ERROR CONCEALMENT FOR SPEECH OVER NOISY CHANNELS BY USING AUDIO WATERMARKING

by

Fatih Er

B.S., in CmpE, Boğaziçi University, 1999

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2008

ERROR CONCEALMENT FOR SPEECH OVER NOISY CHANNELS BY USING AUDIO WATERMARKING

APPROVED BY:

Assoc.Prof. Fatih Alagöz (Thesis Supervisor)	.....
Prof. Fikret Gürgen (Thesis Co-supervisor)	.....
Prof. Emin Anarım	.....
Prof. Ufuk Çağlayan	.....
Asst.Prof. Tuna Tuğcu	.....

DATE OF APPROVAL: 19.09.2008

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my thesis adviser, Assoc.Prof. Fatih Alagöz and to my thesis co-adviser Prof. Fikret Gürgen for their invaluable guidance and help during the preparation of this dissertation. I would like to mention their patience, giving me inspiration and hope when I was stuck at dead-ends.

Also I would like to thank to Gürkan Gür for his support and valuable suggestions.

This thesis is dedicated to my wife & son.

This work has been supported by the State Planning Organization (DPT) of Republic of Turkey under the project number 2007K120610.

ABSTRACT

ERROR CONCEALMENT FOR SPEECH OVER NOISY CHANNELS BY USING AUDIO WATERMARKING

In this thesis, the author presents a novel error concealment method of speech for noisy channels using audio watermarking. The proposed error concealment algorithm abbreviated as ECAW is basically a LSB-based scheme and embeds the k-means clustering of (n)th frame into (n-1)st frame before audio encoding. After the transmission, the erroneous or lost parts of (n)th frame is concealed by the clustering information embedded into (n-1)st frame.

Before sending speech signal into channel, ECAW's error concealment strategy divides speech signal into frames of 5 ms and executes k-means clustering with k=32. Clustering information of each frame is embedded into the previous frame. This introduces a constant delay of 5 ms to channel. Then, on the receiver side, received speech signal is broken into frames of 5 ms, too and k-means clustering with k=32 again is executed. It's assumed that error detection is provided externally and if any error is detected, ECAW's error concealment strategy replace the erroneous part with the corresponding part of the center of assigned k-means cluster.

Experimental results prove the efficacy of the proposed method in reducing distortions despite some restrictions.

ÖZET

GÜRÜLTÜLÜ KANALLARDA KONUŞMA İLETİMİNDE OLUŞAN HATALARI SES DAMGALAMA TEKNİĞİNİ KULLANARAK GİZLEMEK

Bu tezde konuşma için, gürültülü kanallarda ses damgalama teknikleri kullanılarak oluşturulan yeni bir hata gizleme metodu sunulmuştur. Kısaca ECAW olarak anılacak olan hata gizleme yöntemi LSB(en önemsiz bit)-tabanlı bir ses damgalama tekniği kullanarak n. çerçeveye ait küme bilgisini, ses kodlaması öncesinde (n-1). çerçeveye gömmektedir. Transfer sonrasında hatalı ya da kayıp çerçeve, (n-1). çerçeveye gömülmüş olan küme bilgisi kullanılarak gizlenmektedir.

ECAW'ın hata gizleme stratejisi, ses sinyalini, kanala göndermeden önce 5 ms lik çerçevelere böler ve k=32 değeri ile k-means kümeleme yöntemini çalıştırır. Her bir çerçeveye ait kümeleme bilgisi bir önceki çerçeveye gömülür. Bu, kanala 5 ms lik sabit bir gecikme tanıtır. Alıcı tarafta da, gelen ses sinyali 5 ms lik çerçevelere ayrılır ve k=32 değeri ile k-means kümeleme yöntemi tekrar çalıştırılır. Hata tespit etme yönteminin harici olarak sağlandığı farz edilmektedir ve eğer herhangi bir hata tespit edilirse, ECAW'ın hata gizleme stratejisi, hatalı bölümü, bölüme ait k-means kümesine ait merkezin karşılık gelen bölümü ile değiştirir.

Deney sonuçları bazı kısıtlar ile birlikte önerilen yöntemin etkin olduğunu göstermiştir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . iii
ABSTRACT . . . iv
ÖZET . . . v
LIST OF FIGURES . . . ix
LIST OF TABLES . . . xi
LIST OF ABBREVIATIONS . . . xii
1. INTRODUCTION . . . 1
1.1. Speech Communication in General . . . 1
1.2. Audio Compression . . . 1
1.3. Communication Errors and Error Detection . . . 3
1.4. Problem Statement . . . 4
1.5. Reference Error Concealment Methods . . . 4
1.5.1. Fading Pattern Repetition (FPR) . . . 4
1.5.2. Error Concealment Technique for VoIP Based on Forward Contour Prediction (ECFCP) . . . 7
1.6. Outline of the Thesis . . . 7
2. BACKGROUND . . . 8
2.1. Existing Error Resilience and Error Concealment Techniques for the Transmission of Audio . . . 8
2.1.1. Sender-Based Techniques . . . 8
2.1.2. Receiver Based Techniques . . . 9
2.1.3. Sender/Receiver Based Techniques . . . 9
2.2. Watermarking . . . 10
2.2.1. Ownership Protection . . . 11
2.2.2. Proof of Ownership . . . 11
2.2.3. Authentication and Tampering Detection . . . 12
2.2.4. Fingerprinting . . . 13
2.2.5. Broadcast Monitoring . . . 13
2.2.6. Copy Control and Access Control . . . 14

2.2.7. Information Carrier 14
2.3. Requirements of Digital Watermarking 14
2.3.1. Capacity 15
2.3.2. Security 15
2.3.3. Robustness 15
2.3.4. Imperceptibility 16
2.3.5. Public versus Private Watermarking 16
2.4. Audio Watermarking 16
2.4.1. Overview of the Properties of the Human Auditory System (HAS 16
2.5. K-Means Clustering 18
3. PROPOSED ERROR CONCEALMENT METHOD 19
3.1. Sender Side 20
3.2. Receiver Side 20
3.3. Integrated Audio Watermarking Techniques 21
3.3.1. LSB Coding Technique 21
3.3.2. Phase Changing Technique 22
3.3.3. DC-Level Shifting Technique 23
3.3.4. Direct Sequence Spread Spectrum (DSSS) Technique 25
3.3.5. Frequency Hopping Spread Spectrum (FHSS) Technique 25
3.3.6. Time Hopping Spread Spectrum Technique 26
3.3.7. Cepstrum Domain Technique 26
3.3.8. Frequency Masking Technique 27
3.4. Integrated Audio Encoding Techniques 28
3.4.1. PCM 28
3.4.2. ADPCM 28
3.4.3. GSM 6.10 29
3.4.4. DSP Group TrueSpeech 29
3.4.5. MPEG-1 Audio Layer 3 (MP3) 30
3.5. Simulated Channel Model 31
3.5.1. Gilbert-Elliot Channel Model 31
4. EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS 33

4.1. Interaction between Audio Encoding Types and Audio Watermark Techniques . . . 33
4.2. Simulation Environment . . . 33
4.2.1. Development Environment . . . 33
4.2.2. Audio Watermarking Method . . . 33
4.2.3. Channel Model . . . 34
4.2.4. Audio File Encoding . . . 34
4.2.5. Sample Clips . . . 34
4.2.6. Reference Methods for Comparison . . . 37
4.3. Simulation Results . . . 38
4.3.1. Audio Quality after Watermark Embedding . . . 38
4.3.2. SNR (Signal-to-Noise Ratio) Evaluation . . . 38
4.4. Adaptive Error Concealment (AEC) . . . 51
4.4.1. SNR (Signal-to-Noise Ratio) Evaluation . . . 51
5. FUTURE WORK . . . 56
6. CONCLUSIONS . . . 57
REFERENCES . . . 58

LIST OF FIGURES

Figure 1.1. Speech communication 2

Figure 1.2. Error concealment using the FPR strategy : i.original sequence ii.lost packet substituted with silence iii.PR strategy iv.FPR strategy 5

Figure 1.3. Error concealment using the ECFCP strategy. Six conditions for the current packet contour prediction based on the last two packets: i.the tendency should be going on increasing; ii.the tendency should be decreasing after being flat; iii.the tendency should be going on decreasing; iv.the tendency should be going on decreasing; v.the tendency should be going on decreasing until becoming silence. vi.the tendency should be going on increasing. 6

Figure 2.1. Watermark embedding and extraction 11

Figure 3.1. ECAW Schema 19

Figure 3.2. Gilbert channel model. Subscript b is for bad state, whereas g stands for good state and p_{ij} is the transition probability from state i to j where i,j $\in$ {b,g} 31

Figure 4.1. Sample clips used in the simulations. 35

Figure 4.2. Sample clips used in the simulations. 36

Figure 4.3. Auto-correlation sequence of each clip 37

Figure 4.4. Performance results for various clips at good channel conditions for 16 bits packet loss. 40

Figure 4.5. Performance results for various clips at bad channel conditions for 16 bits packet loss. 41

Figure 4.6. Performance results for various clips at good channel conditions for 32 bits packet loss. 43

Figure 4.7. Performance results for various clips at bad channel conditions for 32 bits packet loss. 44

Figure 4.8. Performance results for various clips at good channel conditions for 64 bits packet loss. 46

Figure 4.9. Performance results for various clips at bad channel conditions for 64 bits packet loss. 47

Figure 4.10. Performance results for various clips at good channel conditions for 128 bits packet loss. 49

Figure 4.11. Performance results for various clips at bad channel conditions for 128 bits packet loss. 50

Figure 4.12. Performance results for various clips at good channel conditions for various sized packet loss. 54

Figure 4.13. Performance results for various clips at bad channel conditions for various sized packet loss. 55

LIST OF TABLES

Table 4.1. SNR values (in dB) after watermark embedding 38

Table 4.2. SNR comparison (in dB) for 16 bits packet loss 39

Table 4.3. SNR comparison (in dB) for 32 bits packet loss 42

Table 4.4. SNR comparison (in dB) for 64 bits packet loss 45

Table 4.5. SNR comparison (in dB) for 128 bits packet loss 48

Table 4.6. SNR comparison (in dB) for various sized packet loss. 52

LIST OF ABBREVIATIONS

ADPCM	Adaptive Differential Pulse Code Modulation
AEC	Adaptive Error Concealment
AWGN	Additive White Gaussian Noise
Bark	Bandwidth of one critical band
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DSSS	Direct Sequence Spread Spectrum technique
EC	Error Concealment
ECAW	Error Concealment for speech over noisy channels by using Audio Watermarking
ECFCP	Error Concealment Technique for VoIP Based on Forward Contour Prediction
FCP	Fading Pattern Repetition
FEC	Forward Error Correction
FFT	Fast Fourier Transform
FHSS	Frequency Hopping Spread Spectrum technique
FM	Frequency Masking
GSM	Global System for Mobile communication
HAS	Human Auditory System
HVS	Human Visual System
ITU	Intenational Telecommunication Union
JPEG	Joint Picture Experts Group
LSB	Least Significant Bit
LMS	Land Mobile Satellite
MMT	Minimum Masking Threshold
PCM	Pulse Code Modulation
PHASE	Phase changing technique
PN	Pseudonoise
SNR	Signal-to-Noise Ratio

THSS	Time Hopping Spread Spectrum technique
WM	Watermarking

1. INTRODUCTION

Wireless channels are usually faced with burst error conditions, i.e., errors that are likely to occur in clusters. In addition other sources of distortion, such as congestion in packet switched networks, are also in question. Up to the present some error concealment (EC) techniques have been proposed to cope with these signal degradations. The main purpose of these EC techniques is to obtain a close approximation of the original signal or attempt to make the output signal at the decoder the least objectionable to human sensory systems [1]. Human ears can tolerate the distortion in multimedia signals to a certain degree, unlike the case in data transmission where lossless delivery is absolutely required. Therefore, in error concealment, the principal aim is to reconstruct the distorted signal as close the original signal as possible. Various approaches have been proposed for multimedia EC [5][6]. In this paper, we present a novel error concealment technique for speech, which utilizes LSB audio watermarking scheme and evaluates its performance via experiments.

1.1. Speech Communication in General

In speech communication there are normally three parts involved at sender side and their inverse counter parts at receiver side. In the first two steps the signal is compressed by removing all redundancy from the stream and by discarding unimportant information in order to adapt the source signal rate to the available data rate. In the third step the data stream is prepared for the transmission over the network and therefore divided into network units and extended with redundancy information in order to make the signal robust to transmission errors [2].

1.2. Audio Compression

As seen in Figure 1.1 the speech signal is first transformed and down scaled by a waveform coder. This is a lossy compression and adapts the data rate with a reasonable measure to the available bandwidth of the channel. Lossy means that the original signal

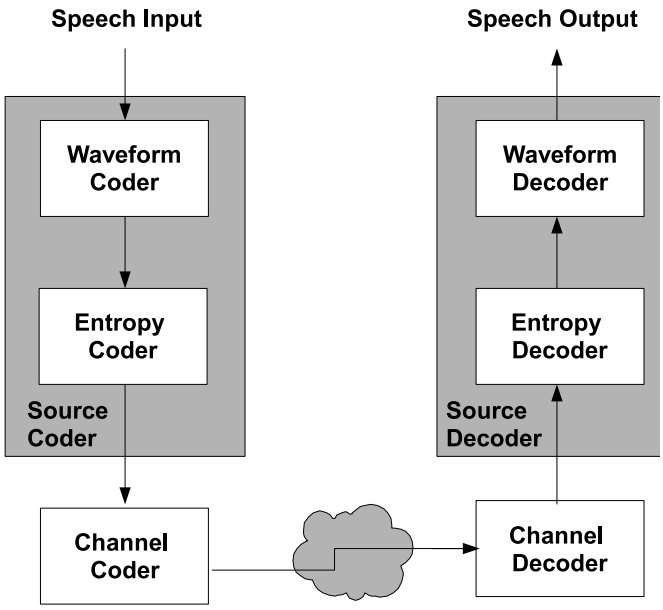


Figure 1.1. Speech communication

cannot be reconstructed with 100 percent fidelity at the receiver side. So we will always have a residual error, which can range from not noticeable over acceptable to annoying in dependence to the compression rate and the used algorithm.

After that the data is compressed further in a second compression step. Here a lossless algorithm (e.g. Huffmann Coding) is used in the so called entropy coder. *Lossless* means, that at the receiver side, the exact data can be reconstructed by the inverse decoding algorithm. Lossless compression schemes are the standard modes for compressing computer data like text or binary. They can work very efficiently with these types of data, because the distribution of the used values is uneven. ASCII files typically can be reduced up to 60 percent and more. But multimedia data is normally distributed in an even way and therefore lossy compression algorithms have to be used additionally, as described above. The combination of waveform and entropy coder is called source coder, as these two parts are mostly source signal dependent.

Speech transmissions are the most frequent used areas of multimedia applications. Because this data contains a lot of redundancy and because the available data rate is limited and expensive in most networks, powerful and strong compression mechanisms are used. Because speech has long stationary states, it is more efficient to code/transmit

only the changes, not always the whole speech sample. This introduces a pervasive structure into the bit stream and creates a high dependence within it. This brings an enormous reduction of needed bandwidth but also comes with the disadvantage, that transmission errors can have catastrophic effects. Not only the damaged sample is lost for the decoding process, but also the error will propagate into the future and make parts of the following data useless [2].

1.3. Communication Errors and Error Detection

When sending information over a network errors can occur. Depending on the infrastructure and the underlying technology these errors can range from bit errors to losses of complete network packets. Bit errors can range from bit deletions over bit insertions to bit inversions and are due to the imperfectness of physical connections and components. Packets can be lost because an intermediate network element is congested and has to discard new incoming data. When a lot of bits are lost (i.e. because a complete network packet does not reach the receiver) one speaks of a burst error.

In normal data communication error detection is done by header information and checksums, created by the transport layer or higher levels. More complicated Forward Error Correction Codes (FEC) (like the Reed-Solomon or Parity) codes are not only able to detect, but also to correct errors under the assumption that only a maximum amount of errors occurs within a single data unit.

When dealing with speech data, one can also exploit knowledge about the data structure itself to detect errors. As mentioned before the data is compressed further with lossless coding algorithms in the entropy coder. In these algorithms a certain set of codewords is used to represent the data. When the decoder now receives an invalid codeword, it will also know that something has gone wrong during the transmission and that there has been a transmission error [3].

1.4. Problem Statement

In this thesis, we present an error concealment method for speech over noisy wireless channels. Our primary concern is to design such a method that it should have the advantages of not consuming extra bandwidth and not introducing retransmission delay compared to other error control mechanisms.

Audio watermarking technique is employed to achieve these primary goals.

The audio watermarking algorithm used in this thesis is basically an LSB-based one and embeds the k-means clustering of the (n)th frame into (n-1)st frame before audio encoding. After the transmission, the erroneous or lost parts of (n)th frame is concealed by the clustering information transmitted on (n-1)st frame.

1.5. Reference Error Concealment Methods

Two error concealment methods which are utilising other techniques for concealment provides the similar advantages mentioned in the previous section, too. These methods are :

- FPR : Fading Pattern Repetition [7]
- ECFCP : Error Concealment Technique for VoIP Based on Forward Contour Prediction [8]

1.5.1. Fading Pattern Repetition (FPR)

The FPR strategy is based on the wellknown Pattern Repetition (PR) algorithm, used in speech, audio and audiovisual applications, where in case a packet is permanently lost, it is substituted by directly repeating a previous correctly received data segment. PR presents an attractive choice, because of its extremely low complexity and computational cost; however, the possible amplitude and phase mismatch between the audio stream and the segment to be repeated may cause audible clicking sounds,

leading in many cases to further deterioration of the overall audio quality. In order to avoid the audible effect of such discontinuities between the substitute packet limits and the correctly received packets, the FPR strategy employs time-domain window functions, as illustrated in Figure 1.3. The playout buffer for each reproduction device is monitored to detect a forthcoming gap in reproduction, while a number of successfully received packets are kept in a different buffer, to be used as possible data-frame substitutes.

