UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA

La Universidad Católica de Loja

ESCUELA DE ELECTRÓNICA Y TELECOMUNICACIONES

ESTUDIO DE UN CÓDEC DE COMPRESIÓN PARA MEJORAR LA CALIDAD DE SERVICIO DE SONIDOS ESTETOSCÓPICOS SOBRE UNA RED IP

Proyecto de Fin de Carrera previo a la obtención del título de
Ingeniería en Electrónica y
Telecomunicaciones

AUTORES

Tuesman Daniel Castillo Calvas

Jaime Stalin Romero Narváez

DIRECTOR

Ing. Katty Alexandra Rohoden Jaramillo

LOJA-ECUADOR

2009

CERTIFICACIÓN: ACEPTACIÓN PROYECTO DE FIN DE CARRERA

Loja, 14 de octubre de 2009

Ing. Katty Alexandra Rohoden
Escuela de Electrónica y Telecomunicaciones – GESE

Dejo constancia de haber revisado y estar de acuerdo con el proyecto de fin de carrera, titulado: "Estudio de un códec de compresión para mejorar la calidad de servicio de sonidos estetoscópicos sobre una red IP"

Presentado por: Tuesman Daniel Castillo Calvas
Jaime Stalin Romero Narváez

Particular que comunico para los fines legales pertinentes.

Ing. Katty Alexandra Rohoden

Visto Bueno Dirección Escuela

F).....
Ing. Jorge Luis Jaramillo Pacheco
DIRECTOR DE LA ESCUELA DE ELECTRÓNICA Y TELECOMUNICACIONES

CESIÓN DE DERECHOS

Tuesman Daniel Castillo Calvas y Jaime Stalin Romero Narváez declaramos ser autores del presente trabajo y eximo expresamente a la Universidad Técnica Particular de Loja y a sus representantes legales de posibles reclamos o acciones legales.

Adicionalmente declaro conocer y aceptar la disposición del Art. 67 del Estatuto Orgánico de la Universidad Técnica Particular de Loja que su parte pertinente textualmente dice: “Forman parte del patrimonio de la Universidad la propiedad intelectual de investigaciones, trabajos científicos o técnicos y tesis de grado que se realicen a través, o con el apoyo financiero, académico o institucional (operativo) de la universidad”

Los Autores

…...……..……... …..……

Tuesman D. Castillo C. Jaime S. Romero N.

AUTORÍA

Las ideas, opiniones, conclusiones, recomendaciones y más contenidos expuestos en el presente informe de tesis son de absoluta responsabilidad de los autores.

Tuesman Daniel Castillo Calvas
Jaime Stalin Romero Narváez

INTRODUCCIÓN

En las últimas décadas el desarrollo tecnológico, tanto en la electrónica como en el procesamiento de señales, ha permitido el aparecimiento de dispositivos de diagnóstico médico más precisos y complejos en comparación con el tradicional estetoscopio. Este dispositivo se ha visto parcialmente reemplazado en los modernos centros de salud por técnicas como el electrocardiograma y el ultrasonido, sin embargo la simplicidad y la eficacia que provee el estetoscopio al momento de realizar una valoración médica inmediata, ha hecho que evolucionen en forma de estetoscopios electrónicos o digitales para integrarse a nuevos escenarios y tecnologías como la construcción de fonocardiogramas en tiempo real, la grabación y posterior reproducción de sonidos en el mismo dispositivo para su análisis y, cómo no, la telemedicina.

Los sistemas de telemedicina aprovechan las bondades de las telecomunicaciones para implementar nuevos métodos de consulta a distancia, valiéndose de dispositivos electrónicos de diagnóstico y de captura de información médica que deben operar de forma remota. Una de las características importantes de la telemedicina es la de transmitir y controlar el abundante flujo de información que se produce en una teleconsulta, el ancho de banda de un sistema de comunicaciones siempre será limitado ya sea por el elevado costo de implementar una nueva red, porque se tiene que operar sobre redes alquiladas o por un proveedor de servicios de internet. Es así, que es necesario contar con técnicas que permiten aprovechar de mejor manera el ancho de banda disponible en una red IP, como la compresión de datos, sonido e imágenes y protocolos de comunicación eficientes

El presente trabajo enfoca la selección de un códec de compresión que permita disminuir el ancho de banda que demanda una red IP para la transmisión de sonidos provenientes de un estetoscopio digital, partiendo del estudio y análisis de las principales características y naturaleza de los sonidos fisiológicos percibidos por dicho dispositivo con el objeto de delinear el tipo de compresión y códec a utilizar, para lo cual se propone un códec modelo. Luego se elegirá un códec existente en el mercado que se adapte tanto a los requerimientos planteados como al modelo

propuesto; finalmente se presenta un acercamiento a la transmisión en tiempo real con redes IP, que permita la auscultación remota de los sonidos cardiacos y respiratorios, como una técnica innovadora en el campo de la telemedicina. Además, en el Anexo 2 se detalla cómo sería la implementación del sistema de auscultación remota sobre una red inalámbrica y el objeto de estudio del presente trabajo que es el códec para mejorar la calidad de servicio en la transmisión de sonidos estetoscópicos.

OBJETIVOS

Objetivo General

Realizar el estudio de un códec de compresión para mejorar la calidad de servicio de sonidos estetoscópicos sobre una red IP.

Objetivos Específicos

- Estudiar las características fisiológicas y técnicas de los sonidos cardiacos y respiratorios para determinar la frecuencia óptima de operación.
- Proponer el esquema de un codec de compresión que se adapte a las características que requiere el sistema para mejorar la calidad de servicio en la transmisión de sonidos estetoscópicos.
- Seleccionar un códec de compresión de audio existente en el mercado con características similares al codec propuesto.
- Estudiar y analizar el codec seleccionado para demostrar su factibilidad en la transmisión de sonidos estetoscópicos sobre redes IP.

DEDICATORIA

A mis padres Jaime y Magdaly.
A mi hermano Henry.
De manera significativa a mi Madrecita que me brindó todo el apoyo incondicional para la culminación de un meta mas en mi vida.
Con especial afecto a quienes me dieron su apoyo moral y espiritual hasta la consecución de mis estudios superiores.

