 ms) from the history buffer are weighted by a triangular window and summed. The added signal replaces the last 20 samples (2.5 ms) of the history buffer.

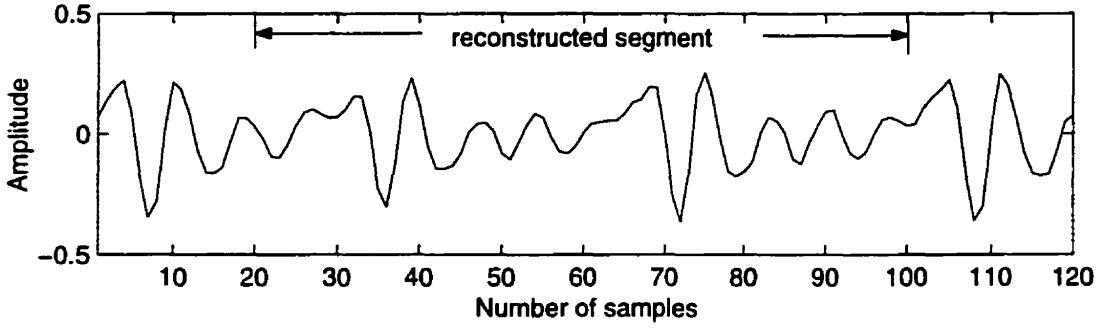


Fig. 5.18 Signal output from the inverse LP filter block (reconstructed segment replaces the lost packet).

Again, 20 samples from the future overlap buffer and the first 20 samples from the future buffer are weighted by the same triangular window and summed. This added signal replaces the first 20 samples of the future buffer.

For a packet size of 80 samples (10 ms), the entire history buffer is shifted 80 samples to the left and the middle 80 samples of the output signal, obtained from the inverse LP filter block, are copied to the end of the history buffer.

Scaling :

The output frame (80 samples) is scaled down before it is played-out as explained in *case 1.*

5.3.5 First Good Packet after the Erasure

At the first good frame after an erasure, a smooth transition is needed between the synthesized erasure speech and the real signal. If the value of the scale is less than 1.0, then the output frame is scaled up before it is played-out to the speaker. Each sample of the output frame is multiplied by the scale and then the scale is increased with a slope of 0.498 per 10 ms packet. This process continues until the last sample of the output frame, or until the scale reaches unity.

5.4 Subjective Test Results and Discussion

In this section we present an evaluation of the PLC techniques described in the previous sections. A comparison between ANSI T1.521a-2000 (Annex B) and the proposed time-scale modification-based PLC technique was conducted by informal subjective listening tests to show the potential of the methods in a real-world application.

The proposed PLC technique supports packet loss rates up to 20% without noticeable distortion between the reconstructed and original signal. Beyond 20%, click and pops are reduced compared to the reconstructed signal using past samples only, and the reconstructed signal is intelligible without annoying sounds. This is a receiver based algorithm and compatible with the G.711 codec.

Subjective Evaluation of the PLC Algorithms :

The test data, used for subjective testing, was chosen to represent a typical conversation over a voice over IP system. Two files consisting of two sentences each were used for this purpose. In one file, two sentences were spoken by two male speakers and in the other file two sentences were spoken by two female speakers. The sentences were originally recorded separately under controlled conditions at a sampling frequency of 8 kHz, using 16-bit linear encoding. The selected sentences in each file were concatenated with a 1 s pause between each sentence.

These two different PLC algorithms were tested using an A–B comparison test. Fourteen test files, 7 male and 7 female, were generated with packet loss rates varying from 5%–35%, in increments of 5%. The packet losses were simulated by randomly dropping packets. Seven sound files were created each for male and female speakers with seven different packet loss rates. Test files were created for all possible combinations, in both presentation orders, of the different speech segments.

An A–B comparison was used to evaluate the efficiency of the proposed algorithm. In this test, each subject was presented with the two output results of two different PLC algorithms: T1.521a-2000 (Annex B) and new time-scale modification based PLC algorithm we have developed. Then they have to indicate whether the first or the second result is preferred[1].