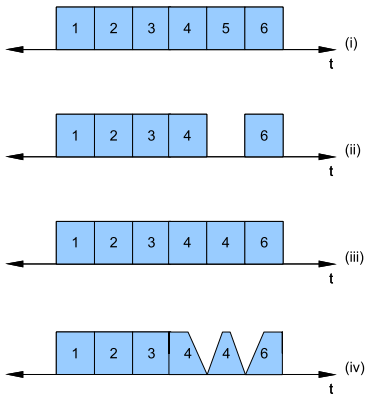


Figure 1.2. Error concealment using the FPR strategy : i.original sequence; ii.lost packet substituted with silence; iii.PR strategy; iv.FPR strategy

According to the FPR scheme, in case the receiver queue is empty while a predetermined number of samples remain to be reproduced, a linear descending gain function similar to a fade-out process is imposed. The substitute segment is added to the playout buffer in packet-by-packet basis, while an ascending gain function similar to a fadein process is imposed to the first packet. When a new packet is received, then the reverse process is followed: The linear descending function is imposed to the last samples of the substitute packet being reproduced, while the ascending function is used for the first samples of the received packet [7].

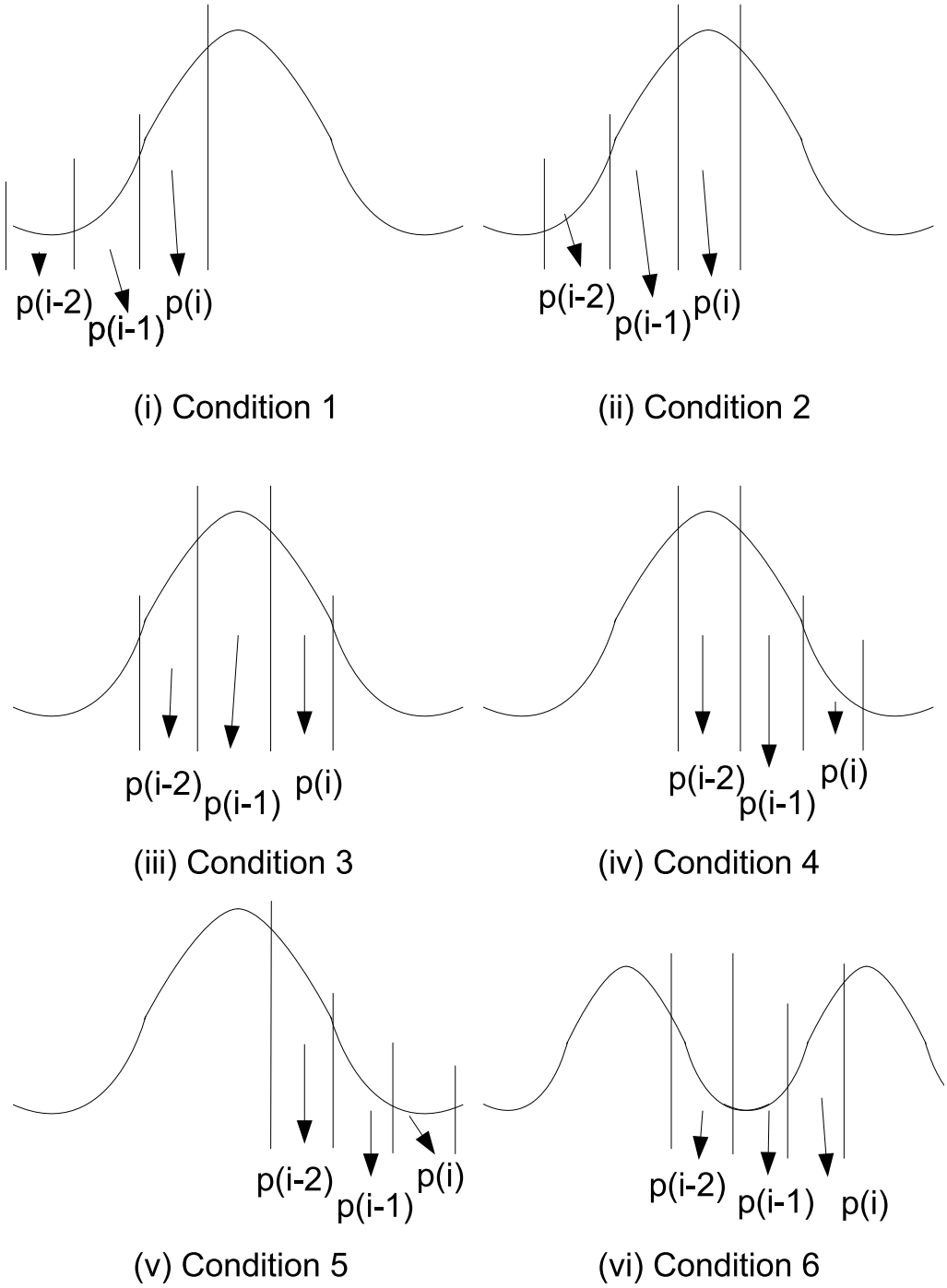


Figure 1.3. Error concealment using the ECFCP strategy. Six conditions for the current packet contour prediction based on the last two packets: i.the tendency should be going on increasing; ii.the tendency should be decreasing after being flat; iii.the tendency should be going on decreasing; iv.the tendency should be going on decreasing; v.the tendency should be going on decreasing until becoming silence. vi.the tendency should be going on increasing.

1.5.2. Error Concealment Technique for VoIP Based on Forward Contour Prediction (ECFCP)

In this algorithm, if the current packet is lost, the tendencies of the last two packets amplitudes are calculated. Then the current packet is recovered under the assumption below.

Assumption: speech is composed of many talkspurts. The waveform of a talkspurt generally has three stages. In the start stage, the amplitude is usually increasing. In the middle stage, the waveform stays relatively flat, i.e. its amplitude changes little. In the finish stage, the amplitude decreases gradually. Neighboring frames in a talkspurt should have a continuous contour tendency [8].

1.6. Outline of the Thesis

The second chapter provides some background information about "existing error resilience and error concealment techniques for the transmission of audio", "watermarking" , "audio watermarking" and "k-means clustering". In Chapter 3, the implemented error concealment technique is introduced and discussed. Simulation results are collected and interpreted in Chapter 4, while Chapter 5 and Chapter 6 concludes the thesis with some Future Work suggestions.

2. BACKGROUND

2.1. Existing Error Resilience and Error Concealment Techniques for the Transmission of Audio

The techniques described here can be divided into three classes: sender-based, receiver-based and sender/receiver-based mechanisms. These three sections are not mutual exclusive and can be used in combination with each other. Most of the sender-based techniques add redundancy overhead to the data and are called error resilience techniques, the receiver-based methods assume that there have been errors and try to correct or at least conceal them, they are called error concealment schemes, and the sender and receiver-based techniques presume a feedback channel from the decoder to the encoder and are based on the interaction of sender and the receiver, therefore called interactive techniques [4].

2.1.1. Sender-Based Techniques

Sender-based techniques normally work in the opposite way by adding redundancy (thus, creating overhead). This is because it is easier and sometimes even only possible to reconstruct a lost signal in the presence of some redundancy. As these methods try to make the data robust to errors in advance they are called error resilient or forward error correction techniques. These techniques can be listed as :

- Layered Encoding
- Multiple Description Coding
- Robust Waveform Coding
- Robust Entropy Coding
- Forward Error Correction (FEC)
- Joint Source and Channel Coding
- Interleaving
- Transport Level Control

2.1.2. Receiver Based Techniques

In contrast to the sender-based techniques the receiver-based methods do not increase the used bandwidth and can be used in combination with the sender-based. Some of these methods can be listed as :

- Insertion of Silence
- Insertion of White Noise
- Interpolation of Lost Data
- Pattern Repetition
- Fading Pattern Repetition (FPR) [7]
- Forward Contour Prediction (ECFCP) [8]

All of the methods described above, try to hide the effects of lost data by inserting harmless errors or by trying to reconstruct the lost data. As described before, the source decoder can also detect errors (in addition to the errors detected by the channel decoder).

In speech transmission the receiver-based techniques are only meaningful, when only small gaps exists and these are infrequent. As to the high error rate and the high possibility of burst errors in the Internet, these methods should only be used in combination with sender-based techniques and not separately.

2.1.3. Sender/Receiver Based Techniques

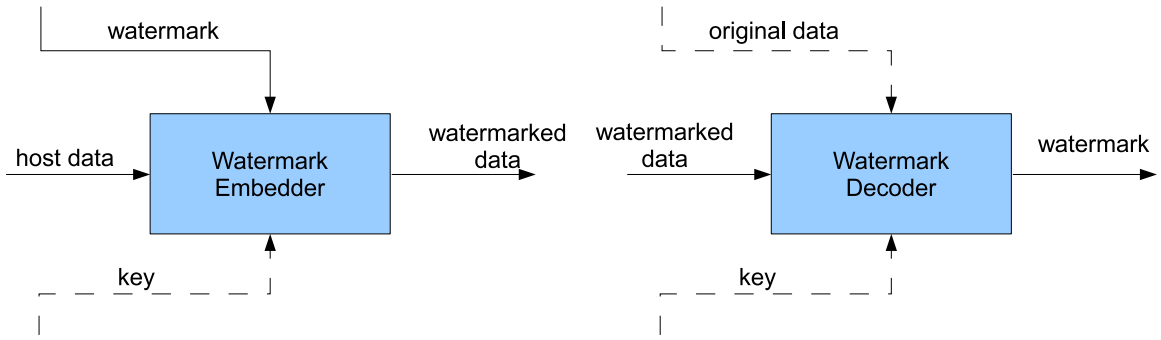
The sender-receiver-based techniques assume that there is a feedback channel between them and therefore the sender can adapt its coding scheme and parameters to the actual channel error characteristics. In many applications such a feedback channel is not available nor effective. This can be because of technical or time-critical reasons: while broadcasting or in multi-point transmission the receivers are possibly not known to the sender. In interactive or real-time communication a feedback channel could not be used because of the delay introduced by it. Some of these techniques can be listed

as :

- Adaptive Transport
- Retransmission without Waiting
- Priorized Multicopy Retransmission
- Error Concealment of MPEG-2 AAC Audio Using Modulo Watermarks(ECUMW): Cheng at al. proposed this error concealment schema for MPEG-2 compressed (AAC) audio using a novel modulo watermarking technique. It can be used on top of other error control schemes. After the modulo watermark is embedded, an MPEG-2 AAC audio only shows negligible file size increase and moderate SNR penalty. For audio transmission over packet-switched networks (e.g. Internet), using this concealment scheme shows consistent SNR gain over using conventional concealment schemes.

2.2. Watermarking

Digital watermarking is considered as an imperceptible, robust and secure communication of data related to the host signal, which includes embedding into and extraction from the host signal. The basic goal is that embedded watermark information follows the watermarked multimedia and endures unintentional modifications and intentional removal attempts. The principal design challenge is to embed watermark so that it is reliably detected in a watermark detector. The relative importance of the mentioned properties significantly depends on the application for which the algorithm is designed. For copy protection applications, the watermark must be recoverable even when the watermarked signal undergoes a considerable level of distortion, while for tamper assessment applications, the watermark must effectively characterize the modification that took place. In this section, several application areas for digital watermarking will be presented and advantages of digital watermarking over standard technologies will be examined [9].



(a) Watermark embedding (b) Watermark extraction

Figure 2.1. Watermark embedding and extraction

2.2.1. Ownership Protection

In the ownership protection applications, a watermark containing ownership information is embedded to the multimedia host signal. The watermark, known only to the copyright holder, is expected to be very robust and secure (i.e., to survive common signal processing modifications and intentional attacks), enabling the owner to demonstrate the presence of this watermark in case of dispute to demonstrate his ownership. Watermark detection must have a very small false alarm probability. On the other hand, ownership protection applications require a small embedding capacity of the system, because the number of bits that can be embedded and extracted with a small probability of error does not have to be large [9].

2.2.2. Proof of Ownership

It is even more demanding to use watermarks not only in the identification of the copyright ownership, but as an actual proof of ownership. The problem arises when adversary uses editing software to replace the original copyright notice with his own one and then claims to own the copyright himself. In the case of early watermark systems, the problem was that the watermark detector was readily available to adversaries. Anybody that can detect a watermark can probably remove it as well. Therefore, because an adversary can easily obtain a detector, he can remove owners watermark and replace it with his own. To achieve the level of the security necessary for proof the of ownership, it is indispensable to restrict the availability of the detector.

When an adversary does not have the detector, the removal of a watermark can be made extremely difficult. However, even if owners watermark cannot be removed, an adversary might try to undermine the owner. An adversary, using his own watermarking system, might be able to make it appear as if his watermark data was present in the owners original host signal. This problem can be solved using a slight alteration of the problem statement. Instead of a direct proof of ownership by embedding e.g. "Dave owns this image" watermark signature in the host image, algorithm will instead try to prove that the adversarys image is derived from the original watermarked image. Such an algorithm provides indirect evidence that it is more probable that the real owner owns the disputed image, because he is the one who has the version from which the other two were created [9].

2.2.3. Authentication and Tampering Detection

In the content authentication applications, a set of secondary data is embedded in the host multimedia signal and is later used to determine whether the host signal was tampered.

The robustness against removing the watermark or making it undetectable is not a concern as there is no such motivation from attackers point of view. However, forging a valid authentication watermark in an unauthorized or tampered host signal must be prevented. In practical applications it is also desirable to locate (in time or spatial dimension) and to discriminate the unintentional modifications (e.g. distortions incurred due to moderate MPEG compression) from content tampering itself. In general, the watermark embedding capacity has to be high to satisfy the need for more additional data than in ownership protection applications. The detection must be performed without the original host signal because either the original is unavailable or its integrity has yet to be established. This kind of watermark detection is usually called a blind detection [9].

2.2.4. Fingerprinting

Additional data embedded by watermark in the fingerprinting applications are used to trace the originator or recipients of a particular copy of multimedia file. For example, watermarks carrying different serial or identity (ID) numbers are embedded in different copies of music CDs or DVDs before distributing them to a large number of recipients. The algorithms implemented in fingerprinting applications must show high robustness against intentional attacks and signal processing modifications such as lossy compression or filtering. Fingerprinting also requires good anti-collusion properties of the algorithms, i.e. it is not possible to embed more than one ID number to the host multimedia file, otherwise the detector is not able to distinguish which copy is present. The embedding capacity required by fingerprinting applications is in the range of the capacity needed in copyright protection applications, with a few bits per second [9].

2.2.5. Broadcast Monitoring

A variety of applications for audio watermarking are in the field of broadcasting. Watermarking is an obvious alternative method of coding identification information for an active broadcast monitoring. It has the advantage of being embedded within the multimedia host signal itself rather than exploiting a particular segment of the broadcast signal. Thus, it is compatible with the already installed base of broadcast equipment, including digital and analogue communication channels [9].

The primary drawback is that embedding process is more complex than a simple placing data into file headers. There is also a concern, especially on the part of content creators, that the watermark would introduce distortions and degrade the visual or audio quality of multimedia. A number of broadcast monitoring watermark-based applications are already available on commercial basis. These include program type identification, advertising research, broadcast coverage research etc. Users are able to receive a detailed proof of the performance information that allows them to:

- Verify that the correct program and its associated promos aired as contracted;

- Track barter advertising within programming;
- Automatically track multimedia within programs using automated software online.

2.2.6. Copy Control and Access Control

In a copy control application, the embedded watermark represents a certain copy control or access control policy. A watermark detector is usually integrated in a recording or playback system. After a watermark has been detected and content decoded, the copy control or access control policy is enforced by directing particular hardware or software operations such as enabling or disabling the record module. These applications require watermarking algorithms resistant against intentional attacks and signal processing modifications, able to perform a blind watermark detection and capable of embedding a non-trivial number of bits in the host signal [9].

2.2.7. Information Carrier

The embedded watermark in this application is expected to have a high capacity and to be detected and decoded using a blind detection algorithm. While the robustness against intentional attack is not required, a certain degree of robustness against common processing like MPEG compression may be desired. A public watermark embedded into the host multimedia might be used as the link to external databases that contain certain additional information about the multimedia file itself, such as copyright information and licensing conditions [9].

2.3. Requirements of Digital Watermarking

Basic watermarking requirements, which are common to all digital media, are as follows:

2.3.1. Capacity

A watermark shall convey as much information as possible. Actually the amount of information that can be stored in a watermark depends on the application. For copy protection purposes, a payload of one bit is usually sufficient. For the protection of intellectually property rights, if someone wants to embed an amount of information similar to that used for ISBN (roughly 10 digits) and also if the year of copyright, the permission granted on the work and rating for it will be added, this means about 60 bits or 70 bits of information should be embedded in the host data [9].

2.3.2. Security

A watermark should in general be secret and should only be accessible by authorized parties. This security can be achieved by using cryptographic keys in watermark embedding and extracting. A watermarking technique is truly secure if knowing the exact algorithms for embedding and extracting the watermark does not help an unauthorised party to remove the watermark [9]. This is the main requirement for any cryptographic algorithm.

2.3.3. Robustness

A watermark must be robust against unintentional and intentional attacks. Attack denotes data manipulation with the purpose of impairing, destroying or removing the embedded watermarks. These attacks can be applying lossy compression techniques, filtering, resampling, requantization, digital-analog and analog-digital conversion. The watermark should stay within the host data unless the perceptual quality of the watermarked data decreases. If the watermark is removed the the watermarked data should be useless [9].