Jaime Stalin

A mis padres, por su apoyo y sacrificio
A mis hermanas por su cariño y comprensión
A mi abuelita Clotilde que me enseñó las primeras letras
A Benito.

Tuesman Daniel

AGRADECIMIENTO

Los autores queremos expresar nuestros más sinceros agradecimientos a las autoridades y maestros de la Universidad Técnica Particular de Loja, de manera especial a la escuela de Electrónica y Telecomunicaciones, a su cuerpo docente por habernos dado la oportunidad de trabajar en el tema: “Estudio de un códec de compresión para mejorar la calidad de servicio de sonidos estetoscópicos sobre una red IP”, facilitándonos las herramientas, asesorías y conocimiento, permitiendo de esta manera obtener el título de Ingeniería en Electrónica y Telecomunicaciones.

A la Ing. Katty Rohoden por su acertada dirección durante el transcurso de este trabajo.

A nuestros padres y familiares por su compresión y ayuda desinteresada en el presente trabajo.

Los Autores

TABLA DE CONTENIDO

LISTA DE FIGURAS.

LISTA DE TABLAS

1 Sonidos Cardiacos y Respiratorios.

1.1 Antecedentes

En el presente capítulo se describe al estetoscopio digital, su estructura, los modelos que existen en el mercado y sus características, también se explica la importancia de la detección de los sonidos torácicos, (cardiacos y respiratorios) además de sus características frecuenciales que son de importancia para delimitar las características del códec a diseñar.

1.2 Auscultación

Dentro del campo del diagnostico médico, la auscultación se destaca debido a que es uno de los métodos más antiguos y rutinarios de valoración de un paciente, percibiendo los sonidos originados por las diferentes estructuras anatómicas ya sea acercando el oído al paciente o por medio de un estetoscopio. [16]

1.3 Estetoscopio Convencional.

Un estetoscopio es un dispositivo que permite escuchar sonidos cardiacos, respiratorios y abdominales, es utilizado en la etapa de diagnóstico durante el proceso de auscultación. Principalmente consta de una pieza a manera de campana que se presiona contra el pecho del paciente, un tubo flexible que lleva el sonido captado por la campana hasta dos piezas que se posicionan en los oídos del médico. Actualmente los estetoscopios acústicos convencionales son similares al mostrado en la Figura 1.1, contienen la siguiente estructura[1]:

1) Binaural. Es la parte metálica que se ajusta al tubo y las olivas.
2) Olivas. Son pequeñas piezas que se ajustan a los canales auditivos del usuario.
3) Arco metálico. Es la parte a la que se ajustan las olivas y el tubo.
4) Diafragma y campana: El diafragma se utiliza para percibir los sonidos agudos o de alta frecuencia y la campana sirve para amplificar los sonidos de baja frecuencia.
5) Vástago. Conecta el tubo con la campana
6) Tubo flexible.

[1] Tomado de Estetoscopios 3M® Littman
http://solutions.3m.com.mx/wps/portal/3M/es_MX/Littmann/stethoscope/

7) Campana entonable: La campana es la parte del estetoscopio a través de la cual se captan los sonidos del paciente.

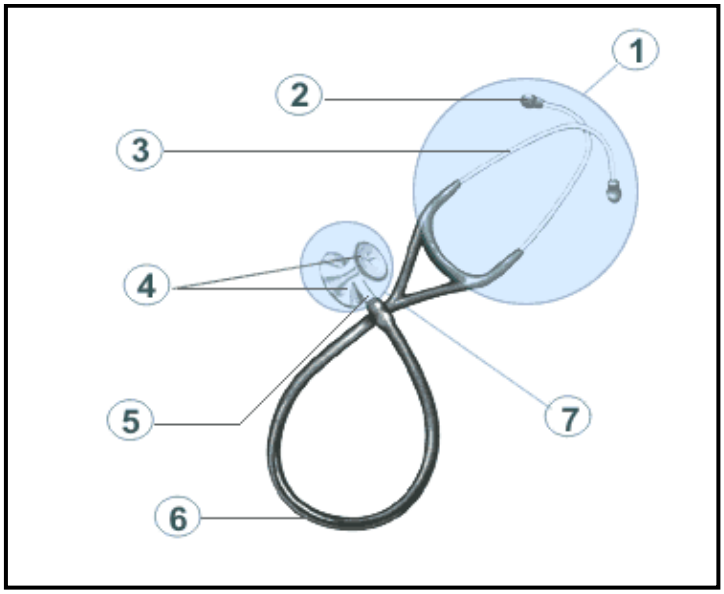


Figura 1.1 Estetoscopio acústico 3M®[2]

El estetoscopio convencional permite acercamientos importantes en el diagnóstico de patologías cardiacas y pulmonares, aunque existen nuevos métodos de diagnóstico como el doppler y el ultrasonido, sin embargo, este dispositivo no puede ser remplazado en su totalidad ya que es una herramienta indispensable y de primera mano para todo profesional de la salud, además destaca su bajo costo y fácil acceso.

1.4 Estetoscopio Digital

El mayor problema con los estetoscopios convencionales es que no poseen un sistema de amplificación lo que hace difícil percibir los sonidos corporales por alguien que posee limitaciones auditivas, o falta de experiencia en la técnica auscultatoria. Un estetoscopio digital o electrónico se caracteriza por tener la capacidad de transformar los sonidos percibidos por la campana o la membrana en señales eléctricas y codificarlos digitalmente. Esta característica amplía la funcionalidad del dispositivo, permitiendo el procesamiento de las señales para la amplificación, el almacenamiento digital para posterior análisis o incluirlos en la

[2] Tomado de Estetoscopios 3M® Littman
http://solutions.3m.com.mx/wps/portal/3M/es_MX/Littmann/stethoscope/

historia médica del paciente, la construcción de fonocardiogramas (FCG) que es el registro gráfico de los sonidos del corazón, o la transmisión a través de una red digital, característica que la telemedicina ha aprovechado.

La microelectrónica y fuentes de energía más eficientes han permitido la miniaturización de estos instrumentos y ahora se pueden encontrar en el mercado estetoscopios electrónicos portátiles e incluso algunos que permiten la grabación de sonidos.