[1] A "No Preference" response was allowed

The sounds were reproduced on a pair of self-powered loudspeakers typical of a computer workstation. The subjects were free to adjust the volume levels to their liking. Playback was directed from the 16-bit linearly encoded sound file to the speaker. The listening test was performed in a small office with some ambient noise, mostly due to the computer workstation.

The test files were presented to 12 subjects (8 male and 4 female) and played-out sequentially for male and female speakers at the same packet loss rate. The subjects were students aged 24 to 28. According to their preferences, the subjects selected one of the two speech segments, each based on a different PLC technique.

Table 5.1 and Table 5.2 shows the test results for the male and the female speakers, with a packet loss rate up to 35%. The entries in the tables refer to the majority decision of the subjects. "Preference" denotes one algorithm being favoured over another while, "Not preferred" indicated the inverse.

Table 5.1 Subjective test results for the male speaker.

Percentage packet loss (%)	T1-521a 2000 (Annex B) algorithm	New time-scale modification-based algorithm
5	No preference	No preference
10	No preference	No preference
15	Not preferred	Preferred
20	Not preferred	Preferred
25	Not preferred	Preferred
30	Not preferred	Preferred
35	Not preferred	Preferred

From the subjective test results in Table 5.1 and Table 5.2, it is clear that the performance of the new algorithm is superior to the existing PLC standard. From Table 5.1 and Table 5.2, we can see that when the packet loss rate is small, the performance of the algorithms is almost identical. For higher packet loss rates, the proposed time-scale modification based PLC algorithm shows better performance.

In Fig. 5.19 the lost segments marked as A, E and F were reconstructed using past packets only. Whereas, lost segments B, C, D, G and H were reconstructed considering future packet information. The reconstructed signals using the time-scale modification-based PLC algorithm in Fig. 5.19(d), the segments B, C, D, E, F, G are very much identical

Table 5.2 Subjective test results for the female speaker.

Percentage packet loss (%)	T1-521a 2000 (Annex B) algorithm	New time-scale modification-based algorithm
5	No preference	No preference
10	No preference	No preference
15	Not preferred	Preferred
20	Not preferred	Preferred
25	Not preferred	Preferred
30	Not preferred	Preferred
35	Not preferred	Preferred

to the original signal than the reconstructed signal using the T1.521a-2000 (Annex B) PLC algorithm. For the segments E, F, G the gain decreases while using the T1-521a PLC algorithm for reconstruction. Whereas, the reconstructed signal values remain almost the same using the time-scale modification based PLC algorithm we have proposed.

It has already been described that the proposed PLC algorithm uses the past and future packets for the reconstruction of missing packets[2]. It can also reconstruct the missing packets based on past packets only. Subjective test evaluation showed, the performance of the time-scale modification based PLC algorithm using only the past packets is similar to the T1-521a-2000 (Annex B) standard.

We have also incorporated the information from future packets into the existing T1.521a standard for reconstruction. This was done by replacing the inverse filter (synthesis filter) coefficients with the interpolated LP coefficients of the past and the future samples. Computed LP coefficients of the future packet were interpolated using LSF's spectral parameter. The graphical presentation showed improvement of the reconstructed signal, but the improvement was not noticeable in the subjective listening test.

Affect of Reconstruction Delay on Packet Loss :

To determine the likelihood of receiving a future packet in time, which can be used for the reconstruction of lost packets in a real network, we have extracted information about packet

[2]Here, the 'missing packets' indicate the discarded packets, which arrive late due to network congestion or are lost due to network errors