2.3.4. Imperceptibility

The watermarking algorithm must embed the watermark such that this does not affect the quality of the underlying host data. A watermark embedding procedure is truly imperceptible if humans can not distinguish the original data from the data with inserted watermark. On the other hand, for high robustness, it is desirable that the watermark amplitude is as high as possible. Thus, the design of a watermarking method always involves a trade-off between imperceptibility and robustness [9].

2.3.5. Public versus Private Watermarking

Private watermarking systems require the original unwatermarked data to find the watermark in the watermark extraction process. The watermark extraction algorithms in the public watermarking, also called blind watermarking, do not use the original unwatermarked data. This renders the watermarked extraction more difficult [9].

2.4. Audio Watermarking

Audio watermarks are special signals embedded into digital audio. These signals are extracted by detection mechanisms and decoded. Audio watermarking schemes rely on the imperfection of the human auditory system. However, human ear is much more sensitive than other sensory motors [10].

2.4.1. Overview of the Properties of the Human Auditory System (HAS

Watermarking of audio signals is more challenging compared to the watermarking of images or video sequences, due to wider dynamic range of the HAS in comparison with Human Visual System (HVS). The HAS perceives sounds over a range of power greater than 109:1 and a range of frequencies greater than 103:1. The sensitivity of the HAS to the Additive White Gaussian Noise (AWGN) is high as well; this noise in a sound file can be detected as low as 70 dB below ambient level.

On the other hand, in contrast to its large dynamic range, HAS contains a fairly small differential range, i.e. loud sounds generally tend to mask out weaker sounds. Additionally, HAS is insensitive to a constant relative phase shift in a stationary audio signal and some spectral distortions interprets as natural, perceptually non-annoying ones.

Auditory perception is based on the critical band analysis in the inner ear where a frequency-to-location transformation takes place along the basilar membrane. The power spectra of the received sounds are not represented on a linear frequency scale but on limited frequency bands called critical bands. The auditory system is usually modeled as a bandpass filterbank, consisting of strongly overlapping bandpass filters with bandwidths around 100 Hz for bands with a central frequency below 500 Hz and up to 5000 Hz for bands placed at high frequencies. If the highest frequency is limited to 24000 Hz, 26 critical bands have to be taken into account.

Two properties of the HAS dominantly used in watermarking algorithms are frequency (simultaneous) masking and temporal masking. The concept using the perceptual holes of the HAS is taken from wideband audio coding (e.g. MPEG compression 1-layer 3, usually called mp3). In the compression algorithms, the holes are used in order to decrease the amount of the bits needed to encode audio signal, without causing a perceptual distortion to the coded audio. On the other hand, in the information hiding scenarios, masking properties are used to embed additional bits into an existing bit stream, again without generating audible noise in the audio sequence used for data hiding [11].

Here are some audio watermarking techniques [11]:

- Least significant bit technique
- Phase changing technique
- Echo hiding technique
- DC-level shifting technique
- Direct sequence spread spectrum technique

- Frequency hopping spread spectrum technique
- Time hopping spread spectrum technique
- Cepstrum domain technique
- Frequency masking technique

2.5. K-Means Clustering

In this section, we briefly describe the k-means clustering. Clustering is finding groups in the data, where the groups are represented by their centers, which are the typical representatives of the group. Vector quantization is one application of clustering, but clustering is also used for preprocessing before a later stage of classification or regression.

Let us say we have an image that is stored with 24bit/pixel and can have up to 16 million colors. Assume we have a color screen with 8 bits/pixel that can display only 256 colors. We want to find the best 256 colors among all 16 million colors such that the image using only 256 colors in the palette looks as close as possible to the original image. This is color quantization where we map from higher to lower resolution. In the general case, the aim is to map from a continuous space to a discrete space; this process is called *vector quantization.*

Of course we can always quantize uniformly but this wastes the colormap by assigning entries to colors not existing in the image, or would not assign extra entries to colors frequently used in the image. For example if the image is a seascape, we expect to see many shades of blue and maybe no red. So the distribution of the colormap entries should reflect the original density as close as possible placing many entries in high-density regions, discarding regions where there is no data [12].

3. PROPOSED ERROR CONCEALMENT METHOD

The error concealment method proposed in this thesis creates a new communication channel between sender and receiver sides of the communication without consuming the existing bandwidth. Besides it uses this channel to carry some redundancy bits. This channel doesn't introduce any retransmission delay but introduces a constant delay of 5 ms. The proposed method employs LSB-based audio watermarking technique to create this hidden communication channel.

As the investigated audio watermarking techniques do not afford the adequate capacity for transmitting a highly compressed version of the audio signal, we try to carry some clustering information of the signal in order to refrain from distorting the original signal significantly.

Here is the overall system and how we use this capacity to conceal errors and the lost :

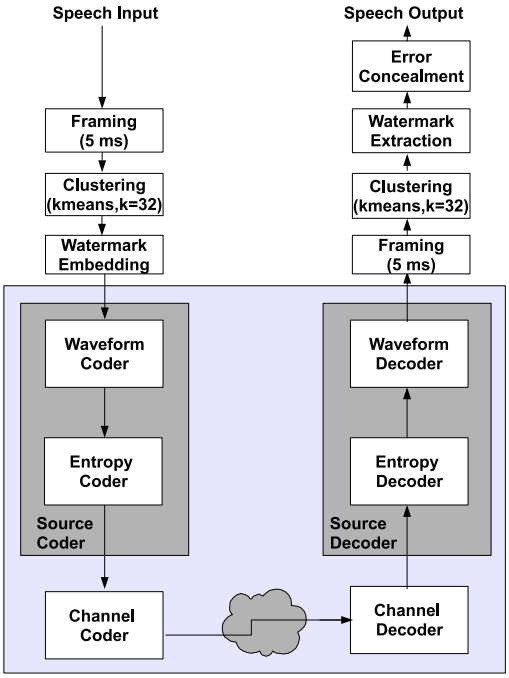


Figure 3.1. ECAW Schema

3.1. Sender Side

1. *A/D conversion* : This is the fundamental step for digital speech communications. The analogue signal is converted into digital format. In all our simulations, we use PCM encoded 8000 Hz, 8 bit mono speech signal samples.
2. *Framing* : At this step, speech signal is broken into frames of 5 ms.
3. *Clustering* : By using 5 bits of hidden information, k-means clustering with k=32 is executed.
4. *Embedding watermark* : In the end, the clustering information of (n)th frame calculated in previous step is embedded into (n-1)st frame.

Then the speech data are sent through the channel. The channel from this step to the first step of the receiver side is defined as *speech channel*. The important point, here, is that all of waveform coder, entropy coder, channel coder, physical channel itself, channel decoder, entropy decoder and waveform decoder are defined as speech channel in our system for the sake of simplicity. Our design is integrated before and after this speech channel which provides a pretty compatible system to existing speech channels. Then on the receiver side, the following procedure is performed:

3.2. Receiver Side

1. *Framing* : At this step, received speech signal is broken into frames of 5 ms.
2. *Clustering* : This step is same as the Step 3 of Sender Side. This time k-means clustering algorithm is executed by using the received speech signal.
3. *Error detection* : It is assumed that error detection is provided externally, in this step.
4. *Produce speech* : If any error is detected at Step 3, we replace the erroneous part with the corresponding part of the centre of assigned k-means cluster. This action provides the error concealment capability.
5. *A/D conversion* : This is a fundamental step for digital speech communications. The digital signal is converted to analogue format.

3.3. Integrated Audio Watermarking Techniques

3.3.1. LSB Coding Technique

One of the earliest techniques studied in the information hiding and watermarking area of digital audio is LSB coding. A natural approach in the case of the audio sequences is to embed watermark data by alternation of the individual samples of the digital audio stream having the amplitude resolution of 16 bits per sample. It usually does not use any psychoacoustics model to perceptually weight the noise introduced by LSB replacement.

The watermark encoder uses a subset of all available host audio samples chosen by a secret key. The substitution operation on the LSBs is performed on this subset. The extraction process simply retrieves the watermark by reading the value of these bits. Therefore, the decoder needs all the samples of the watermarked audio that were used during the embedding process.

The modification of the LSBs of the samples used for data hiding introduces a low power Additive White Gaussian Noise. HAS is very sensitive to the AWGN and this fact limits the number of LSBs that can be imperceptibly modified. The main advantage of the method is a very high watermark channel capacity; the use of only one LSB of the host audio sample gives capacity of 44.1 kbps. The obvious disadvantage is the extremely low robustness of the method, due to fact that random changes of the LSBs destroy the coded watermark. In addition, it is very unlikely that embedded watermark would survive digital to analogue and subsequent analogue to digital conversion.

Since no computationally demanding transformation of the host signal in the basic version of this method needs to be done, this algorithm has a very small algorithmic delay. This permits the use on this LSB in real-time applications. This algorithm is a good basis for steganographic applications for audio signals and a base for steganalysis [9].

3.3.2. Phase Changing Technique

One of the frequency domain techniques is the phase changing technique (PHASE). This technique is based on changing the phase of the certain number of samples in the audio signal [11].

The audio signal is broken into blocks of length N and each block is divided into sub-blocks of length N_q. The phase values of the samples in one of the sub-blocks are replaced with a new phase value, ϕ_w that is multiplied by the watermarking (WM) bit [13].

Algorithms that embed watermark into the phase of the host audio signal do not use masking properties of the HAS, but the fact that the HAS is insensitive to a constant relative phase shift in a stationary audio signal [14]. There are two main approaches used in the watermarking of the host signals phase; first, phase coding and, second, phase modulation.

The basic phase coding method was presented in [14]. The basic idea is to split the original audio stream into blocks and embed the whole watermark data sequence into the phase spectrum of the first block. One drawback of the phase coding method is a considerably low payload because only the first block is used for watermark embedding. In addition, the watermark is not dispersed over the entire data set available, but is implicitly localized and can thus be removed easily by the cropping attack. It is a non-blind watermarking method (as the phase modulation algorithm) that limits the number of applications it is suitable for.

The watermark insertion in the phase modulation method is performed using an independent multiband phase modulation [15, 16]. Imperceptible phase modifications are exploited in this approach by the controlled phase alternation of the host audio. To ensure perceptual transparency by introducing only small changes in the envelope,

the performed phase modulation has to satisfy the following constraint

$$|\Delta\phi(z)/\Delta z| < 30 \tag{3.1}$$

where $\phi(z)$ denotes the signal phase and z is the Bark scale. Each Bark constitutes one critical bandwidth; the conversion of frequency between Bark and Hz is given in [17]. Using a long block size N (e.g. $N = 2^{14}$) algorithm attains a slow phase change over time. The watermark is converted into a phase modulation by having one integer Bark scale carry one message bit of the watermark, with the frequency in Hz. The robustness of the modulated phase can be increased by using multiple Bark values carrying one watermark bit.

The watermark extraction requires a perfect synchronization procedure to perform a block alignment for each watermarked block, using the original signal as a reference. A matching of the particular segments of the modulated phase to the encoded watermark bits is possible if no significant distortions of the watermarked signal took place. The data rate of the watermark depends on three factors: first, the amount of the redundancy added, second, the frequency range used for watermark embedding, and, third, the energy distribution of the host audio. If the selected Barks energy is too low, that Bark should be skipped during the watermark embedding procedure. For audio signals sampled at 44.1 kHz, 0-15 kHz (0-24 in Bark scale) proved to be a sensible range for watermark embedding. If, for example, two Barks carry one watermark bit, the watermark data rate is $(24 = 2)(44100 = 2^{14}) = 32bps$.

3.3.3. DC-Level Shifting Technique

The DCSHIFT is proposed by Uludağ et al. [13] which is based on shifting the DC level of each blocks of the host audio signal to positive or negative levels to indicate the WM bits +1 or -1.

A secret key with 32 bits generate the binary watermark sequence associated with the copyright owner with that key. Linear Feedback Shift Registers (LFSR) are used

in this process. Tap weights are selected to generate a binary sequence with maximum period (m-sequence).

Then, for every frame with duration 25 ms. in this audio file, frame means and frame powers are calculated. Every individual frame is set to zero DC level by subtracting found frame means from audio samples in the associated frames.

This frame-wise DC zero audio sequence is processed to include watermark bits 0 and 1. Namely, the owner's watermark sequence is embedded as follows: If the bit to embed is a zero, the corresponding frame's DC level is shifted to a negative level with the value

$$level_0 = -DCBiasMultiplier * FramePower \tag{3.2}$$

If the bit to embed is a one, the corresponding frame's DC level is shifted to a positive level with the value

$$level_1 = +DCBiasMultiplier * FramePower \tag{3.3}$$

The level shifting is made proportional to the power of the frame for inaudibility purposes.

In watermark decoding, the frame means of every frame in the input watermarked audio sample is calculated. Same watermark sequence is generally written onto the input audio sample multiple times (epochs). Epoch number increases if the duration of input audio is increased. For each of these epochs, the binary watermark sequence is decoded according to the sign of the frame means. Namely, if a frame has positive frame mean, the associated bit is 1; if a frame has a negative frame mean, the associated bit is 0 [13].

3.3.4. Direct Sequence Spread Spectrum (DSSS) Technique

The most important part of the Direct Sequence Spread Spectrum (DSSS) technique is the pseudonoise (PN) sequence. The initial state of the PN sequence is used as a key. In DSSS and in other spread spectrum techniques, the key is needed to encode the information in the transmitter and the same key is needed to decode it at the receiver [11, 14].

For the watermark embedding process, the audio signal is broken into blocks of length N. The watermark bit for each block is multiplied by the PN sequence (pn). PN sequence has a flat frequency response over the frequency range, i.e., white noise. As a consequence, the spectrum of the watermark signal is spread over the available band. Then the spread data sequence is attenuated by multiplying by α and added to the host signal as an additive random noise. In the decoder part it is assumed that the initial state of the PN sequence is known. The watermarked signal is broken into blocks with length N. The watermark is decoded according to the sign of the correlation between the block samples and the PN sequence for each block [11].

3.3.5. Frequency Hopping Spread Spectrum (FHSS) Technique

This method is proposed by Cox et al. for image watermarking but they mentioned that it can be used for audio and video watermarking. They change only a selected set of DCT coefficients in a block. Also Barni et al. have proposed a DCT domain technique similar to Cox's algorithm. Frequency Hopping Spread Spectrum (FHSS) technique, uses PN sequence like DSSS technique but here the embedding is done in the frequency domain [11]. For the watermark embedding process, the audio signal is broken into blocks with length N. The DCT transform of each block is computed. In every block the watermark is embedded to only a selected set of DCT coefficients as described in [11].

In FHSS, to make the watermark detection, first the watermarked signal is broken into blocks with length N. The DCT of each block is computed. The watermark is

decoded according to the sign of the correlation between the DCT coefficients the selected components of each block and the PN sequence as described in [11].

3.3.6. Time Hopping Spread Spectrum Technique

Time Hopping Spread Spectrum technique (THSS) uses the PN sequence like the DSSS and the FHSS techniques. In the THSS technique, only a selected number of the components in each block are modified and the embedding is done in the time domain.

For the watermark embedding process, the audio signal is broken into blocks with length N. The total number of the selected coefficients is N_s. If the set of selected components in a block is called K then the embedding is done as described in [11].

In THSS technique, the watermark extraction method is similar to the one, which is done for the FHSS technique, but the difference is there is not a DCT transformation in extracting the watermark [11].

3.3.7. Cepstrum Domain Technique

Li and Yu have proposed an audio watermarking technique in the cepstrum domain. They have embedded the watermark by manipulating the mean of the cepstrum coefficients [11].

For the watermark embedding process, the audio signal is broken into blocks with length N. The cepstrum coefficients of each block are calculated. The mean of the real part of the cepstrum coefficients in each block is set to zero. The watermark embedding is done in such a way; if the watermark bit is +1 then a bias is added to the cepstrum coefficients and if the watermark bit is -1 then the cepstrum coefficients are not changed. The watermarked signal is found by calculating the inverse cepstrum of each block as described in [11].

In the watermark extraction part, first the watermarked signal is broken into

blocks with length N. The cepstrum coefficients of each block are calculated. Watermark is determined by the mean value of the real parts of the cepstrum coefficients of the related block. If the mean value is smaller than the half of the bias value then the extracted watermark bit is found as -1 and if mean is greater than the half of the bias value then the extracted watermark bit is found as +1 as described in [11].

3.3.8. Frequency Masking Technique

Swanson et al. have proposed a watermarking technique for audio which directly exploits temporal and frequency perceptual masking to guarantee that the embedded watermark is inaudible and robust. The watermark is constructed by breaking the host audio signal into blocks and adding a perceptually shaped PN sequence. For the frequency masking (FM) technique, the masking model defined in ISO-MPEG Audio Psychoacoustic Model for Layer 1 is used [11].

For the watermark embedding process, the host audio signal is broken into blocks with a length of 512 samples. The power spectrum is calculated and by using MPEG Layer 1 model, minimum masking threshold, MMT is found for each block. The Fast Fourier Transform (FFT) of the PN sequence is calculated. The FFT of the PN sequence is weighted by MMT in the frequency domain for inaudibility purposes. In fact frequency shaping is not enough to guarantee that the embedded watermark will be inaudible. A watermark computed using frequency domain masking will be spread in time over the entire analysis block. If the energy is concentrated in a time interval that is shorter than the analysis block length, the watermark is not masked outside of that subinterval. This leads to audible distortion. A temporal mask is used which guaranties that the quite regions, which have low amplitude are not disturbed by the watermark signal. Here the temporal masking effects are approximated by using the envelope of the host audio signal for each block. The estimated envelope increases with the signal and decays as $e^{-\alpha}$. The watermark bit is multiplied by the shaped PN sequence, *pn* and attenuated by multiplying by α and then shaped by the temporal mask, *temp* in the time domain and added to the host signal to get the watermarked signal as described in [11].