Existen diferentes modelos de estetoscopios digitales en etapa de prototipo como por ejemplo:

Prototipo de Estetoscopio Inteligente para el Telediagnóstico de Sonidos y Soplos Cardíacos.

El estetoscopio esta desarrollado en base a un sistema denominado ASEPTIC[3] (Aided System for Event-based Phonocardiogram Telediagnosis with Integrated Compression), que permite el procesamiento y compresión del fonocardiograma. ASEPTIC consta de dos partes principales:

- Primera etapa de procesamiento que analiza la señal FCG y determina el estado cardiovascular.
- Segunda etapa que comprime el FCG para transmitirlo de forma remota con unos requerimientos de ancho de banda bajos.

El diseño electrónico del prototipo de estetoscopio electrónico consta de varios módulos:

1. Subsistema analógico, para capturar y acondicionar la señal FCG
2. Subsistema digital (FPGA), para procesar el FCG después de haberse convertido a digital con el conversor Análogo/Digital
3. Microprocesador.
4. Interfaz humano (pantalla LCD y teclado).
5. Módulo Bluetooth para comunicaciones.
6. Sistema de configuración de la FPGA.

[3]Martínez Alajarín J., López Candel J., Ruiz Merino R. Prototipo de Estetoscopio Inteligente para el Telediagnóstico de Sonidos y Soplos Cardíacos

1.4.1 Estetoscopios Digitales Comerciales

Algunas de los principales modelos de estetoscopios electrónicos que se pueden encontrar en el mercado se han resumido en la Tabla 1.1. Otros dispositivos poseen además un sistema de comunicación por puerto serial RS-232 para conectarse a un PC, un módem, o un códec de videoconferencia H.320. Esto permite transmitir las señales en tiempo real a través de IP o RDSI. También permite realizar FCGs con herramientas de software, que tienen una respuesta de frecuencia de 20 a 1000 Hz y una frecuencia de muestreo de 1,8KHz con una resolución de 16 bits.

Tabla 1.1 Ejemplos de estetoscopios digitales comerciales

Marca	Modelo	Características
3M	Littmann® Model 4100WS	Reducción de ruido externo de 75%
		Incluye software
		Transmisión por infrarrojo 115Kbps de velocidad
		3 modos de frecuencia: campana (20-200Hz), diafragma (100 – 500 Hz). Extensión hasta 1Khz.
		Grabación y reproducción de 6 pistas de 8 segundos en formato WAV
		Frecuencia cardiaca en LCD
Sonolife Prosound		Frecuencia 20-2000Hz
		Pantalla LCD que muestra FCG
		Amplificación X32
Thinklabs	ds32a	Respuesta en frecuencia de 15Hz-20000Hz
		Modo acústico y electrónico
		Diafragma ajustable
Cardionics	E-Scope II	Interface PDA
		Cambio de frecuencia: Corazón (20 – 1000Hz), pulmones (70 – 2000Hz)
		Varios modos de audífonos
Welch Allyn	Master Elite	Software para PC
		Modos de campana y diafragma (20-20000Hz)
		170 gramos
ADSCOPE	657	Campana (15-200Hz), diafragma (100 – 500Hz), extensión (15- 4000Hz)
		Batería de litio 3V (150hrs)
		Apagado automático

1.5 Sonidos Cardiacos.

Los eventos cardíacos, que se presentan desde el inicio de un latido hasta el inicio del próximo, se conocen como ciclo cardíaco y constan básicamente de un período de diástole, durante el cual los ventrículos se llenan de sangre seguida de un período de sístole, en el que la sangre es expulsada a las arterias. Los sonidos producidos por estos eventos son causados según la teoría valvular por vibraciones variantes en el tiempo, resultado de la tensión abrupta que se produce sobre las valvas de las válvulas cardiacas al final de su cierre y apertura, pero la teoría hemodinámica plantea que los sonidos son causados por las vibraciones de toda la estructura cardiaca como consecuencia de la aceleración y desaceleración de los fluidos de sangre intracardiacos seguidos por el cierre y apertura de las válvulas, esta última teoría es la más aceptada actualmente [9].

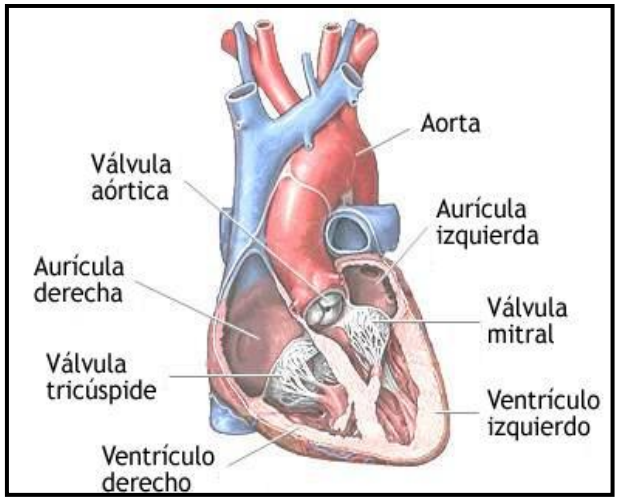


Figura 1.2 Principales partes funcionales que componen al corazón[4]

1.5.1 Ruidos cardiacos normales.

A los ruidos cardiacos normales se los separa comúnmente en dos sonidos que se los denomina, primero y segundo ruido (S1 y S2) [9].

El Primer Ruido (S1) es de tono bajo, timbre suave y duración algo más prolongada que el segundo. Se produce por el cierre de las válvulas mitral (M1) y tricúspide (T1) al comienzo de la sístole ventricular, este sonido se divide en cuatro componentes, el primer componente es de baja frecuencia cuando la primera contracción del miocardio en el ventrículo empuja la sangre hacia las aurículas, el segundo es de alta frecuencia y comienza con la tensión abrupta del cierre de las

[4] Tomado de: www.clinicadam.com/salud/5/003266.html

válvulas auriculoventriculares (AV), desacelerando la sangre, luego las válvulas sigmoideas se abren y el flujo de la sangre es expulsado hacia los ventrículos, el tercer es causado por la oscilación de la sangre entre la raíz de la aorta y las paredes ventriculares, el cuarto componente es generado por las vibraciones producidas por la turbulencia en la expulsión de la sangre a través de la aorta y de la arteria pulmonar [9].