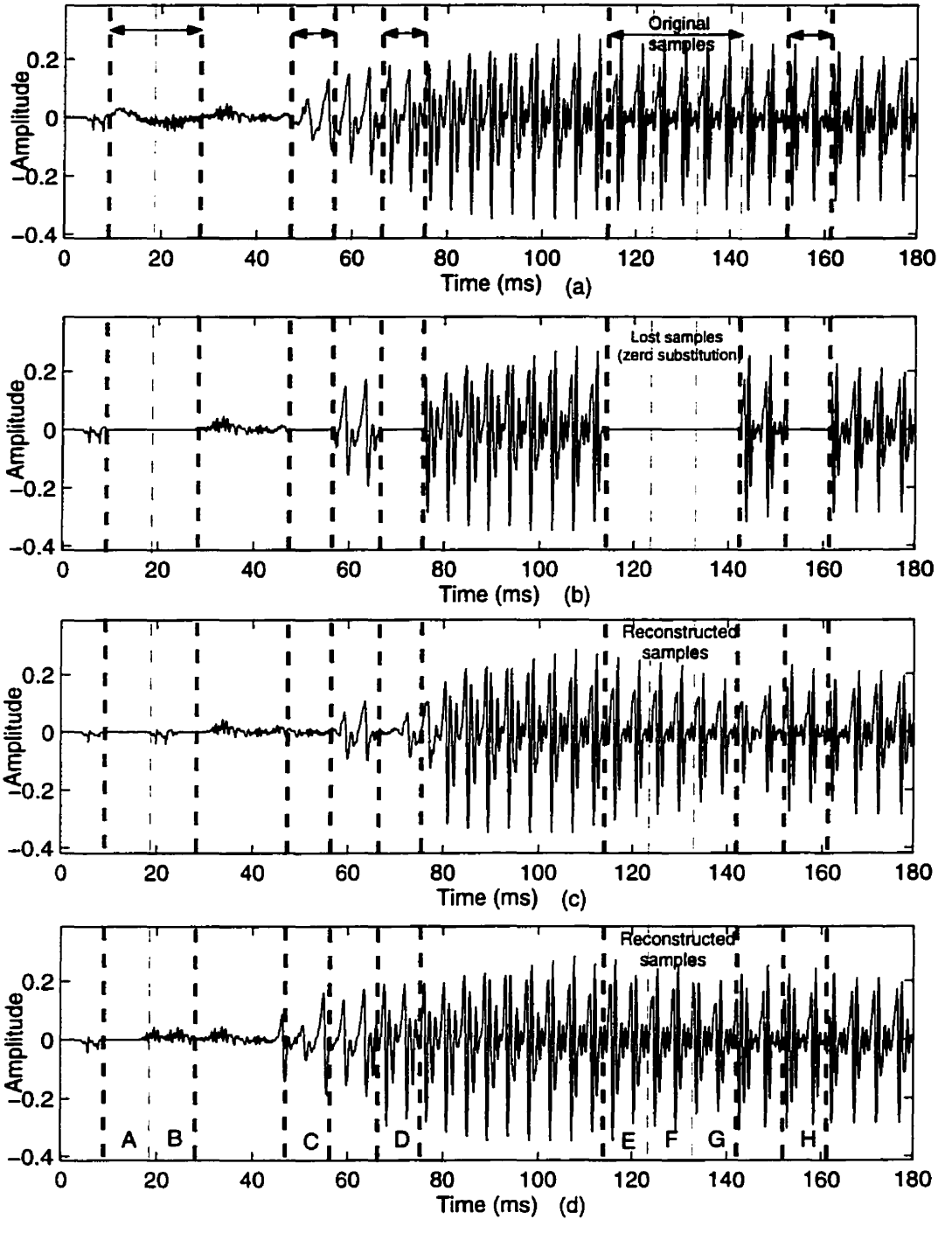


Fig. 5.19 (a) Waveform without packet loss. (b) Signal with lost packet (30%) replaced by zero. (c) Reconstructed signal using T1.521a-2000 algorithm. (d) Reconstructed signal using time-scale modification based PLC algorithm (lost segments 'A', 'E' and 'F' were reconstructed using past samples only and segments 'B', 'C', 'D', 'G' and 'H' were reconstructed using past and future samples (our work)).

loss and delay of a network from a plot given in [51]. In Fig. 5.20, the X-axis represents the relative time of packets transmission at the transmitter and the Y-axis shows the delays. The solid line shows the maximum allowed overall delay, and the dots show the packet delays through the network. The packet size considered in this paper is 20 ms, coded with the standard ITU-T G.711 using 8-bit quantization and 8 kHz sampling rate.