In the FM technique, the watermark extraction method is similar to the one, which is done for the DSSS technique as described in [11].

3.4. Integrated Audio Encoding Techniques

3.4.1. PCM

Pulse Code Modulation (PCM) codecs are the simplest form of waveform codecs. Narrowband speech is typically sampled 8000 times per second, and then each speech sample must be quantized. If linear quantization is used then about 12 bits per sample are needed, giving a bit rate of about 96 kbits/s. However this can be easily reduced by using non-linear quantization. For coding speech it was found that with non-linear quantization 8 bits per sample was sufficient for speech quality which is almost indistinguishable from the original. This gives a bit rate of 64 kbits/s, and two such non-linear PCM codecs were standardised in the 1960s. In America u-law coding is the standard, while in Europe the slightly different A-law compression is used. Because of their simplicity, excellent quality and low delay both these codecs are still widely used today. For example the .au audio files that are often used to convey sounds over the Web are in fact just PCM files [20].

3.4.2. ADPCM

Adaptive Differential Pulse Code Modulation (ADPCM) codecs are waveform codecs which instead of quantizing the speech signal directly, like PCM codecs, quantize the difference between the speech signal and a prediction that has been made of the speech signal. If the prediction is accurate then the difference between the real and predicted speech samples will have a lower variance than the real speech samples, and will be accurately quantized with fewer bits than would be needed to quantize the original speech samples. At the decoder the quantized difference signal is added to the predicted signal to give the reconstructed speech signal. The performance of the codec is aided by using adaptive prediction and quantization, so that the predictor and difference quantizer adapt to the changing characteristics of the speech being coded

[?].

3.4.3. GSM 6.10

The "Global System for Mobile communications" (GSM) is a digital mobile radio system which is extensively used throughout Europe, and also in many other parts of the world.

The GSM full rate speech codec operates at 13 kbits/s and uses a Regular Pulse Excited (RPE) codec. For details on RPE codecs see the section earlier on hybrid codecs. Basically the input speech is split up into frames 20 ms long, and for each frame a set of 8 short term predictor coefficients are found. Each frame is then further split into four 5 ms sub-frames, and for each sub-frame the encoder finds a delay and a gain for the codec's long term predictor. Finally the residual signal after both short and long term filtering is quantized for each sub-frame as follows. The 40 sample residual signal is decimated into three possible excitation sequences, each 13 samples long. The sequence with the highest energy is chosen as the best representation of the excitation sequence, and each pulse in the sequence has its amplitude quantized with three bits. At the decoder the reconstructed excitation signal is fed through the long term and then the short term synthesis filters to give the reconstructed speech. A postfilter is used to improve the perceptual quality of this reconstructed speech [?].

3.4.4. DSP Group TrueSpeech

TrueSpeech is a family of speech compression and decompression algorithms and software. It is designed for personal computers and personal communications devices. With the high compression ratios ranging from 15:1 to 27:1, TrueSpeech improves the storage and communications transmission of digital voice information and can be used in the integration of personal computers and telephones. TrueSpeech can be utilized in many products and applications such as [21]:

- Multimedia PCs

- Sound cards and modems
- Computer/telephony and teleconferencing
- Voice mail systems and PBX systems
- Wireless/cellular applications
- Personal digital assistants
- Games, education
- Video/cable and on-line services

Truespeech is a proprietary audio codec produced by the DSP Group. It is designed for encoding voice data at low bit rates, and to be embedded into DSP chips. True speech has been integrated into Windows Media Player, and is the format used by the voice chat features of Yahoo Messenger [22].

3.4.5. MPEG-1 Audio Layer 3 (MP3)

MPEG-1 Audio Layer 3, more commonly referred to as MP3, is a digital audio encoding format.

This encoding format is used to create an MP3 file, a way to store a single segment of audio, commonly a song, so that it can be organized or easily transferred between computers and other devices such as MP3 players.

MP3 uses a lossy compression algorithm that is designed to greatly reduce the amount of data required to represent the audio recording, yet still sound like a faithful reproduction of the original uncompressed audio to most listeners. An MP3 digital file created using the mid-range bit rate setting of 128 kbit/s results in a file that is typically about 1/10th the size of the digital data found on an audio CD.

MP3 is an audio-specific format. It was invented by a team of international engineers at Philips, IRT, AT&T-Bell Labs and Fraunhofer Society, and it became an ISO/IEC standard in 1991. The compression works by reducing accuracy of certain parts of sound that are deemed beyond the auditory resolution ability of most people.

This method is commonly referred to as Perceptual Coding.

It provides a representation of sound within the frequency domain, by using psychoacoustic models to discard or reduce precision of components less audible to human hearing, and recording the remaining information in an efficient manner. This is relatively similar to the principles used by some other lossy compression algorithms such as JPEG image compression format [23].

3.5. Simulated Channel Model

3.5.1. Gilbert-Elliot Channel Model

When data packets are transferred over channels with bursty errors, packet error statistics are more important than bit error statistics to analyze the communication performance. Error modeling for such channels can be done using the Gilbert-Elliot model.

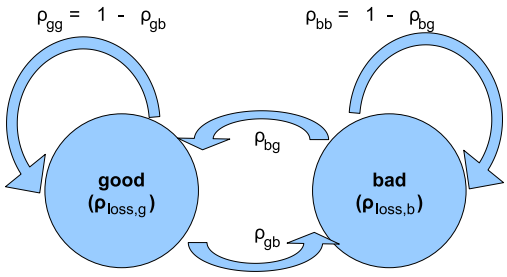


Figure 3.2. Gilbert channel model. Subscript b is for bad state, whereas g stands for good state and p_{ij} is the transition probability from state i to j where i,j $\in$ {b,g}

Error-prone channel in this thesis has been modeled as a two-state discrete-time Markov process in accordance with Gilbert model illustrated in fig:gilbert. The state, when channel is in deep fade, is called bad state. Packets belonging to a speech are fully corrupted in the bad channel state ($P_{loss,b} = 1$) while packet losses in the good channel states are negligibly small, i.e., packet loss of 10^{-4}) ($P_{loss,g} \approx 0$). Accordingly,

$$P = \begin{pmatrix} P_{gg} & P_{gb} \\ P_{bg} & P_{bb} \end{pmatrix} \tag{3.4}$$

is the transition matrix for the packet error process. Therefore, the probability of having packet errors is given by

$$\rho_{loss} = \frac{P_{loss,b}.P_{gb} + P_{loss,g}.P_{bg}}{P_{bg} + P_{gb}} \tag{3.5}$$

We assume that the packet losses are in the consecutive orders based on the given transition probabilities. The higher-layer oriented modeling methods, such as the elucidated one, directly provide packet error probabilities without the need to understand the complex physical-layer schemas and long conditional probability calculations. Even though this simplified model does not capture all the details of fading, it does provide a robust model for wireless channels with bursty-error characterics. In the simulations, the channel starts at the bad state, and transition state probabilities are adjusted in a way that desired packet losses are achieved. We note that due to the fading dynamics of an error-prone channel and/or the bottleneck at the transmitters, this scenario maybe highly expected in particular applications [24].

4. EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS

4.1. Interaction between Audio Encoding Types and Audio Watermark Techniques

Audio watermarking techniques explained in Section 3.1 and audio encoding methods mentioned in Section 3.2 were integrated into the simulation system using various combinations. Among the watermarking techniques, LSB Coding turned out to be the only one to supply the enough capacity to carry the required clustering information, while PCM turned out to be the only one to preserve the embedded watermarking signal during its encoding/decoding operations among encoding types. Therefore, in the rest of the simulation, LSB Coding watermark - PCM encoding type combination was used to prove the efficacy of the method. Discovering alternative combinations, which may necessitate investigation of other audio encoding types and watermarking techniques or modification of those mentioned in this thesis, could constitute another research subject for future work.

4.2. Simulation Environment

Simulation results are generated using the following environment.

4.2.1. Development Environment

Matlab R13 was used .

4.2.2. Audio Watermarking Method

LSB Coding technique was employed to bind the clustering information into host signal. On each 5 ms audio signal frame, 5 bits clustering information is embedded.

This introduces a hidden auxiliary channel that has the capacity of 1 kbits/sec for transmitting the clustering information.

4.2.3. Channel Model

Gilbert-Elliot channel model was used to introduce the noisy environment. Channel parameters were set in a way that percentage of audio signal frames, lost during transmission, spanned from 1 per cent to 8 per cent under good channel conditions and from 10 per cent to 80 per cent under bad channel conditions. In three packet sizes experiments were carried out for lost packets to prove the efficacy of the presented method and reference methods against losses of small, middle and big packets. Packet sizes were 16 bits, 32 bits, 64 bits and 128 bits, respectively.

4.2.4. Audio File Encoding

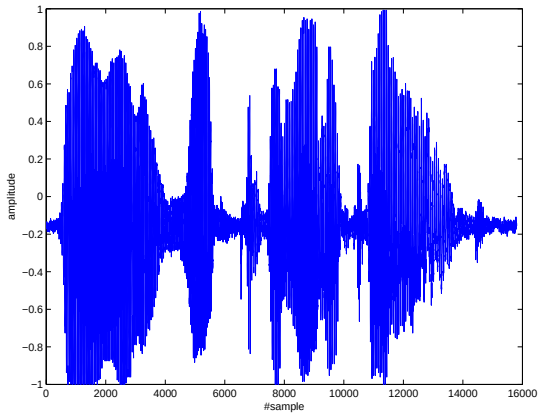
Pulse Code Modulation (PCM) 8 bit, mono, 8000 hz encoded audio signal was used in simulations.

4.2.5. Sample Clips

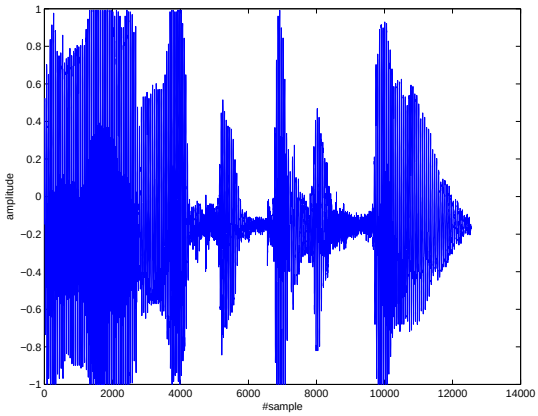
Some suitable English-language phrases as suggested by ITU-T recommendation P.800 were used in simulations :

- Clip 1 : You will have to be very quiet. (15.807 sample, ≈1.98 sec)
- Clip 2 : There was nothing to be seen. (12.539 sample, ≈1.56 sec)
- Clip 3 : They worshipped wooden idols. (14.127 sample, ≈1.77 sec)
- Clip 4 : I want a minute with the inspector. . (15.906, ≈1.99 sec)
- Clip 5 : Did he need any money? (12.937 sample, ≈1.62 sec)
- Clip 6 : All five clips (109.760 sample, ≈13.72 sec)

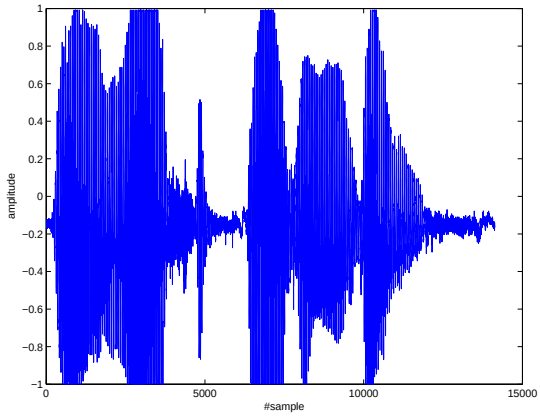
fig:clips shows waveforms of each clip.



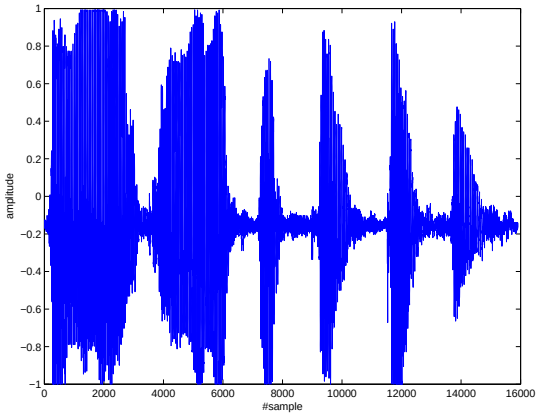
(a) Clip-1



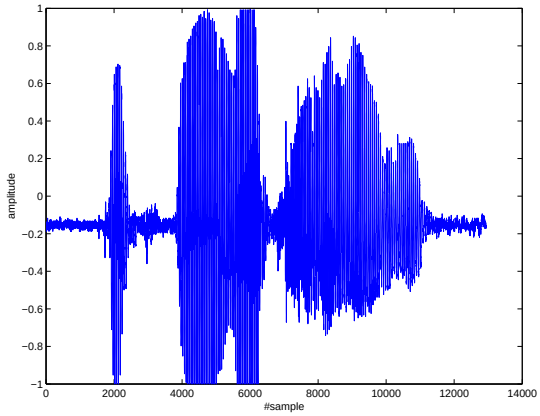
(b) Clip-2



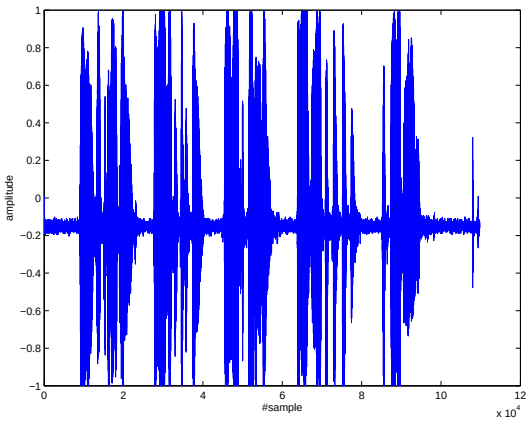
(c) Clip-3



(d) Clip-4

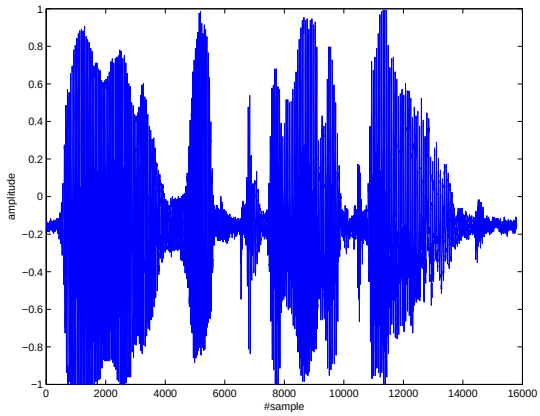


(e) Clip-5

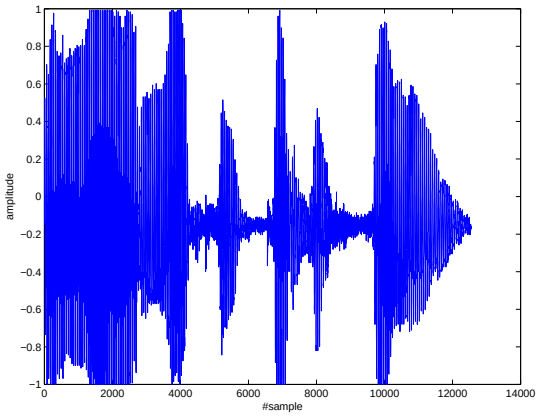


(f) Clip-6

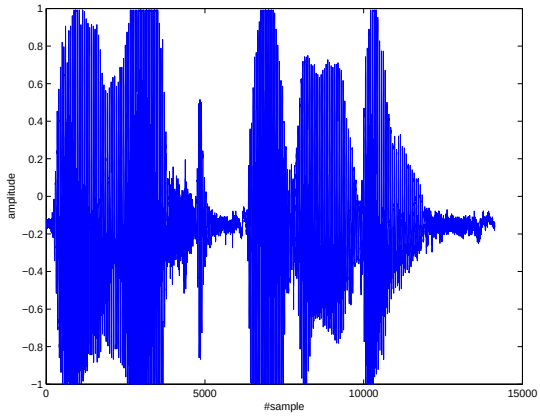
Figure 4.1. Sample clips used in the simulations.