El segundo ruido (S2) es de tono algo más agudo que el primero y de duración más breve. Ocurre al finalizar la sístole ventricular y al empezar la relajación ventricular, lo constituyen dos componentes de alta frecuencia, que son causados por el cierre de la válvula aórtica (A2) y pulmonar (P2) [9].

Ambos sonidos se pueden diferenciar en la banda de los 20 Hz y 150 Hz del espectro audible [6]. En la Figura 1.3 se puede distinguir la distribución de estos sonidos en un periodo cardiaco.

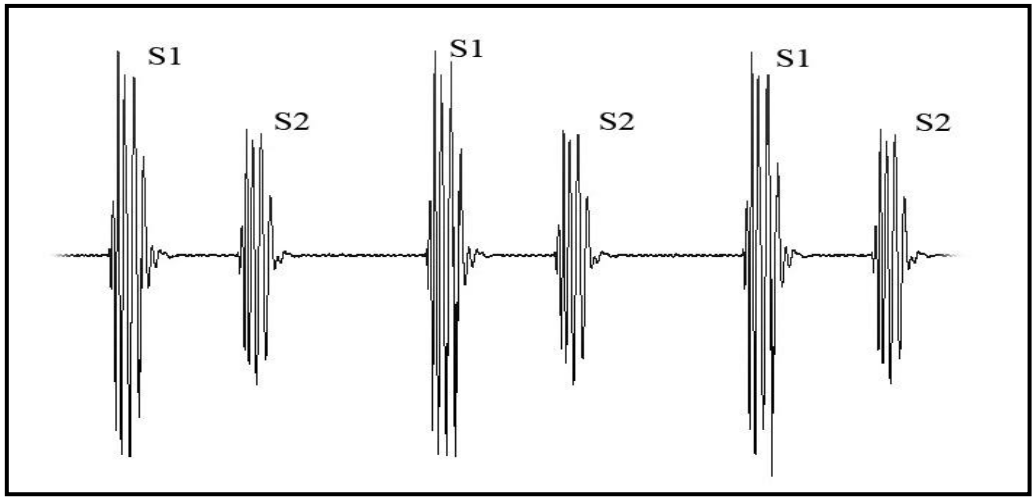


Figura 1.3 Nomenclatura de los sonidos cardiacos S1 y S2[5]

1.5.2 Ruidos cardiacos anormales

Se encuentran los siguientes ruidos:

- Ruidos de llenado ventricular: Tercer ruido y Cuarto ruido (S3 y S4).
- Soplos Cardiacos.

A continuación se describe cada uno de estos ruidos anormales [9].

[5] Ilustración obtenida a través de la representación gráfica de un sonido cardiaco normal descargado desde: http://solutions.3m.com.mx/wps/portal/3M/es_MX/Littmann/stethoscope/3M CL Sonidos de Corazón y Pulmón.mht

1.5.2.1 Tercer ruido

Se trata de un ruido de baja frecuencia que ocurre entre 0.12s a 0.18s después de S2 que corresponde al llenado ventricular. Puede ser fisiológico en niños y puede escucharse en individuos incluso hasta la adolescencia. [16]

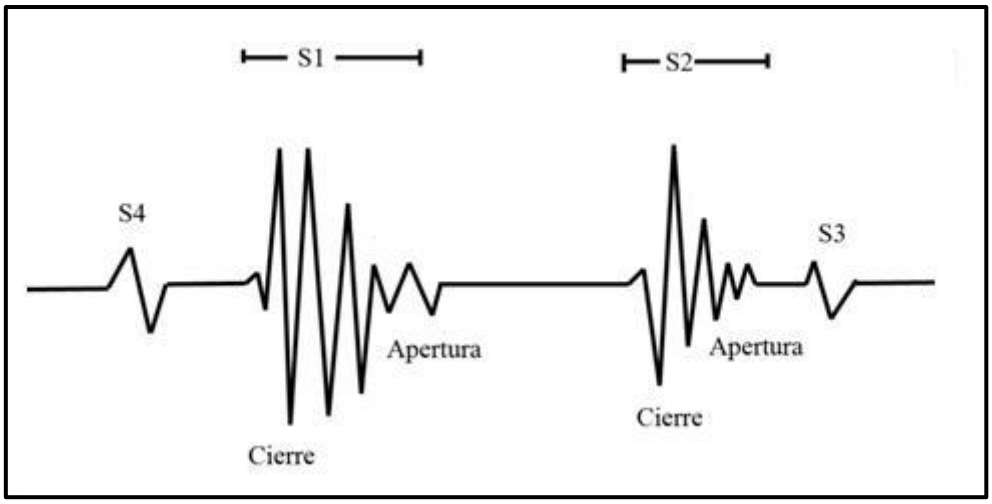


Figura 1.4 Componentes de los ruidos cardiacos [6]

1.5.2.2 Cuarto ruido

Se trata de un ruido de baja frecuencia que aparece al final de la diástole, justo antes del primer ruido. Es generado por la contracción de las aurículas desplazando el flujo dentro de los ventrículos. [16]

Tabla 1.2 Rango de frecuencias de sonidos cardiacos [6]

Ruido	Duración [s]	Rango frecuencial [Hz]
S1	0.1-0.12	20-150
S2	0.08-0.14	50-60
S3	0.04-0.05	20-50
S4	0.04-0.05	<25

1.5.2.3 Soplos cardiacos.

Los soplos ocurren cuando una válvula no se cierra bien y la sangre se regresa o cuando la sangre fluye a través de una abertura estrecha o de una válvula rígida. Se conocen principalmente dos tipos de deficiencias en el funcionamiento valvular: la estenosis, que consiste en la inadecuada apertura valvular, y la insuficiencia o regurgitación, que se presenta cuando la válvula no se cierra suficientemente lo que ocasiona un reflujo de sangre en sentido inverso al normal. [25]

1.5.2.3.1 Soplos sistólicos.

- Soplo de expulsión: Comienza cuando el flujo se inicia en uno de los grandes vasos y termina antes del cierre valvular.
- Soplo holosistólico: Comienzan con S1 y continúan hasta S2, es decir, ocupan todo el período sistólico. Normalmente, son causados por insuficiencia de una o ambas válvulas AV (mitral o tricúspide), o por comunicación interventricular [25].