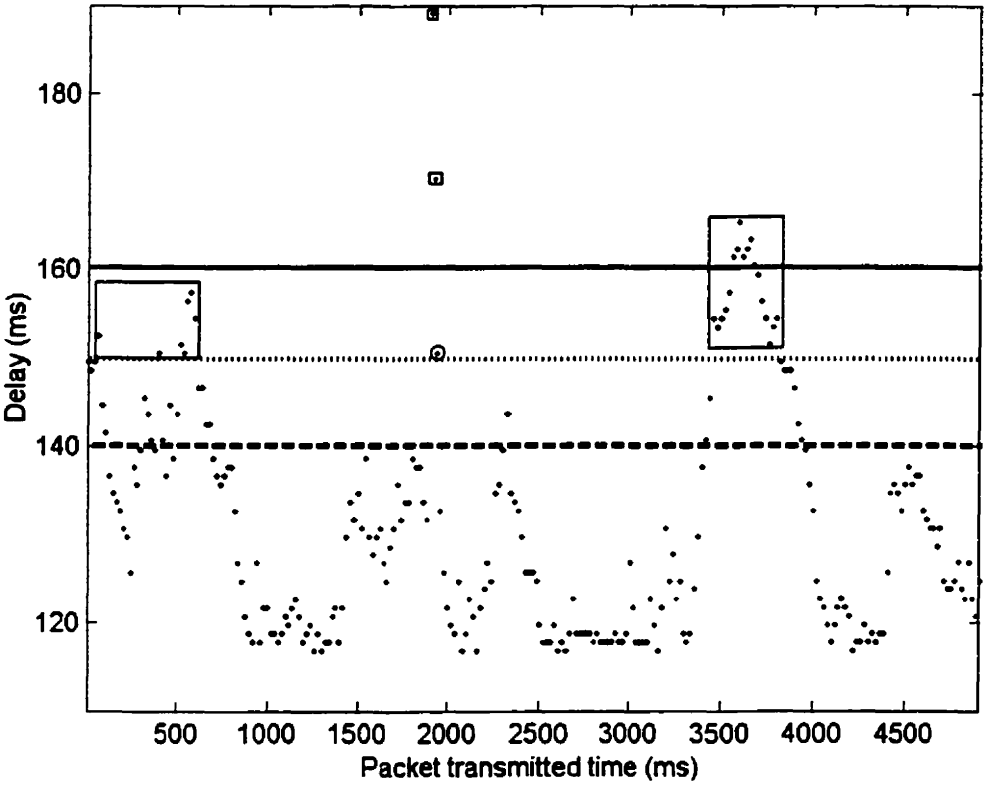


Fig. 5.20 Variation of network delay and the amount of missing packets. Speech packets in the rectangular boxes and the packet marked as a circle are considered missing for allowed end-to-end of 160 ms and and the packet size of 20 ms (160 samples). These data were acquired from [51].

If the minimum playout delay[3] for each packet is 10 ms and the total end-to-end delay (network packet delay plus playout delay) is 160 ms, then a packet has to arrive within the delay limit of 150 ms, otherwise, it is considered as a missing packet. A reconstruction

[3]The 'playout delay' indicates the amount of time a packet has to wait in the buffer before it is played-out.

algorithm is necessary to generate the missing packets. We can always use past packets for reconstruction from the saved speech buffer. The plot shows the variation of the network packet delay where the missing packet rate is 10%, considering the end-to-end delay of 160 ms.