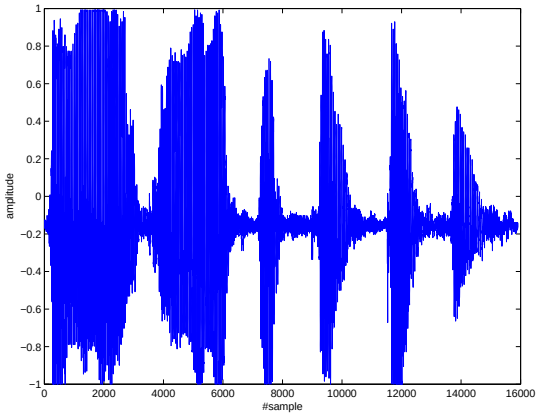
(a) Clip-1



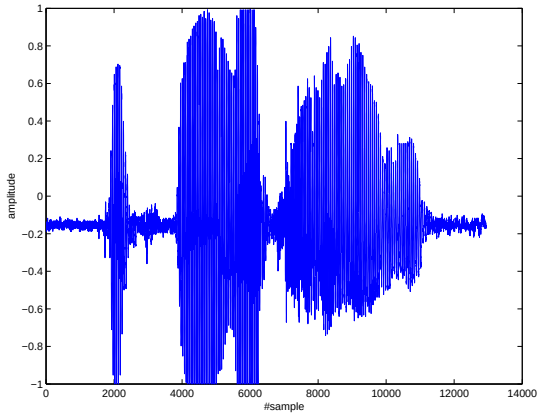
(b) Clip-2



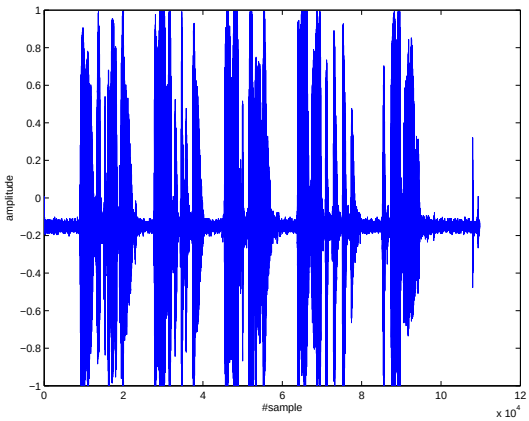
(c) Clip-3



(d) Clip-4

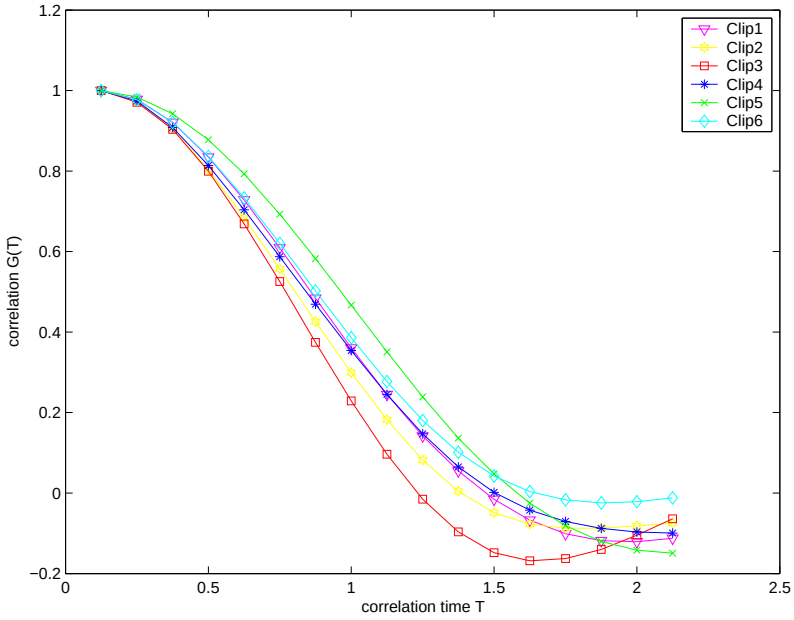


(e) Clip-5



(f) Clip-6

Figure 4.2. Sample clips used in the simulations.



(a)

Figure 4.3. Auto-correlation sequence of each clip

4.2.6. Reference Methods for Comparison

Performance of the proposed method was compared to the following novel error concealment methods :

- FPR : Fading Pattern Repetition [7]
- ECFCP : Error Concealment Technique for VoIP Based on Forward Contour Prediction [8]

In order to compare the performance of the proposed method to the reference methods, reference methods were implemented and comparision results were reported using the SNR values generated by each method.

4.3. Simulation Results

4.3.1. Audio Quality after Watermark Embedding

Deterioration in the quality of speech signal to a small degree is stipulated following the embedment of LSB-based audio watermark into speech signal. However, our test shows that the perceptual quality of the watermarked audio clips is acceptable. As an objective measure, we report the SNR values of each clip after watermark embedding.

Table 4.1. SNR values (in dB) after watermark embedding

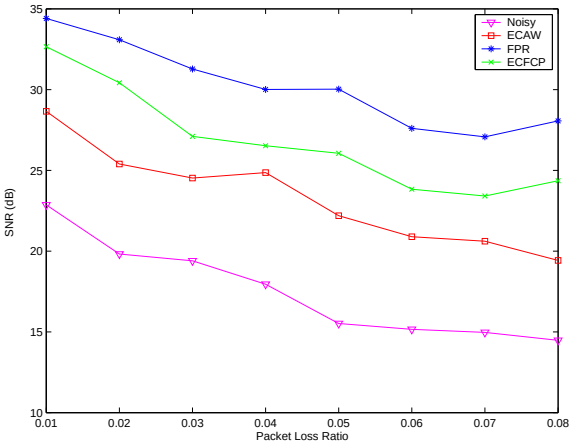
Clip1	Clip2	Clip3	Clip4	Clip5	Clip6
37.49	38.07	38.38	37.59	37.16	36.25

4.3.2. SNR (Signal-to-Noise Ratio) Evaluation

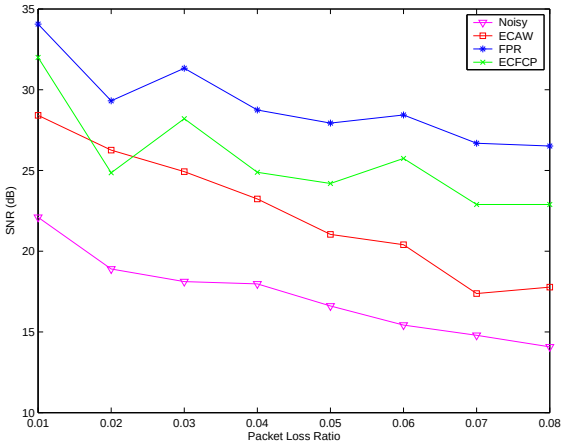
As shown in Table 4.2 - 4.5, Figure 4.4 and Figure 4.11, ECAW offers better SNR performance than FPR and ECFCP method when the loss packet size is 64 bit or greater. However, FPR and ECFCP offer better SNR performance for smaller loss packet size as stipulated. Auto correlation graphs of each clip given in section entitled "Sample Clips" provide an explanation for this. Auto correlation is stronger while the time lag between the packets is smaller. On the other hand auto correlation decreases while the lag increases. Since their performance highly depends on the likeness of adjacent packet to the loss packet, FPR and ECPCPs performance is affected negatively when the loss packet size is greater. However, ECAW does not have such a dependency. It uses the clustering information, carried by watermarking signal, to recover the loss frame.

Table 4.2. SNR comparison (in dB) for 16 bits packet loss

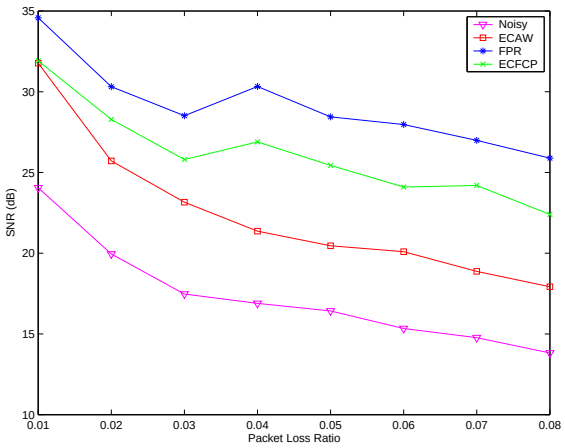
		Packet Loss Ratio				
		0.01	0.02	0.05	0.10	0.20
Clip1	ECAW	28.65	25.40	22.19	18.88	14.41
	FPR	34.41	33.09	30.03	26.00	23.18
	ECFCP	32.65	30.43	26.06	22.42	19.02
Clip2	ECAW	28.41	25.27	21.04	18.11	13.79
	FPR	34.06	29.31	27.93	24.87	22.35
	ECFCP	32.00	24.85	24.19	21.80	18.39
Clip3	ECAW	31.77	25.71	20.46	17.91	13.06
	FPR	34.58	30.31	28.44	24.86	21.94
	ECFCP	31.91	28.28	25.44	22.20	18.37
Clip4	ECAW	27.31	25.36	21.02	17.65	13.65
	FPR	34.55	31.79	29.84	25.74	22.69
	ECFCP	31.88	30.18	25.92	22.40	18.88
Clip5	ECAW	25.62	25.65	18.15	15.12	13.09
	FPR	35.46	32.73	29.95	27.00	24.50
	ECFCP	33.65	30.96	26.67	23.70	20.14
Clip6	ECAW	24.19	21.88	19.53	14.91	12.34
	FPR	33.59	31.39	29.00	26.46	23.11
	ECFCP	31.78	28.87	25.66	22.76	19.24



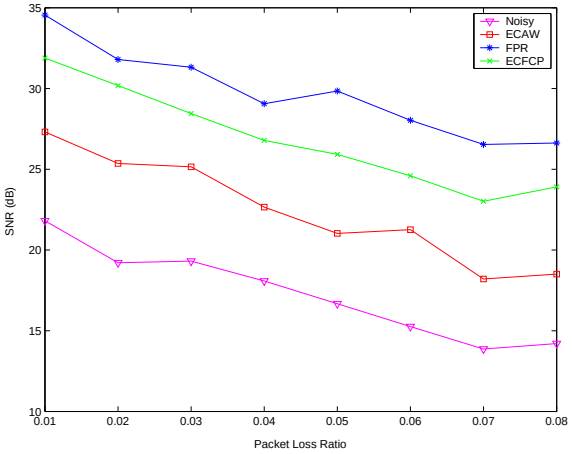
(a) SNR values of Clip-1



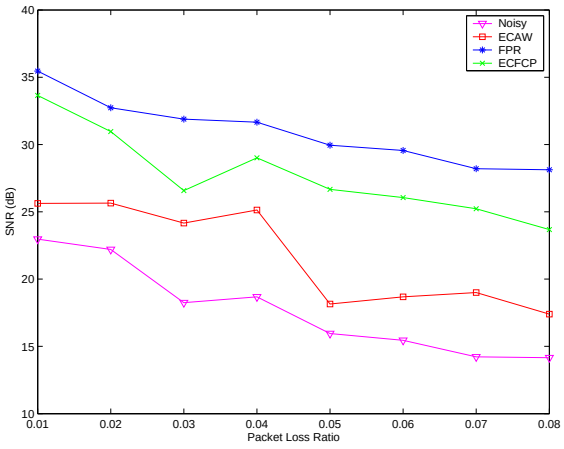
(b) SNR values of Clip-2



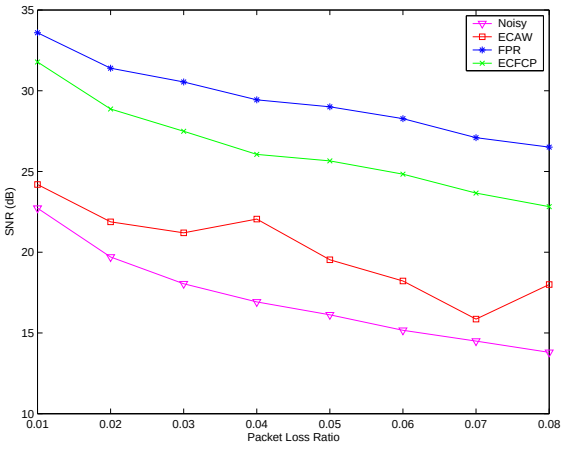
(c) SNR values of Clip-3



(d) SNR values of Clip-4

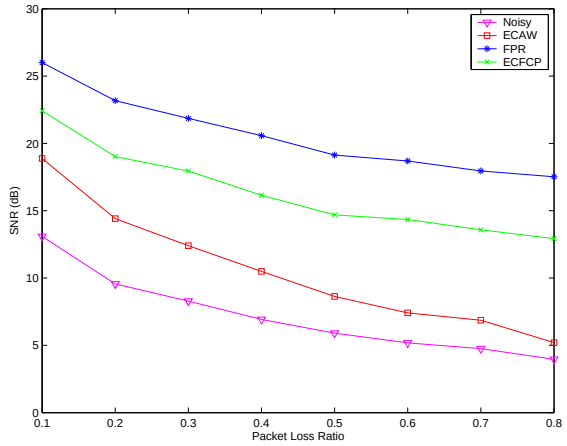


(e) SNR values of Clip-5

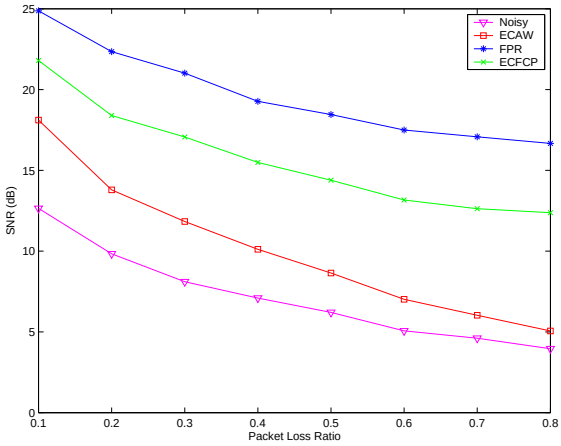


(f) SNR values of Clip-6

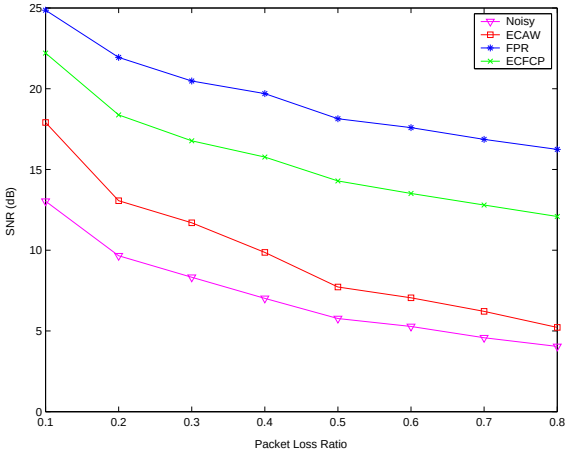
Figure 4.4. Performance results for various clips at good channel conditions for 16 bits packet loss.