1.5.2.3.2 Soplos diastólicos.

Son causados por valvulopatías graves.

- Protodiastólicos: Se presentan solamente al inicio de la diástole. Generalmente, son causados por insuficiencia de una o ambas válvulas sigmoideas (aórtica y pulmonar).
- Presistólico: Son causados por una disminución del radio en las válvulas AV, como por ejemplo en casos de estenosis mitral y tricuspídea [25].

En la Figura 1.5 se muestra a una señal proveniente de un FCG que muestra un soplo cardiaco sistólico, nótese cómo el primer ruido S1 es alterado en comparación con el S1 de la señal normal.

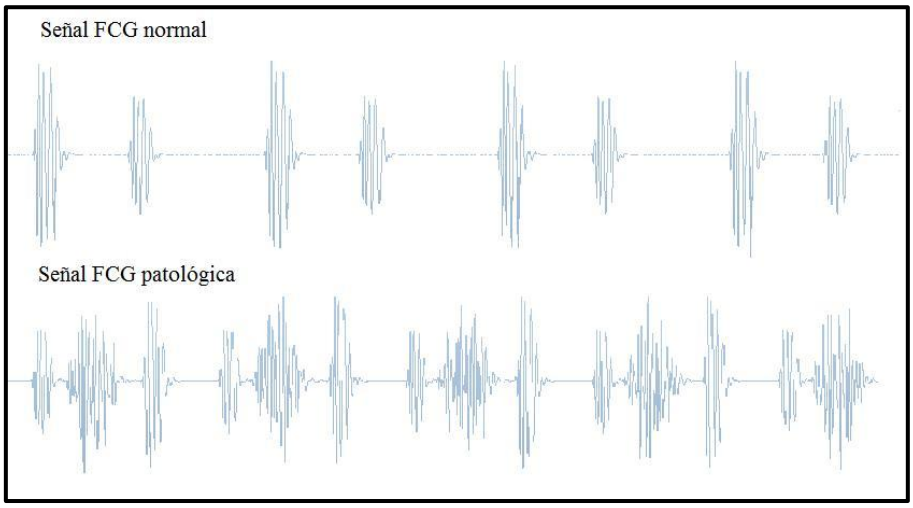


Figura 1.5 Comparación entre una señal cardiaca normal y una que denota patología[6].

[6] Ilustración obtenida a través de la representación gráfica de un sonido cardiaco normal descargado desde: http://solutions.3m.com.mx/wps/portal/3M/es_MX/Littmann/stethoscope/3M CL Sonidos de Corazón y Pulmón.mht

1.6 Sonidos Respiratorios.

Estos sonidos son producidos por las estructuras pulmonares durante la acción de la respiración, su análisis permite diagnosticar patologías o síndromes en un paciente, en la Figura 1.6 se observa el espectro de los sonidos normales.

La clasificación de estos sonidos basados en el análisis digital es [19]:

1) Sonidos Básicos o normales
2) Sonidos Adventicios o anormales.

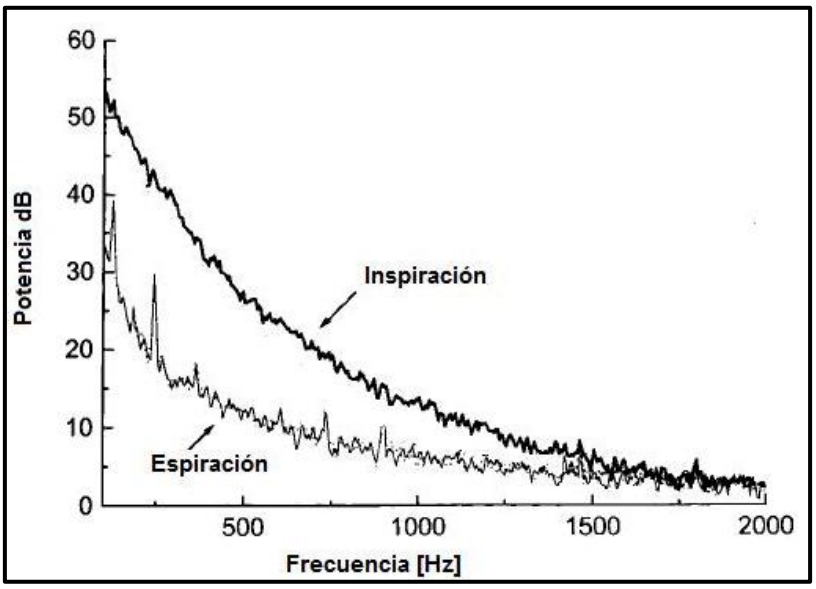


Figura 1.6 Espectro de los sonidos de los pulmones de un adulto sano [20]

1.6.1 Sonidos Básicos o Normales [19]

Soplo Traqueal: Muestra dos componentes: inspiratorio y espiratorio, es audible en la región donde se proyecta la tráquea y región esternal, se origina por el paso del aire a través de la hendidura glótica y la tráquea.

Soplo bronquial: Corresponde al ruido traqueal audible en la zona donde se proyectan los bronquios de mayor calibre, en la cara anterior del tórax y proximidades del esternón. Es muy similar al ruido traqueal, del cual se distingue solo por su componente espiratorio menos intenso.

Murmullo vesicular: Los ruidos respiratorios que se escuchan en la mayor parte del tórax se deben a la turbulencia del aire circulante al chocar contra las partes salientes de las bifurcaciones bronquiales y al pasar de una cavidad a otra de diámetro diferente, como de los bronquiolos a los alveolos y viceversa.

El componente inspiratorio es más intenso, duradero y de tonalidad más alta que el componente espiratorio. El murmullo vesicular es un sonido más débil y suave que

la respiración bronquial. Este sonido se escucha en bases, vértices y regiones costales del tórax.