In Fig. 5.20, packets in the rectangular boxes and the packet marked as a circle are considered missing. To reconstruct the missing packets using future packets, the future packet has to arrive within 140 ms after it has been transmitted, because at the transmitter packets are being sent every 20 ms. The circled missing packet has a future packet available for reconstruction. Packets that do not have a future packet available, are reconstructed using past packets only. If the total end-to-end delay is decreased from 160 ms to 140 ms, the percentage of missing packets increases, but more missing packets will have future packets available for reconstruction and hence the perceived voice quality will not decrease significantly. Also, if the packet size is reduced to 10 ms and the playout delay is reduced to 5 ms (because of smaller packet size), with the same amount of packet loss (as given in this plot), there are more missing packets that can be reconstructed considering the future packet. The selection of future packets depends on the look ahead. In our simulation we have considered a look ahead of one 10 ms packet. Shortening the packet sizes will obviously increase the perceived quality of the reconstructed signal, because, future packets are being considered more frequently than before.

Chapter 6

Summary and Future Work

Reliable transmissions of real-time voice stream over the Internet are difficult because the underlying protocol does not support transfer with given Quality of Service (QoS) requirements. TCP is not appropriate for dealing with delay-sensitive multimedia data. By using UDP, which is a connectionless best-effort transport layer protocol, applications need to handle out-of-order packets, packet loss, long delay and delay jitter, that are inherent in the resource limited Internet.

This thesis introduced a packet loss concealment (PLC) technique with the objective of improving the transmitted voice quality over IP networks (the Internet). Waveform Similarity Overlap-Add (WSOLA) based time-scale modification (TSM) has been used to reconstruct the lost packets. By applying the WSOLA algorithm to the residual signal, pitch period information can be kept very close to the original signal. Also the WSOLA algorithm and the output of the synthesis filter makes the transitions smoother between the segments. Thus clicks and pops are reduced and have better quality reconstructed speech compared to the reconstructed signal by other PLC techniques. In this Chapter, the research will be summarized and will suggest the future work that can be done using the proposed PLC technique described earlier.

6.1 Summary of Our Work

After presenting a brief description of the IP networks for voice transmission, Chapter 1 outlines the motivation and scope of this work with previous related research. A brief idea about the approach made in this research is also given in the same chapter. The second

chapter explains the motivation of using IP networks for voice transmission along with the advantages and disadvantages. This chapter also gives the details about the voice-over-IP, describing the protocols used for voice transmission, the QoS provided by the IP protocol and the factors affecting the quality of service. A short description on the implementation of a voice-over-IP system is given at the end of Chapter 2.

Chapter 3 builds a framework for Linear Prediction (LP) analysis and synthesis of the speech signal. The proposed PLC algorithm has been applied to the residual signal. The autocorrelation function has been used to compute LP filter coefficients and to detect the voiced and unvoiced segments. An LSF spectral representation has been used to compute the LSF coefficients for the synthesis filter by interpolating the LP coefficients of past and future frames.

For real-time application (such as, voice-over-IP), Synchronized Overlap-Add (SOLA) and WSOLA time-scale modification techniques are widely used. The basic idea of building these methods start from the simple time-scale modification technique called Overlap-Add (OLA). WSOLA has certain advantages over SOLA and OLA. Chapter 4 describes the mathematical formulation of these three time-scale modification algorithms and describes the drawbacks of OLA and SOLA algorithms and the advantages of using the WSOLA algorithm.

Finally, in Chapter 5, a detailed description of the packet loss concealment algorithm of the ANSI T1.521a-2000 (Annex B) standard and the proposed TSM-based PLC algorithm have been provided. In the same chapter, the comparison between these two PLC algorithms are given by informal subjective listening tests. The reason for using the residual signal for the proposed TSM-based algorithm has also been discussed. The chapter ends with the description of a plot, showing the variation of packet loss with the change of delay variance, and the chance of considering future packets for the reconstruction of missing packets.

6.2 Motivation for Future Work

Prioritizing Packets :

If an IP network provides a certain level of QoS, coded and packetized audio signals can be prioritized by determining the most informative packets of a voiced segment. Transmitters

can send only the prioritized packets through the network. At the receiver, it may be possible to reconstruct the whole voiced signal which closely matches the original signal waveform, using the packet loss concealment algorithms described in this research. By implementing this idea less information is to be sent over the network which helps to use the network resources effectively.