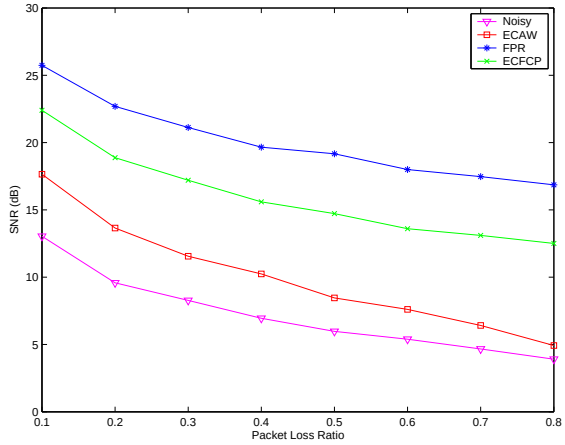
(a) SNR values of Clip-1



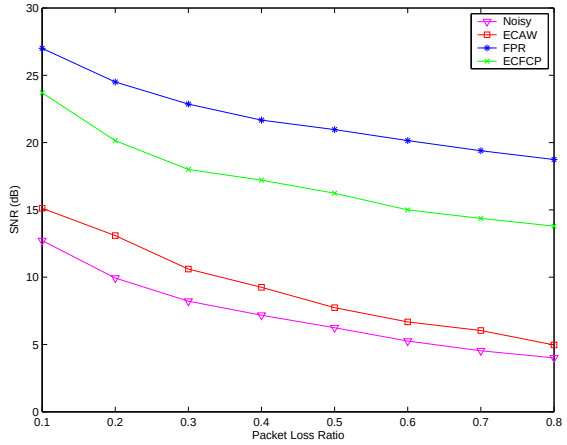
(b) SNR values of Clip-2



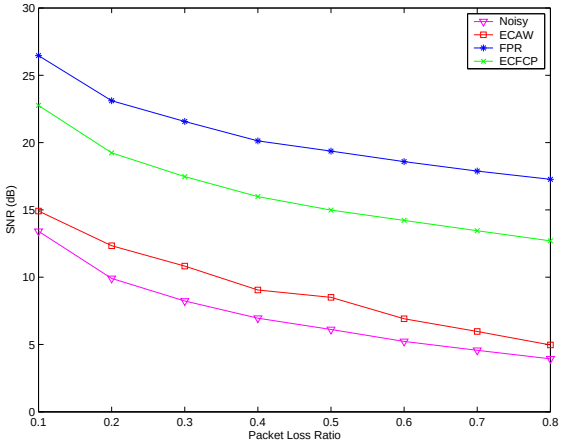
(c) SNR values of Clip-3



(d) SNR values of Clip-4



(e) SNR values of Clip-5

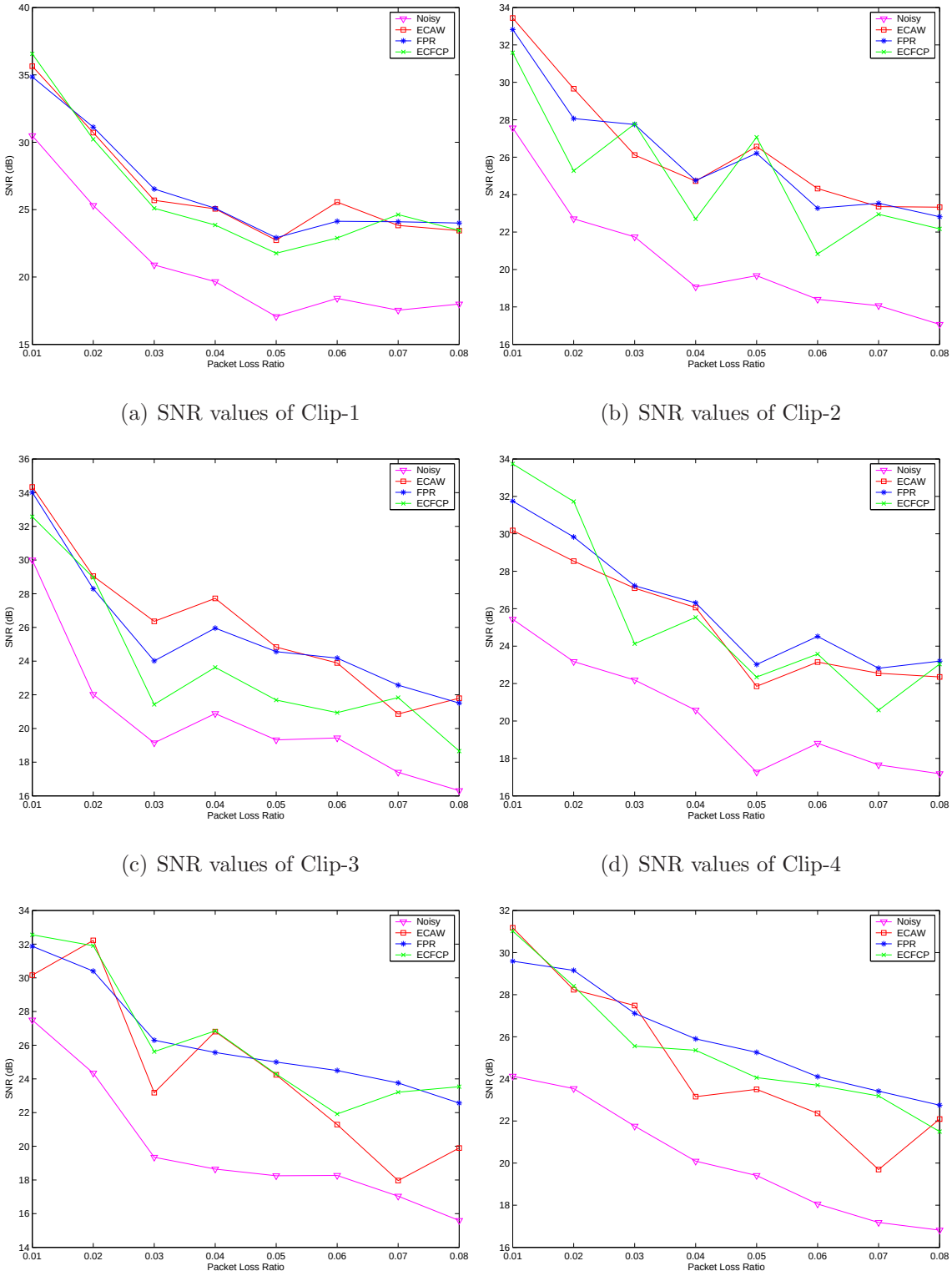


(f) SNR values of Clip-6

Figure 4.5. Performance results for various clips at bad channel conditions for 16 bits packet loss.

Table 4.3. SNR comparison (in dB) for 32 bits packet loss

		Packet Loss Ratio				
		0.01	0.02	0.05	0.10	0.20
Clip1	ECAW	35.64	30.71	22.73	24.79	21.00
	FPR	34.85	31.12	22.91	16.63	15.58
	ECFCP	36.56	30.22	21.76	16.63	15.58
Clip2	ECAW	33.43	29.66	26.57	22.49	17.27
	FPR	32.82	28.06	26.21	21.98	19.20
	ECFCP	31.58	25.28	27.07	18.34	17.51
Clip3	ECAW	34.34	29.04	24.83	21.37	17.78
	FPR	34.02	28.29	24.56	21.69	19.19
	ECFCP	32.57	28.96	21.69	21.20	17.13
Clip4	ECAW	30.18	28.54	21.85	20.32	17.11
	FPR	31.75	29.83	23.02	20.86	19.03
	ECFCP	33.73	31.73	22.35	20.55	18.66
Clip5	ECAW	30.16	32.22	24.24	16.37	14.52
	FPR	31.87	30.40	25.00	22.16	20.08
	ECFCP	32.55	31.91	24.29	21.52	20.70
Clip6	ECAW	31.18	28.23	23.50	18.53	15.44
	FPR	29.59	29.15	25.26	21.80	19.23
	ECFCP	31.02	28.41	24.06	21.06	17.91



(a) SNR values of Clip-1

(b) SNR values of Clip-2

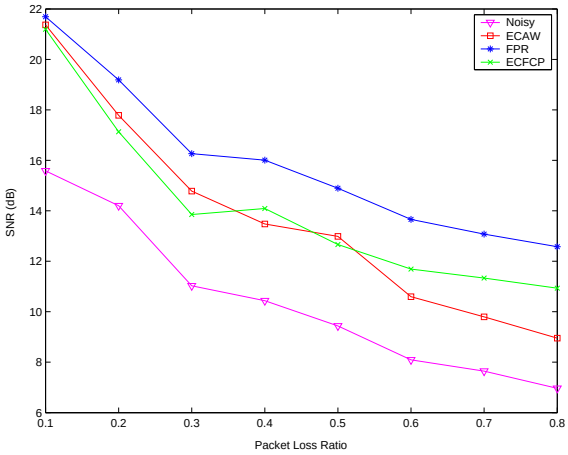
(c) SNR values of Clip-3

(d) SNR values of Clip-4

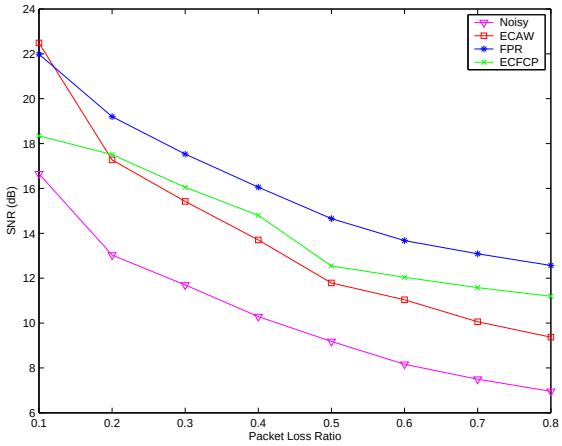
(e) SNR values of Clip-5

(f) SNR values of Clip-6

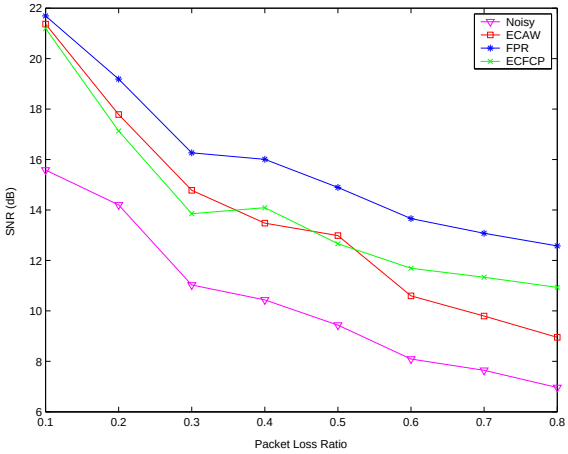
Figure 4.6. Performance results for various clips at good channel conditions for 32 bits packet loss.



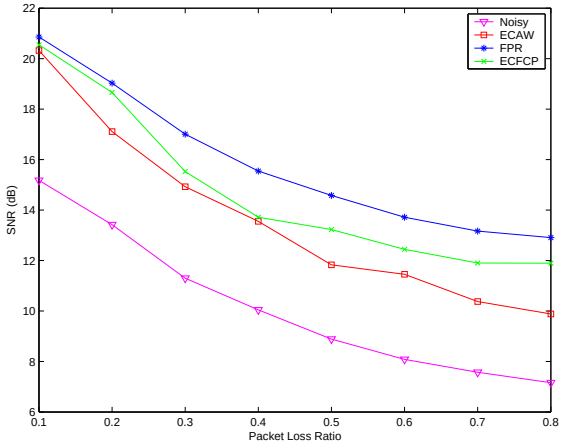
(a) SNR values of Clip-1



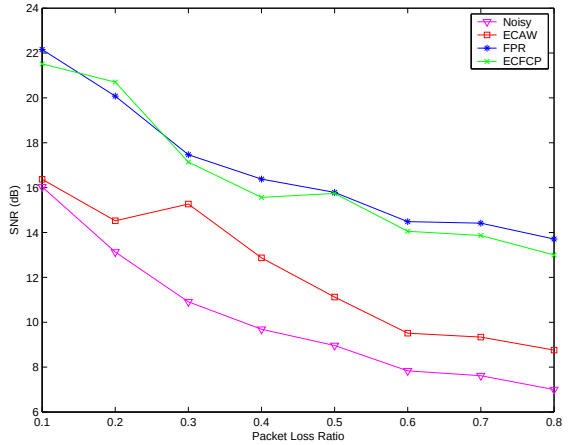
(b) SNR values of Clip-2



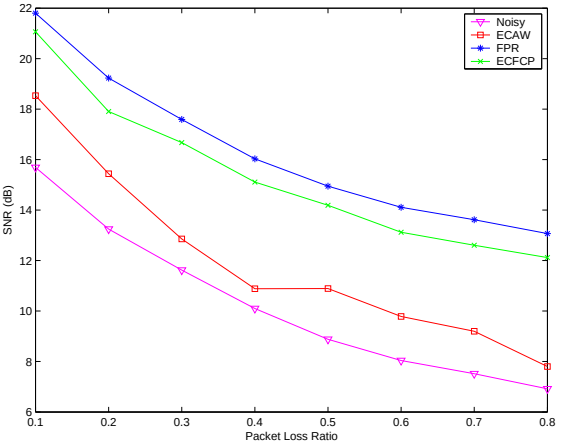
(c) SNR values of Clip-3



(d) SNR values of Clip-4



(e) SNR values of Clip-5

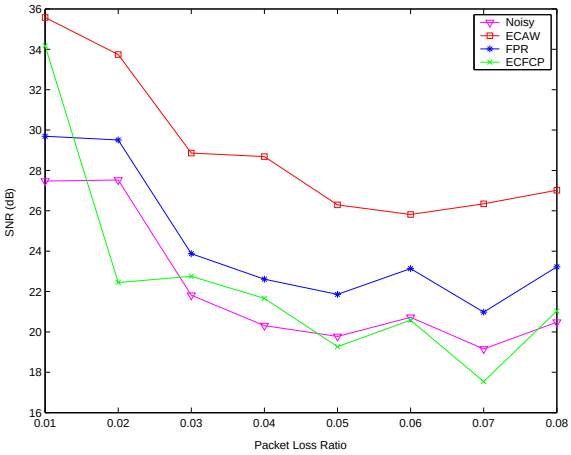


(f) SNR values of Clip-6

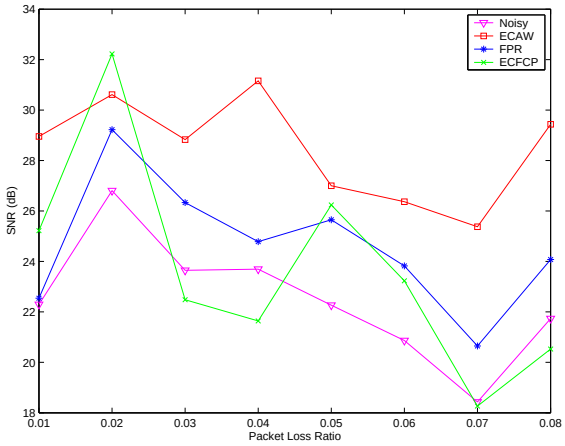
Figure 4.7. Performance results for various clips at bad channel conditions for 32 bits packet loss.

Table 4.4. SNR comparison (in dB) for 64 bits packet loss

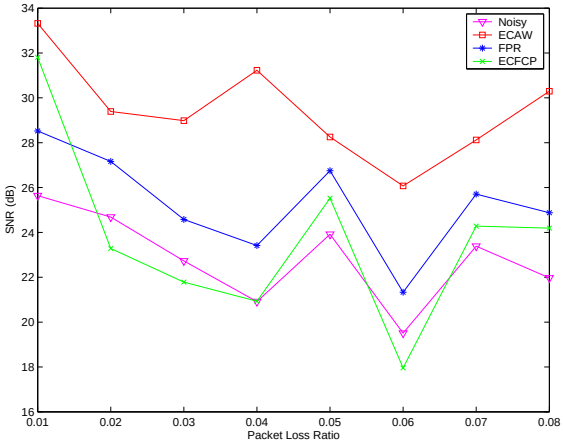
		Packet Loss Ratio				
		0.01	0.02	0.05	0.10	0.20
Clip1	ECAW	35.58	33.74	26.30	24.82	22.11
	FPR	29.69	29.51	21.86	18.90	18.73
	ECFCP	34.18	22.44	19.27	17.04	15.73
Clip2	ECAW	28.95	30.61	27.00	20.86	21.19
	FPR	22.54	29.22	25.65	16.77	18.11
	ECFCP	25.22	32.22	26.24	14.37	18.74
Clip3	ECAW	33.32	29.40	28.25	24.14	18.40
	FPR	28.52	27.16	26.75	20.87	17.62
	ECFCP	31.80	23.28	25.52	18.83	13.93
Clip4	ECAW	33.89	30.75	24.42	22.23	20.95
	FPR	33.59	24.95	22.01	21.33	18.99
	ECFCP	27.55	26.75	23.29	17.48	19.25
Clip5	ECAW	35.58	35.37	26.57	19.63	19.76
	FPR	28.54	30.40	25.35	19.64	17.82
	ECFCP	30.44	28.26	22.39	18.14	14.26
Clip6	ECAW	32.05	31.01	26.11	24.53	18.63
	FPR	28.83	27.62	25.80	20.52	17.95
	ECFCP	26.44	25.84	22.08	19.27	15.75



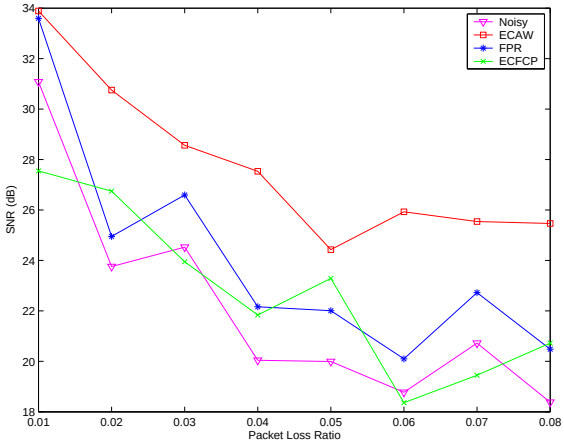
(a) SNR values of Clip-1



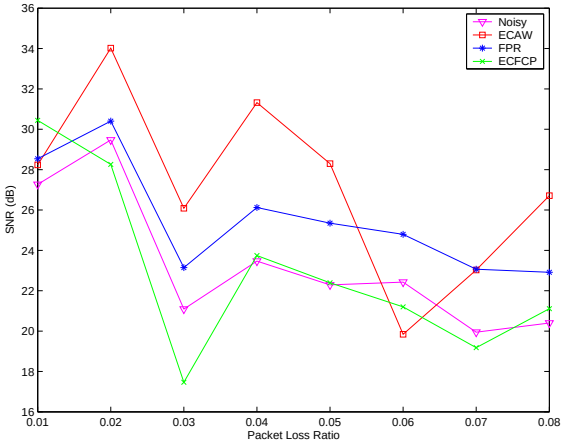
(b) SNR values of Clip-2



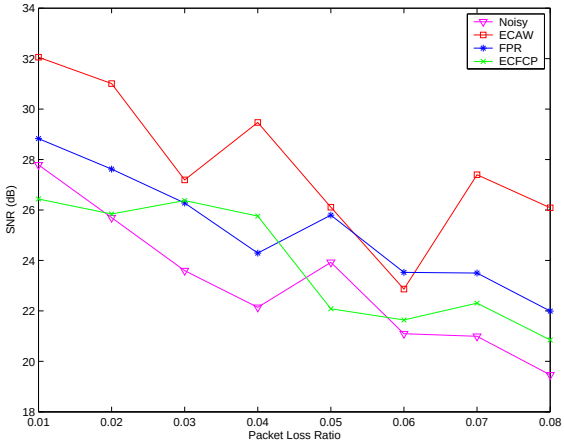
(c) SNR values of Clip-3



(d) SNR values of Clip-4

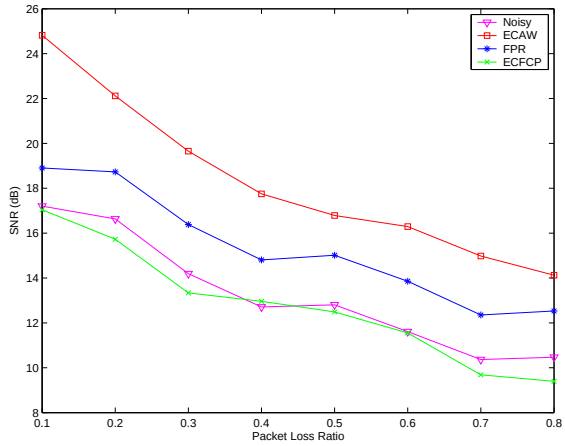


(e) SNR values of Clip-5

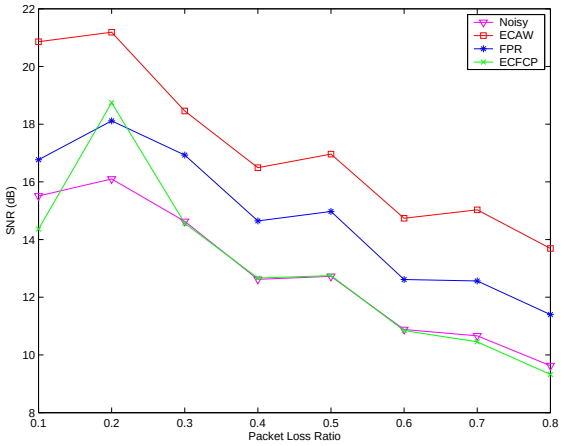


(f) SNR values of Clip-6

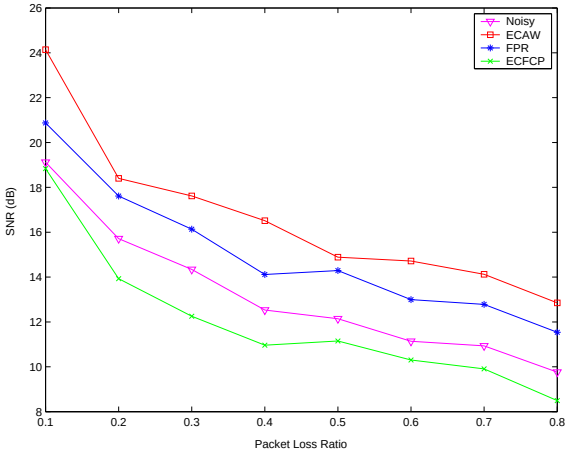
Figure 4.8. Performance results for various clips at good channel conditions for 64 bits packet loss.