Murmullo broncovesicular: En este sonido se suman las características de la respiración bronquial con las del murmullo vesicular. La intensidad y la duración de la inspiración y espiración son de igual magnitud, ambas son más fuertes que el murmullo vesicular.

Los sonidos básicos o normales se resumen para objeto de análisis digital en sonidos pulmonares normales y el sonido traqueal normal, estos se encuentran entre los 100 Hz y 800Hz para el sonido pulmonar normal y de 200 Hz a 1500Hz en el caso del sonido traqueal normal [19].

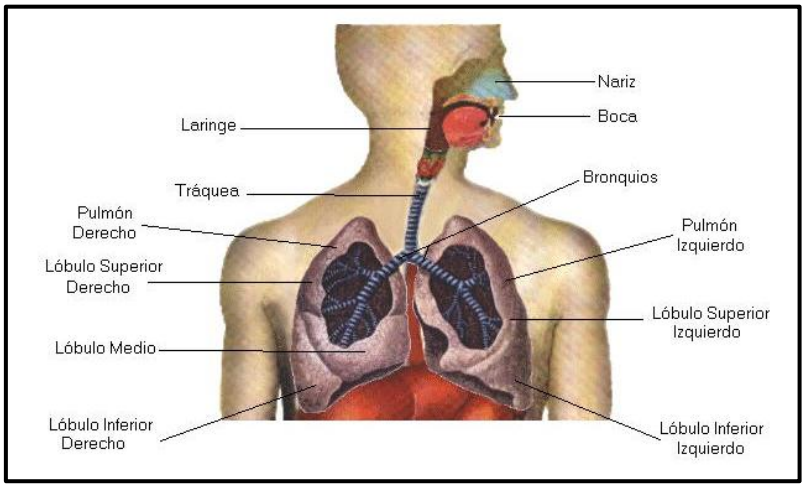


Figura 1.7 Aparato respiratorio[7]

1.6.2 Sonidos Adventicios o anormales [19]

Roncus: Son ruidos graves e intensos que se originan por la vibración de las paredes bronquiales y del contenido gaseoso cuando hay estrechamiento en estos conductos, por espasmo, edema de la pared o presencia de secreciones adheridas a las paredes de la vía respiratoria, se encuentran en frecuencias menores a 300Hz.

Sibilancias: Son ruidos agudos originados por el paso del aire a través del árbol traqueobronquial que ha disminuido su calibre por obstrucción, son característicos del asma. Cuando las sibilancias son circunscritas a determinada región indican la presencia de obstrucción parcial por cuerpo extraño, su rango de frecuencia varía entre 100 a 1000Hz.

[7] Tomado de: http://aire-en-aqp-msalazar-4a-35.blogspot.com/2007/06/aparato-respiratorio.html

Estridor: Es un ruido agudo y sibilante producido por obstrucción parcial de la laringe o la tráquea y puede deberse a difteria, laringitis aguda, cáncer laríngeo y estenosis traqueal, su frecuencia es mayor a 200Hz y menor a 1500Hz [19].

Tabla 1.3 Características frecuenciales de los sonidos respiratorios [20]

Sonido	Rango frecuencial [Hz]
Pulmonar normal	100 - 800
Traqueal normal	200 - 1500
Roncus	< 300
Sibilancias	100 - 1000
Estridor	200 - 1500

2 Codificación y Compresión de Audio Digital.

2.1 Introducción.

La compresión de audio digital permite el eficiente almacenamiento y transmisión de datos de audio. Existen varias técnicas de compresión que nos ofrecen distintos niveles de complejidad, calidad de compresión de audio, y cantidad de compresión de datos.

En este capítulo se analizaran el proceso de digitalización de la señal de audio y las diferentes técnicas utilizadas para comprimir señales de audio digital.

2.2 Audio Digital

La frecuencia máxima que el oído humano puede percibir es 20 o 22kHz, como nos muestra el teorema de Nyquist la optima digitalización de audio se debe hacer a una tasa de muestreo al doble de la frecuencia máxima, es decir a una frecuencia alrededor de 44kHz, esta frecuencia es la que se utiliza para grabar sobre un CD por ejemplo [24].

La representación digital de una señal de audio ofrece muchas ventajas: alta inmunidad al ruido, estabilidad y reproductibilidad. Además el audio digital permite la eficiente implementación de funciones de procesamiento de audio como: mezcladores, filtros, ecualizadores.

El proceso de conversión de una señal de audio analógica a digital, inicia con el muestreo de la señal en intervalos discretos de tiempo, para luego cuantizar la misma, asignando a cada muestra un nivel con un valor diferente. Los datos de audio digital consisten en una secuencia de valores binarios que representan el número de niveles cuantizados para cada muestra de audio. El método para presentar cada muestra con un código independiente se denomina modulación por codificación de pulso (PCM) [18].

Para el proceso completo de digitalización es necesario atravesar tres etapas: Muestreo, Cuantización y Codificación.

2.2.1 Codificación de Audio

Las técnicas de compresión se dividen en codificación con o sin pérdida. La mayoría de los codificadores de audio que se utilizan hoy en día, emplean las dos técnicas para obtener un mayor porcentaje de compresión manteniendo un nivel de audio aceptable.

2.2.2 Codificación sin Pérdida

Un codificador sin pérdida es capaz de reconstruir la señal perfectamente. Se basa en reducir la información redundante que está presente en los datos a comprimir [14].

Dentro de las técnicas empleadas para compresión sin pérdidas, las más utilizadas para codificar audio tenemos: codificación Huffman y codificación Aritmética.

Debido a la naturaleza del audio, resulta difícil conseguir una buena compresión utilizando las técnicas de codificación sin pérdida.

2.2.2.1 Codificación Huffman

La codificación Huffman es una codificación por entropía la cual analiza la ocurrencia de cada uno de los símbolos y los reemplaza con un código de longitud menor.

El algoritmo básico consiste en, tras ordenar los símbolos de mayor a menor en probabilidad, ir juntando parejas de menor probabilidad formando un árbol. Cuando solamente hay dos raíces, se asigna los símbolos 0 y 1 a cada raíz, y se itera hacia atrás [12].