State Information :

The proposed TSM-based PLC algorithm has been implemented for use with the G.711 coder. This algorithm may be adapted for use with other coders that do not provide PLC, such as ITU-T Rec. G.726 [62]. G.711 does not need any state information [28], but almost all other speech coders have such information. When using this algorithm with other coders that need state information, it is important to maintain that state information in the coder. If maintaining the coder's state variable is not an issue, the PLC algorithm described in this thesis can be used to generate synthetic speech during an erasure. The coder's internal state variables have to track the synthetic speech; otherwise, after the erasure noticeable artifacts and discontinuities will appear as the decoder would have incorrect *a priori* state information. Better results can be obtained by having the decoder's state variables track the synthesized speech during the erasure. This would require converting the decoder into an encoder for the duration of an erasure and using the synthesized output of the concealment algorithm as the input to the encoder. Unlike typical encoders, this encoder is only run to maintain the state information and its output is never used. So some technique may be incorporated to this encoder which can help in updating the state information such that artifacts are avoided when the erasure is over.

References

[1] H. Schulzrinne, *Voice communication across the Internet: A Network Voice Terminal.* Department of Electrical and Computer Engineering and Department of Computer Science, University of Massachusetts, Amherst, Massachusetts, July 1992.

[2] O. Hodson, S. Varakliotis, and V. Hardman, "A software platform for multiway audio distribution over the Internet," *IEE Colloquium on Audio and Music Technology: The Challenges of Creative DSP*, pp. 114–116, Nov. 1998.

[3] INRIA, *Free Phone*, Feb. 1999. *http://www-sop.inria.fr/rodeo/fphone/index.html.*

[4] N. Jayant and S. W. Christensen, "Effect of packet losses in waveform coded speech and improvement due to an add-even sample-interpolation procedure," *IEEE Trans. Communications*, vol. 29, pp. 101–109, Feb. 1981.

[5] D. O'Shaughnessy, *Speech Communications: Human and Machine.* New York: IEEE Press, 2nd ed., 2000.

[6] A. Watson and M. Sasse, "Measuring perceived quality of speech and video in multimedia conferencing applications," *Proc. of ACM Multimedia*, pp. 55–60, Sept. 1998.

[7] H. Shulzrinne, S. Casner, R. Frederick, and V. Jacobsen, *A Transport Protocol for Realtime Applications.* Network Working Group RFC:1889, Internet Engineering Task Force, Jan. 1996.

[8] V. Hardman, M. Sasse, M. Handley, and A. Watson, "Reliable audio for use over the Internet," *Proc. of Internet Networking (INET)*, 1995.

[9] T. Yletyinen, "Quality of voice over IP," Master's thesis, Helsinki University of Technology, Telecomm tech., 1997.

[10] J. Suzuki and M. Taka, "Missing packet recovery techniques for low-bit-rate coded speech," *IEEE J. Select. Areas Commun.*, vol. 7, pp. 707–717, June 1989.

[11] O. J. Wasem, D. J. Goodman, C. A. Dvordak, and H. G. Page, "The effect of waveform substitution on the quality of PCM packet communications," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 36, pp. 342–348, Mar. 1988.

[12] ANSI, *Packet Loss Concealment for use with ITU-T Recommendation G.711*, Dec. 1999. ANSI Recommendation T1.521-1999 (Annex A).

[13] INRIA Sophia Antipolis, *Packet Reconstruction in Free Phone*, Feb. 1999. *http://www-sop.inria.fr/rodeo/fphone/redund.html.*

[14] L. A. DaSilva, D. W. Petr, and V. S. Frost, "A class-oriented replacement technique for lost packets," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 37, pp. 1597–1600, Oct. 1989.