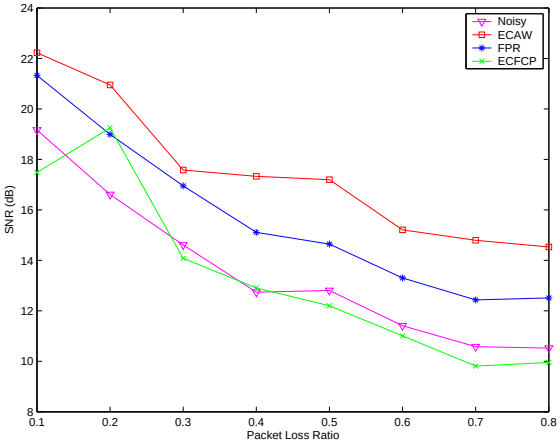
(a) SNR values of Clip-1



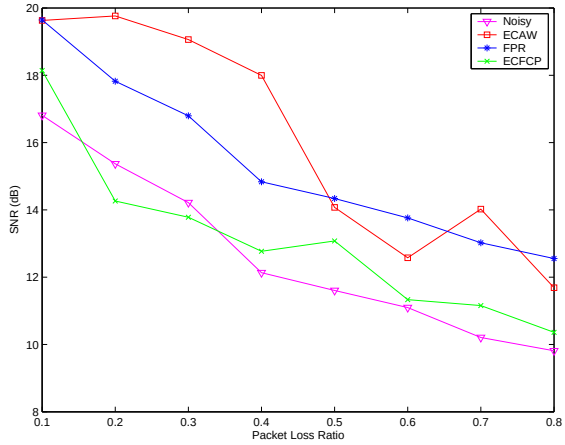
(b) SNR values of Clip-2



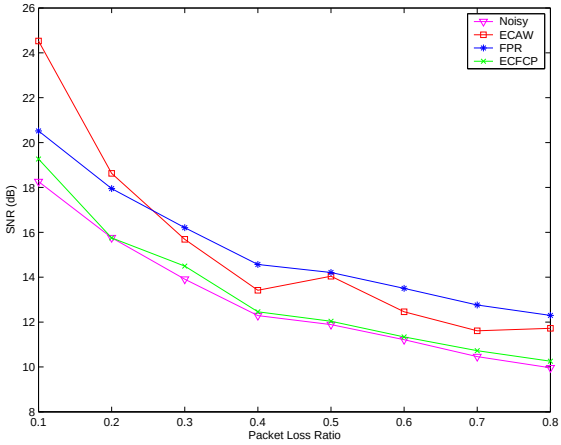
(c) SNR values of Clip-3



(d) SNR values of Clip-4



(e) SNR values of Clip-5

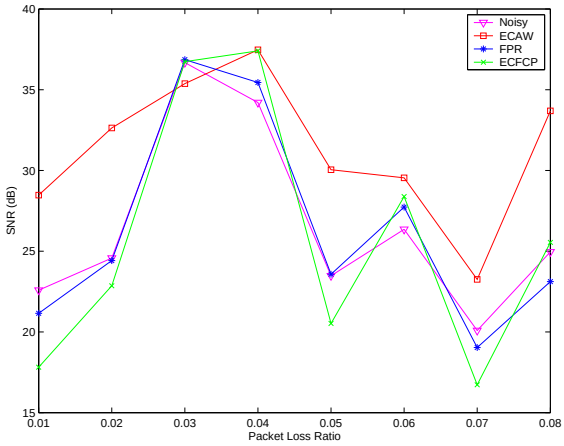


(f) SNR values of Clip-6

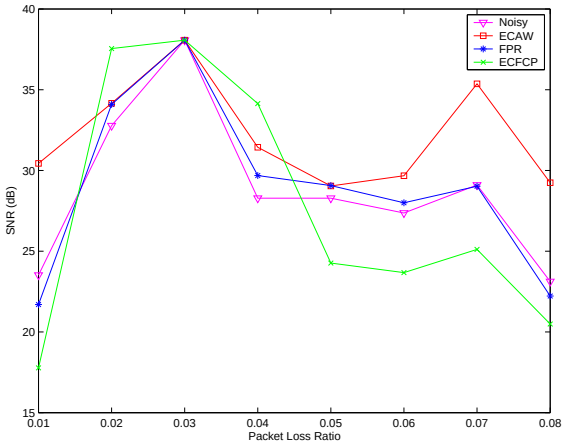
Figure 4.9. Performance results for various clips at bad channel conditions for 64 bits packet loss.

Table 4.5. SNR comparison (in dB) for 128 bits packet loss

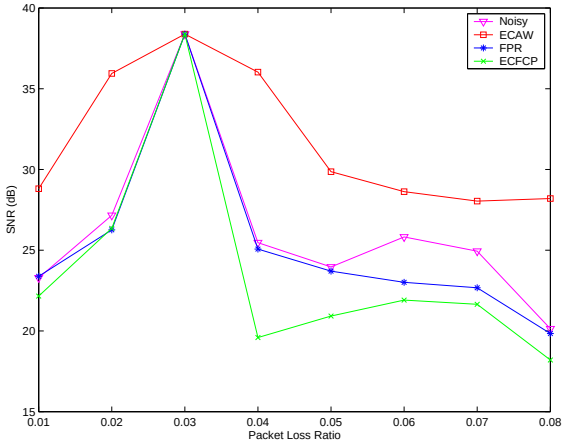
		Packet Loss Ratio				
		0.01	0.02	0.05	0.10	0.20
Clip1	ECAW	28.47	32.63	30.05	22.69	20.20
	FPR	21.15	24.41	23.57	17.41	15.37
	ECFCP	17.81	22.86	20.53	13.79	11.65
Clip2	ECAW	35.67	33.72	28.58	23.67	23.57
	FPR	30.05	29.15	26.68	17.41	19.63
	ECFCP	27.43	29.62	23.17	14.78	16.17
Clip3	ECAW	28.81	35.96	29.86	20.42	22.04
	FPR	23.38	26.27	23.70	15.85	16.60
	ECFCP	22.16	26.35	20.92	13.32	13.72
Clip4	ECAW	31.66	33.41	27.73	25.01	20.50
	FPR	22.97	28.39	22.46	20.00	17.26
	ECFCP	22.08	28.61	20.62	20.09	15.45
Clip5	ECAW	33.99	36.38	30.51	18.86	18.55
	FPR	27.38	35.78	23.33	21.31	15.94
	ECFCP	27.68	35.34	23.45	19.24	13.24
Clip6	ECAW	30.43	34.16	29.04	25.90	23.60
	FPR	21.70	34.09	29.06	22.46	18.64
	ECFCP	17.77	37.55	24.27	20.15	16.87



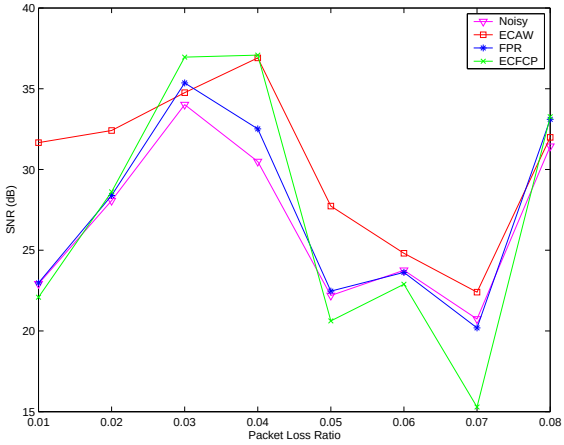
(a) SNR values of Clip-1



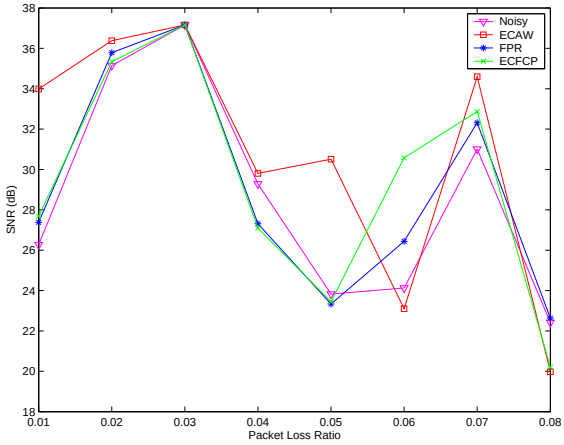
(b) SNR values of Clip-2



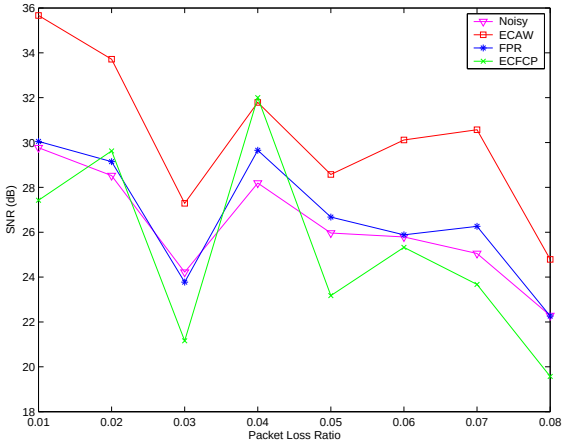
(c) SNR values of Clip-3



(d) SNR values of Clip-4

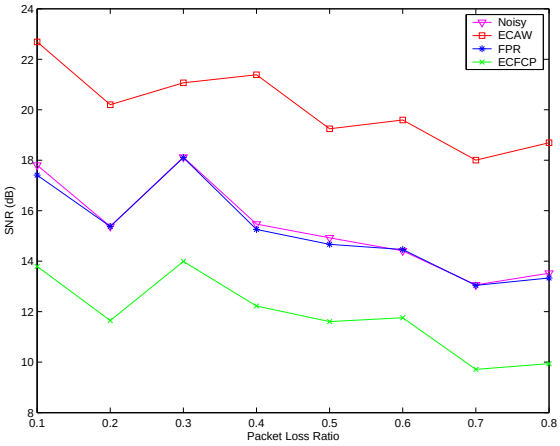


(e) SNR values of Clip-5

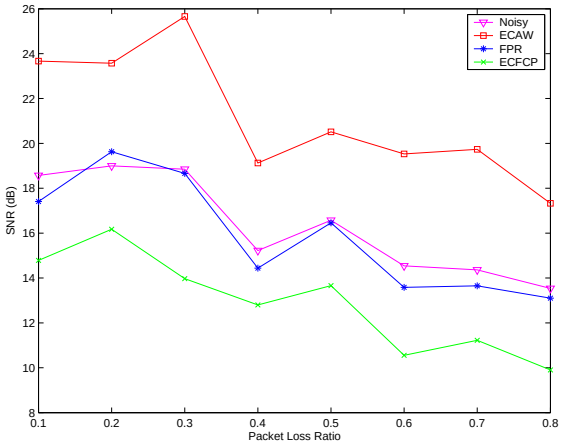


(f) SNR values of Clip-6

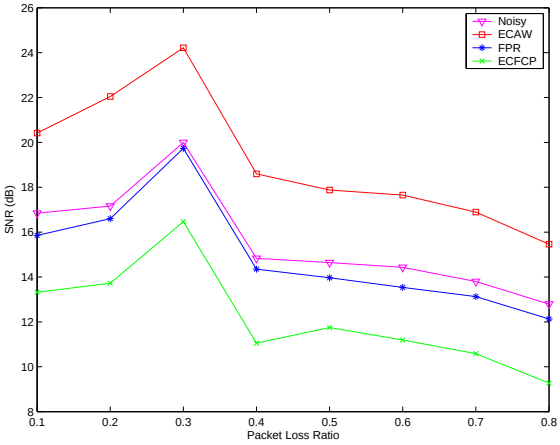
Figure 4.10. Performance results for various clips at good channel conditions for 128 bits packet loss.



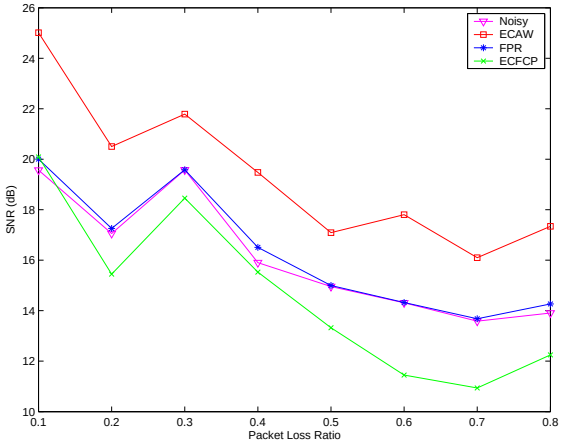
(a) SNR values of Clip-1



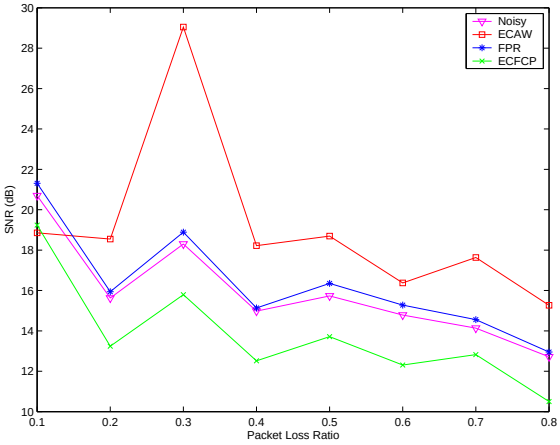
(b) SNR values of Clip-2



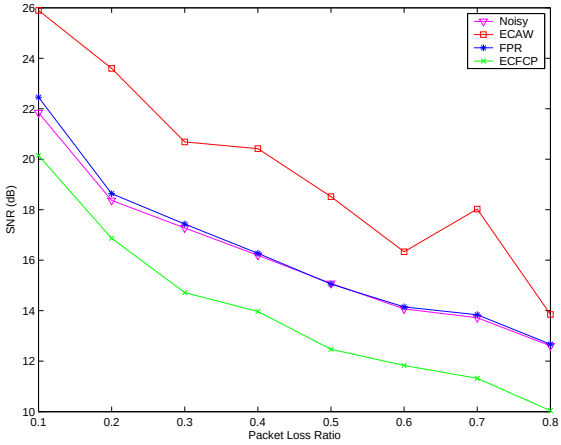
(c) SNR values of Clip-3



(d) SNR values of Clip-4



(e) SNR values of Clip-5



(f) SNR values of Clip-6

Figure 4.11. Performance results for various clips at bad channel conditions for 128 bits packet loss.

4.4. Adaptive Error Concealment (AEC)

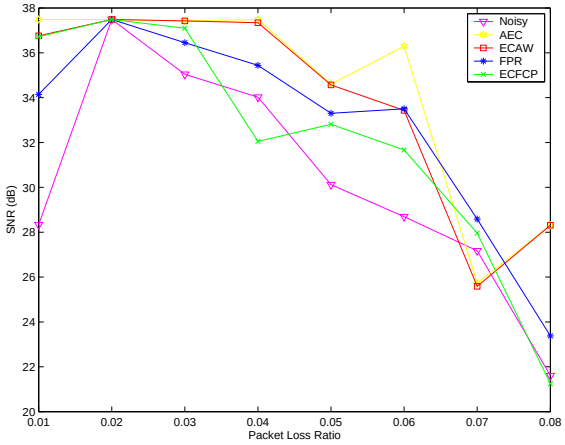
In the previous section, ECAW SNR performance evaluation against FPR and ECFCP displays that each error concealment method is effective for different conditions. This suggests the implementation of an adaptive error concealment method, which detects the condition and switches between these error concealment methods to offer the best performance. In this section, the simulation result of such an adaptive implementation is presented. The simulation environment is the same as the previous section, except that the loss packet sizes are not constant but changing randomly between 16,32,64 and 128 bits.