Dada, por ejemplo, una fuente F con 5 símbolos de probabilidades {1/2, 1/4, 1/8, 1/16, 1/16} se podría construir el siguiente diagrama de árbol Figura 2.1. En cada etapa los dos elementos con menor probabilidad se unen y se obtiene un elemento resultante cuya probabilidad es la suma de las dos anteriores. Cada vez que se realiza una unión de dos elementos se asigna a cada uno de ellos un 1 o un 0. El proceso termina cuando únicamente quedan dos. Finalmente para conocer el código asociado a cada probabilidad se recorre el árbol en sentido inverso y se relacionan los unos o ceros por los que se va pasando hasta llegar al principio de cada ramificación [12].

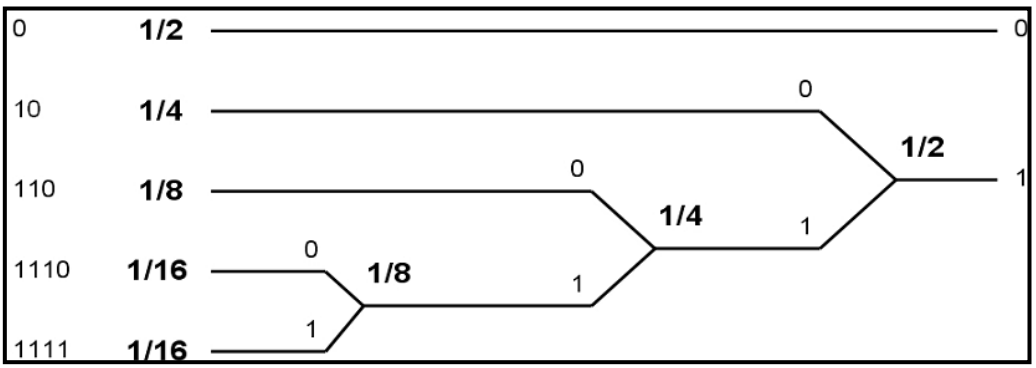


Figura 2.1 Árbol Codificación Huffman [12]

2.2.2.2 Codificación Aritmética

Con esta técnica de codificación no es necesario que las probabilidades de los símbolos del alfabeto fuente sean potencias de dos para obtener una eficiencia óptima.

En la codificación aritmética no se asigna una palabra de código a cada uno de los símbolos del alfabeto fuente como se hace en la técnica anteriormente vista. En esta técnica lo que se hace es codificar una secuencia de entrada de símbolos del alfabeto fuente mediante un número representado en punto flotante [8].

El proceso de codificación se basa en asignar a cada símbolo un intervalo entre 0 y 1, de forma que la amplitud de cada intervalo sea igual a la probabilidad de cada símbolo. La suma de las amplitudes de los intervalos debe ser igual a la unidad.

Previamente es necesario establecer un orden entre los símbolos. No es necesario seguir algún criterio especial para establecer un orden entre los símbolos del alfabeto fuente, pero el orden establecido debe ser conocido por el decodificador para poder hacer una correcta decodificación en la recepción.

Para realizar la codificación de una determinada cadena de entrada se siguen los siguientes pasos [8]:

- Se selecciona el primer símbolo de la secuencia de entrada y se localiza el intervalo asociado a ese símbolo.
- A continuación se selecciona el siguiente símbolo y se localiza su intervalo. Se multiplican los extremos de este intervalo por la longitud del intervalo asociado al símbolo anterior (es decir, por la probabilidad del símbolo anterior) y los resultados se suman al extremo inferior del intervalo asociado al símbolo anterior para obtener unos nuevos extremos inferior y superior.
- El paso anterior se repite hasta que todos los símbolos del mensaje hayan sido procesados.
- Para el símbolo i-ésimo se calcula su intervalo de la siguiente forma:
  inf(i)=inf(i-1)+(sup(i-1)-inf(i-1))*inf(i)
  sup(i)=inf(i-1)+(sup(i-1)-inf(i-1))*sup(i)
- Por último se selecciona un valor dentro del intervalo del último símbolo de la secuencia. Este valor representará la secuencia que se desea enviar.

Por ejemplo, se tiene un alfabeto 'a', 'b', 'c', 'd', y 'e' con probabilidades de 0.3, 0.15, 0.25, 0.1 y 0.2, respectivamente. Se elige el intervalo de cada símbolo de acuerdo a su probabilidad Tabla 2.1 [15]:

Tabla 2.1 Rango de los Símbolos [15]

Simbolo	Probabilidad	Intervalo
a	0.30	[0, 0.30)
b	0.15	[0.30, 0.45)
c	0.25	[0.45, 0.70)
d	0.10	[0.70, 0.80)
e	0.20	[0.80, 1.00)

Se desea codificar la secuencia **a c e** con las probabilidades de la Tabla 2.1:
Para codificar cada símbolo de la secuencia se realizan los pasos detallados anteriormente:
Codificación de la 'a'
Rango actual = 1 - 0 = 1
Límite superior = 0 + (1 × 0,3) = 0,3
Límite inferior = 0 + (1 × 0,0) = 0,0

Codificación de la 'c'
Rango actual = 0,3 - 0,0 = 0,3
Límite superior = 0,0 + (0,3 × 0,70) = 0,210
Límite inferior = 0,0 + (0,3 × 0,45) = 0,135

Codificación de la 'e'
Rango actual = 0,210 - 0,135 = 0,075
Límite superior = 0,135 + (0,075 × 1,00) = 0,210
Límite inferior = 0,135 + (0,075 × 0,80) = 0,195

La secuencia "ace" se puede codificar mediante cualquier valor dentro del rango [0,195, 0,210).

2.2.3 Codificación con Perdida

Los métodos de compresión con pérdida usados en audio utilizan técnicas donde se alcanzan grandes porcentajes de compresión, debido a que se elimina

información de tipo perceptiva, es decir imperceptible por el oído humano, y además redundancias estadísticas que se encuentran en la señal original.

Debido a que hay partes de la señal original que se eliminan no va a ser posible reconstruir la señal al completo tras el proceso de decodificación. Comparando la codificación de audio sin pérdida donde la tasa de información normalmente alcanza alrededor de 10 bits/muestra, la codificación con pérdida puede alcanzar tasas de menos de 1 bit/muestra.