[15] C. Padhye, K. Christensen, and W. Moreno, "A new adaptive FEC loss control algorithm for voice over IP applications," *Performance, Computing, and Commun. Conf. Proc. of the IEEE Int.*, pp. 307–313, 2000.

[16] ANSI, *Packet Loss Concealment for use with ITU-T Recommendation G.711*, July 2000. ANSI Recommendation T1.521a-2000 (Annex B).

[17] W. Verhelst and M. Roelands, "An overlap-add technique based on waveform similarity (WSOLA) for high quality time-scale modification of speech," *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing* (Minneapolis, Minnesota), pp. 554–557, Apr. 1993.

[18] H. Sanneck, A. Stenger, K. B. Younes, and B. Girod, "A new technique for audio packet loss concealment," *Proc. IEEE Global Telecom. Conf. and Exhibition* (London, UK), pp. 48–52, Nov. 1996.

[19] M. S. Shuster, "Diffusion of network innovation: Implications for adoption of Internet services," Master's thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, June 1998.

[20] D. Lin, "Real time voice transmissions over the Internet," Master's thesis, University of Illinois at Urbana–Champaign, Urbana, Illinois, 1999.

[21] G. Held, *Voice Over Data Networks.* New York: McGraw–Hill, 1998.

[22] Lucent Technologies, *Voice over IP: An Overview for Enterprise Organizations and Carriers. http://www.lucent.com/livelink/whitepaper.pdf.*

[23] Ericsson News Centre, *World first for Voice-over-IP over WCDMA*, Feb. 2001. *http://www.ericsson.com/infocenter/news/voice_over_ip.html.*

[24] W. R. Stevens, *TCP/IP Illustrated.* Addison-Wesley, 3rd ed., 1997.

[25] J. Davidson and J. Peters, *Voice over IP Fundamentals.* Indianapolis, Indiana: Cisco Press, 1st ed., 2000.

[26] C. Huitema, *IPv6: The New Internet Protocol.* Prentice Hall, 1996.

[27] R. Braden, *Resource Reservation Protocol [RSVP]-Version 1, Functional Specification.* Network Working Group RFC:2205, Internet Engineering Task Force, Sept. 1997.

[28] ITU-T, *Pulse Code Modulation (PCM) of voice frequencies*, Nov. 1988. ITU-T Recommendation G.711.

[29] ITU-T, *Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s*, Mar. 1996. ITU-T Recommendation G.723.1.

[30] ITU-T, *Coding of Speech at 8 kbit/s using Conjugate-Structure Algebraic-Code-Excited Linear-Prediction (CS-ACELP)*, Mar. 1996. ITU-T Recommendation G.729.

[31] ITU-T, *Reduced complexity 8 kbit/s CS-ACELP speech codec*, Nov. 1996. ITU-T Recommendation G.729A.

[32] W. B. Kleijn and K. K. Paliwal, eds., *Speech Coding and Synthesis.* Elsevier, 1995.

[33] ITU-T, *Packet-based multimedia communications systems - To be published*, Nov. 2000. ITU-T Recommendation H.323.

[34] G. A. Thom, "H.323: The multimedia communication standard for local area networks," *IEEE Communications Magazine*, Dec. 1996.

[35] Hyperlink, *H.323 Overview: Multimedia Teleconferencing Standards for Voice over IP. http://www.imtc.org/h323.htm.*

[36] ITU-T, *Call signalling protocols and media stream packetization for packet-based multimedia communication systems - To be published*, Nov. 2000. ITU-T Recommendation H.225.0.

[37] ITU-T, *Gateway control protocol*, June 2000. ITU-T Recommendation H.248.

[38] D. Minoli and E. Minoli, *Delivering Voice over IP Networks.* John Wiley and Sons, Inc., 1st ed., 1998.

[39] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: principles, algorithms and applications.* Macmillan Publishing Company, 1992.

[40] S. Haykin, *Adaptive Filter Theory.* New Jersey: Prentice-Hall, 3rd ed., 1996.