4.4.1. SNR (Signal-to-Noise Ratio) Evaluation

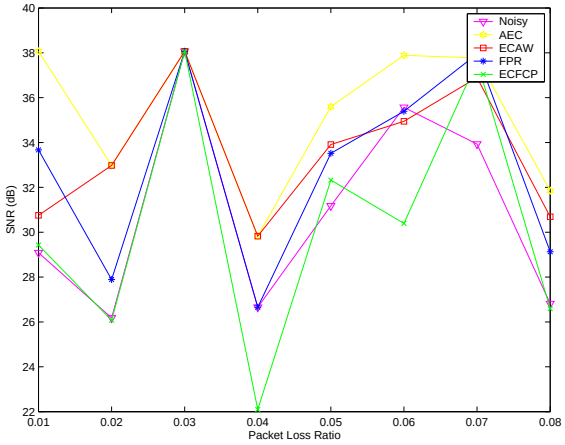
As shown in Table 4.6, Figure 4.12 and Figure 4.13, AEC produces the best SNR values as stipulated. AEC algorithm is sensitive to size of loss packet and it switches to FPR for small loss packet size and to ECAW for large loss packet size. It explains why this adaptive algorithm offers a better SNR performance than FPR and ECAW.

Table 4.6. SNR comparison (in dB) for various sized packet loss.

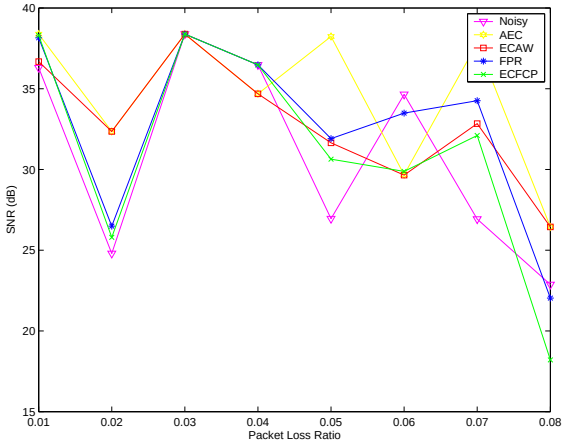
		Packet Loss Ratio				
		0.01	0.02	0.05	0.10	0.20
Clip1	AEC	37.49	37.49	34.60	27.90	28.43
	ECAW	36.76	37.49	34.57	27.29	27.37
	FPR	34.15	37.49	33.30	23.28	23.11
	ECFCP	36.72	37.49	32.81	18.86	21.83
Clip2	AEC	38.07	32.98	35.60	29.42	29.80
	ECAW	30.75	32.98	33.91	29.05	28.41
	FPR	33.67	27.89	33.52	24.74	23.09
	ECFCP	29.43	26.07	32.33	24.41	20.91
Clip3	AEC	38.43	32.35	38.23	27.26	30.32
	ECAW	36.68	32.35	31.65	24.73	30.22
	FPR	38.15	26.49	31.91	20.78	24.52
	ECFCP	38.31	25.80	30.64	18.51	22.36
Clip4	AEC	37.59	37.59	37.25	29.93	27.93
	ECAW	37.35	37.59	34.65	28.80	26.41
	FPR	36.97	37.59	32.11	22.91	21.83
	ECFCP	36.62	37.59	32.47	22.53	19.03
Clip5	AEC	37.16	37.14	37.08	28.45	27.77
	ECAW	33.06	37.14	36.55	28.45	27.76
	FPR	34.77	36.31	34.27	25.19	23.32
	ECFCP	35.56	35.55	36.70	18.75	23.96
Clip6	AEC	36.25	34.61	33.95	31.64	27.74
	ECAW	35.33	34.59	33.80	31.32	26.78
	FPR	35.88	31.74	30.13	27.90	23.79
	ECFCP	35.84	33.33	30.86	24.81	22.45



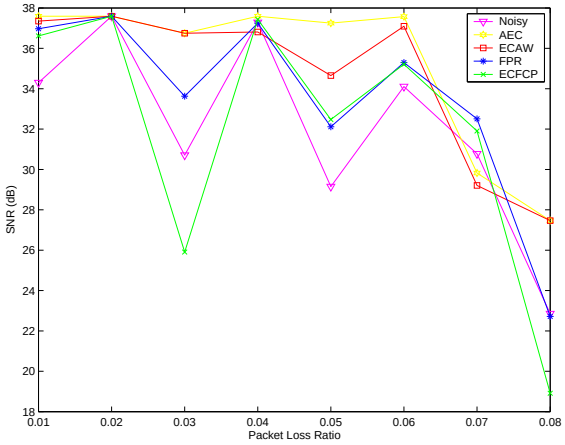
(a) SNR values of Clip-1



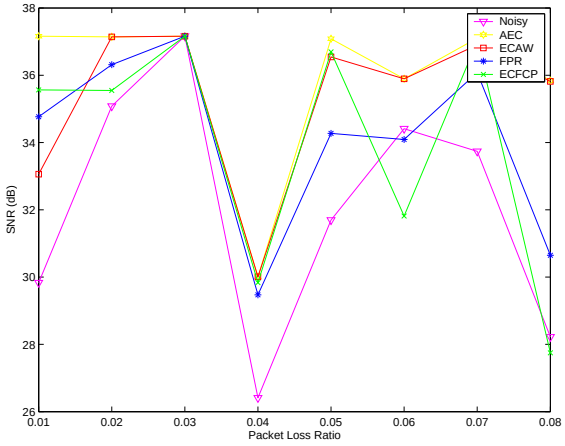
(b) SNR values of Clip-2



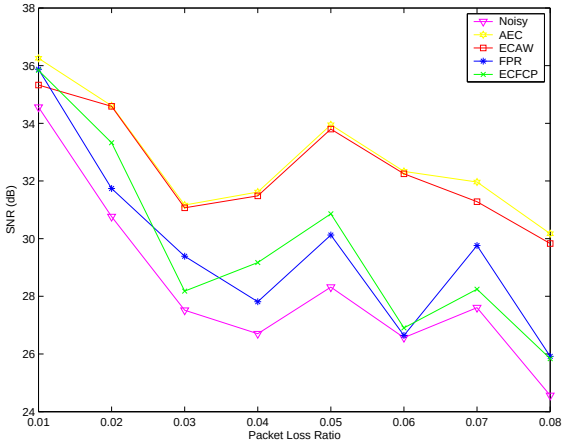
(c) SNR values of Clip-3



(d) SNR values of Clip-4

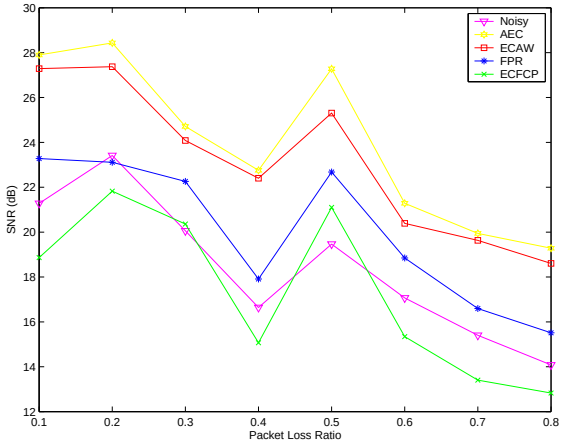


(e) SNR values of Clip-5

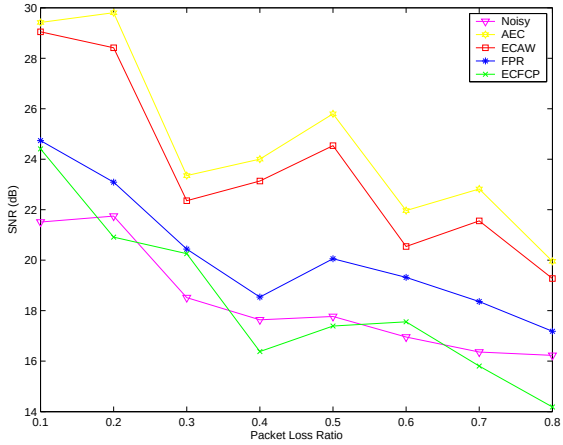


(f) SNR values of Clip-6

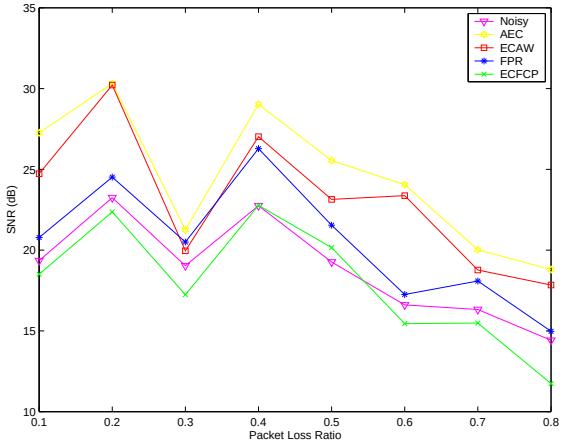
Figure 4.12. Performance results for various clips at good channel conditions for various sized packet loss.



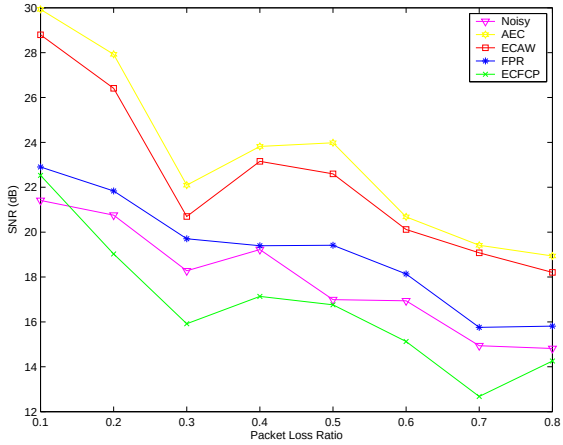
(a) SNR values of Clip-1



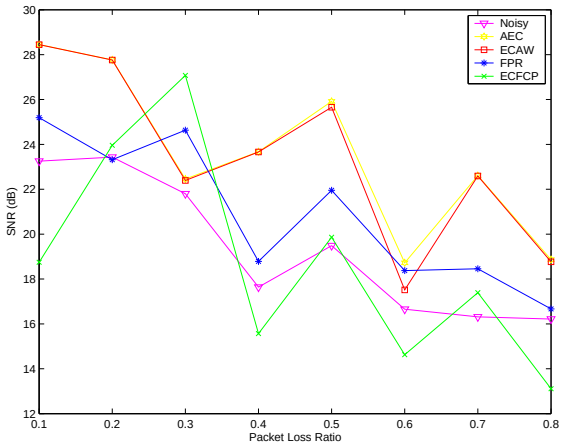
(b) SNR values of Clip-2



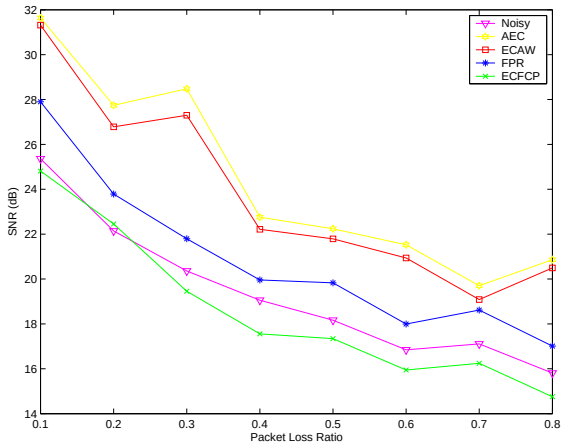
(c) SNR values of Clip-3



(d) SNR values of Clip-4



(e) SNR values of Clip-5



(f) SNR values of Clip-6

Figure 4.13. Performance results for various clips at bad channel conditions for various sized packet loss.

5. FUTURE WORK

Not all combinations of audio watermarking techniques and speech encoding techniques work well together, as they may use conflicting manipulations for their own goals. For example, watermarking least significant bit technique creates capacity by using least significant bit of the speech signal samples, while all lossy speech encoding techniques change these bits during their operation, i.e. they erase the embedded hidden data.

In the early phase of the simulations, the capacity of implemented audio watermarking techniques and robustness against various audio encoding methods were analyzed and best fitting ones which are LSB Coding technique and PCM 8000 hz, 8 bit mono encoding, were used to complete the rest of the simulations.

As a future work, other audio encoding methods and watermarking techniques can be modified to enhance the robustness and capacity of the auxiliary hidden channel created by the proposed method.

The presented simulation results were generated by using the Gilbert-Elliot channel model for various channel conditions. Other channel models, e.g. Land Mobile Satellite (LMS) Channel, Binary Symmetric Channel, Multipath Rayleigh Fading Channel and Rician Fading Channel can be simulated to investigate the efficacy the system for these channels.

6. CONCLUSIONS

Considering the below-mentioned requirements as mandatory and some audio quality loss as acceptable in speech communication, then the proposed error concealment method is applicable and shows consistent SNR advantage over conventional concealment methods.

- No retransmission delay
- No extra redundancy overheads

Performance of the proposed method was compared against to FPR and ECFCP methods for various speech clips and channel conditions. Simulation results prove that the proposed method has significant SNR advantage over these methods when the loss packet size is big enough to carry the meaningful signal features. For smaller packet sizes, reference methods are more successfull than ECAW.

In this thesis, the implementation of an adaptive error concealment method, namely AEC, which detects the condition and switches between these error concealment methods to offer the best performance, is suggested, too. AEC produced the best SNR values as stipulated. AEC algorithm is sensitive to size of loss packet and it switches to FPR for small loss packet size and to ECAW for large loss packet size. This explains why this adaptive algorithm offers a better SNR performance than FPR and ECAW.

Although, in this thesis, the performance evaluation of the proposed method was carried out on PCM speech signal transmitted through Gilbert-Elliot channel model, our method can be extended and its performance can be improved by using other combinations of speech encoding formats and channel models.

REFERENCES

1. Wang, Y. and Q. F. Zhu, "Error Control and Concealment for Video Communication: A Review", *Procceedings of the IEEE*, Vol. 86, pp. 974-997, May 1998.

2. Wah, B., X. Su and D. Lin, "A Survey of Error Concealment Schemes for Real-Time Audio and Video Transmissions over the Internet", *Proceedings of the IEEE International Symposium on Multimedia Software Engineering*, December 2000

3. Perkins, C., O. Hodson and V. Hardman, "A Survey of Packet-Loss Recovery Techniques for Streaming Audio", *IEEE Network Magazine*, October 1998

4. Wang, Y., S. Wenger, J. Wen and A. K. Katsaggelos, "Review of Error Resilient Coding Techniques for Real-Time Video Communications", *IEEE Signal Processing Magazine*, Vol. 17, No. 4, July 2000

5. Cheng, S., H. Yu, and Z. Xiong, "Error Concealment of MPEG-2 AAC Audio Using Modulo Watermarks", *IEEE International Symposium on Publication*, Vol. 2, pp. 261-264, 2002

6. Gruber, J. and L. Strawezynski, "Subjective Effects of Variable Delay and Clipping in Dynamically Managed Voice Systems", *IEEE Trans. Communication*, Vol. 33-8, pp. 801-808, August 1985

7. Tatlas, N., A. Floros, T. Zarouchas and J. Mourjopoulos, "An Error Concealment Technique for Wireless Digital Audio Delivery", *5th International Conference on Communication Systems, Networks and Digital Signal Processing*, Patras, July 2006

8. Yin, H., X. Xie and J. Kuang, "A New Error Concealment Technique for VoIP Based on Forward Contour Prediction", *Proceeding of the 7th International Conference on Signal Processing*, 2006

9. Cvejic, N., "Algorithms for Audio Watermarking and Steganography", *M.S. Thesis*, Oulu University, Finland, 2004

10. Kim, H.J., "Audio Watermarking Techniques", *M.S. Thesis*, Department of Control and Instrumentation Engineering, Kangwon National University, Korea, 2003

11. Şehirli, M., "A Comparative Study of Audio Watermarking Techniques in Time, Frequency and Cepstrum Domains", *M.S. Thesis*, Department of System and Control Engineering, Boğaziçi University, Turkey, 2002

12. Alpaydın, E., *Introduction to Machine Learning*, The MIT Press, 2004

13. Gürgen, F., M. Şehirli and S. İkizoğlu, "Performance Evaluation of Digital Audio Watermarking Techniques Designed in Time, Frequency and Cepstrum Domains", *Springer Verlag LNCS Proc.*, 2004

14. Bender, W., D. Gruhl and N. Morimoto, "Techniques for Data Hiding", *IBM Systems Journal*, Vol. 35, pp. 313-336, 2000

15. Kuo, S., J. Johnston, W.Turin and S.Quackenbush, "Covert Audio Watermarking Using Perceptually Tuned Signal Independent Multiband Phase Modulation", *Proceeding of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 1753-1756, Orlando, FL, 2001

16. Gang, L., A. Akansu and M. Ramkumar, "Mp3 Resistant Oblivious Steganography", *Proceeding of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1365-1368, Salt Lake City, UT, 2001

17. Arnold, M., S. Wolthusen and M. Schmucker, *Techniques and Applications of Digital Watermarking and Content Protection*, Artech House, Norwood, MA, 2003

18. Woodard, J., *Speech Coding*, http://www-mobile.ecs.soton.ac.uk/speech_codecs/standards/pcm.html, 2003

19. Woodard, J., *Speech Coding*, http://www-mobile.ecs.soton.ac.uk/speech_codecs/standards/adpcm.html, 2003

20. Woodard, J., *Speech Coding*, http://www-mobile.ecs.soton.ac.uk/speech_codecs/standards/gsm.html, 2003

21. DSP Group, *TrueSpeech from DSP Group*, http://www.speech.cs.cmu.edu/comp.speech/Section3/Software/truespeech.html, 1996

22. Cherian, T., *Wikipedia*, http://en.wikipedia.org/wiki/Truespeech, 2008

23. Latimer, D., *Wikipedia*, http://en.wikipedia.org/wiki/MP3, 2008

24. Gür, G., "Error Concealment for Images Over Wireless Network Using Watermarking", *M.S. Thesis*, Department of Electrical and Electronics Engineering, Boğaziçi University, Turkey, 2005