[41] G. H. Golub and C. F. V. Loan, *Matrix Computations*. The Johns Hopkins University Press, 2nd ed., 1989.

[42] Y. Tohkura, F. Itakura, and S. Hashimoto, "Spectral smoothing technique in PARCOR speech analysis-synthesis," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-26, pp. 587–596, Dec. 1978.

[43] R. Viswanathan and J. Makhoul, "Quantization properties of transmission parameters in linear predicitive systems," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-23, pp. 309–321, June 1975.

[44] J. S. Erkelens and P. M. T. Broersen, "Analysis of spectral interpolation with weighting dependent on frame energy," *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing* (Adelaide, Australia), pp. 481–484, Apr. 1994.

[45] R. Gnanasekaran, "A note on the new 1-d and 2-d stability theorems for discrete systems," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 29, pp. 1211–1212, Dec. 1981.

[46] F. K. Soong and B.-H. Juang, "Line Spectrum Pair (LSP) and speech data compression," *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing* (San Diego, California), pp. 1.10.1–1.10.4, Mar. 1984.

[47] F. K. Soong and B.-H. Juang, "Optimal quantization of LSP parameter," *IEEE Trans. Speech and Audio Processing*, vol. 1, pp. 15–24, Jan. 1993.

[48] P. Kabal and R. P. Ramachandran, "The computation of line spectral frequencies using Chebyshev polynomials," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-34, pp. 1419–1426, Dec. 1986.

[49] H. K. Kim and H. S. Lee, "Interlacing properties of line spectrum pair frequencies," *IEEE Trans. Speech and Audio Processing*, vol. 7, pp. 87–91, Jan. 1999.

[50] T. Islam and P. Kabal, "Partial-energy weighted interpolation of linear prediction coefficients," *Proc. IEEE Workshop on Speech Coding* (Delavan, Wisconsin), pp. 105–107, Sept. 2000.

[51] Y. J. Liang, N. Farber, and B. Girod, "Adaptive playout scheduling using time-scale modification in packet voice communications," *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing* (Utah, USA), May 2001.

[52] H. Sanneck, , K. B. Younes, R. Reng, and B. Girod, "A new error concealment technique for audio transmission with packet loss," *European Signal Processing Conference* (Trieste, Italy), Sept. 1996.

[53] F. Liu, J. W. Kim, and C. C. J. Kuo, "Adaptive delay concealment for Internet voice applications with packet-based time-scale modification," *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing* (Utah, USA), May 2001.

[54] M. Covell, M. Slaney, and A. Rothstein, "FastMPEG: Time-Scale Modification of Bit-Compressed Audio Information," *Interval Research Corporation* (San Jose, USA), 2000.

[55] D. Malah, "Time-domain algorithm for harmonic bandwidth reduction and time scaling of speech signals," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-27, pp. 121–133, Apr. 1979.

[56] S. Roucos and A. M. Wilgis, "High quality time-scale modification for speech," *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing* (Tokyo, Japan), vol. AU-16, No.2, pp. 262–266, Mar. 1985.

[57] J. R. Deller, *Discrete-Time Processing of Speech Signals.* Englewood Cliffs, NJ: Prentice-Hall, 1993.

[58] D. W. Griffin and J. S. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-32, pp. 236–243, Apr. 1984.

[59] E. Moulines and W. Verhelst, "Time-domain and frequency-domain techniques for prosodic modification of speech," in *Speech Coding and Synthesis* (W. B. Kleijn and K. K. Paliwal, eds.), pp. 536–540, Elsevier, 1995.

[60] ITU-T, *Coding of speech at 16 kbit/s using low-delay code excited linear prediction*, Sept. 1992. ITU-T Recommendation G.728.

[61] Nortel Networks Corporation, *Packet Loss and Packet Loss Concealment Technical Brief. http://www.nortelnetworks.com/products/01/succession/es/doclib.html.*

[62] ITU-T, *40, 32, 24, 16 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM)*, Dec. 1990. ITU-T Recommendation G.726.