2.2.4 Codificación de audio de tipo psicoacústico

La codificación de tipo psicoacústico comprende un conjunto de métodos de compresión con pérdida que intentan eliminar el sonido que no es percibido por el oído humano. La mayoría de las personas tienen un rango de audición que va desde los 20 Hz hasta los 20 KHz con una resolución de la frecuencia para cada intervalo de unos 2 Hz. Escuchar un determinado sonido va a variar dependiendo de su frecuencia, el oído humano es normalmente más sensible en la región de los 3 KHz como podemos observar en la Figura 2.3. Los sonidos que tienen una amplitud muy pequeña cuando estamos analizando el espectro de la señal de audio pueden directamente ser eliminados [5].

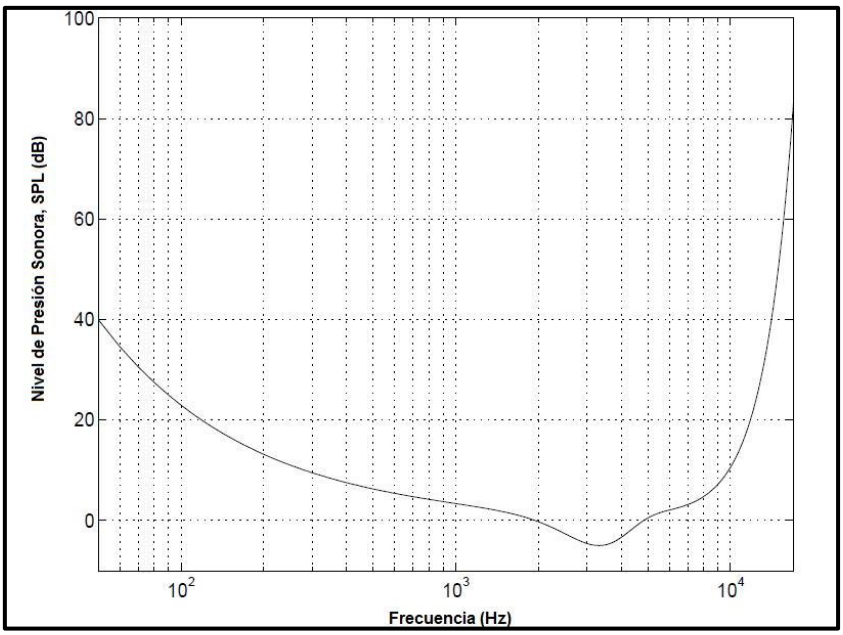


Figura 2.2 Nivel de Presión Sonora [18]

2.2.4.1 La Escala Decibel

Para que un sonido sea perceptible por el oído humano la onda sonora debe tener una intensidad mayor o igual $1x10^{-12}$watts/m^2, esta intensidad llamada umbral auditivo es extremadamente pequeña si se compara con el umbral del dolor cuyo valor es de 10 watts/m^2 [5]:

Para poder operar dentro de este rango, se utiliza una escala logarítmica denominada escala decibel, que se denota por la siguiente ecuación [10]:

$$L_q = 10\, log_{10} \frac{q}{q_{ref}} \quad dB \qquad (2.1)$$

Donde:

$L_q =$ Intensidad sonora [dB]

$q =$ Cantidad que se desea expresar en dB

$q_{ref} =$ Cantidad de referencia

Utilizando la ecuación anterior se puede calcular el nivel de intensidad de sonido como [5]:

$$L_I = 10\, log_{10} \frac{I}{I_{ref}} \quad dB \qquad (2.2)$$

Donde:

$I_{ref} = 1x10^{-12}$watts/m^2

El nivel de presión se calcula mediante [5]:

$$SPL = 10\, log_{10} \frac{P^2}{P_{ref}{}^2}\, dB = 20\, log_{10} \frac{P}{P_{ref}}\, dB \qquad (2.3)$$

Donde:

$P =$ Presión del sonido RMS en Pascales (Pa)

$P_{ref} =$ 20 µPa, que presenta la presión de sonido a 1000Hz

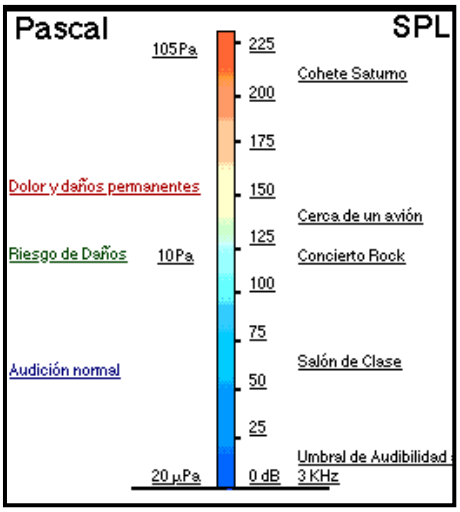


Figura 2.3 Niveles de Presión Sonora

2.2.4.2 Umbral Auditivo

El umbral auditivo varía según la frecuencia y es el nivel de presión sonoro que es necesario para que un tono puro se pueda oír en un ambiente sin ruido. El umbral en dB [23] puede ser aproximado por la ecuación (2.4):

$$A(f) = 3{,}64\left(\frac{f}{1kHz}\right)^{-0{,}8} - e^{-0{,}6\left(\frac{f}{1kHz}-3{,}3\right)^2} + 10^{-3}\left(\frac{f}{1kHz}\right)^4 dB \qquad (2.4)$$

Donde:

$f =$ Frecuencia expresada en KHz

Además, el umbral auditivo es más bajo en frecuencias que se aproximan a la de la voz y más alto en frecuencias altas y bajas.

2.2.4.3 Enmascaramiento simultáneo

El enmascaramiento simultáneo es un fenómeno que se produce cuando un sonido se hace inaudible en presencia de otro sonido más fuerte. Tanto la frecuencia como la fase de la señal afectan al enmascaramiento [29].

Como ejemplo del enmascaramiento, supongamos que estamos en un concierto de música y deseamos tener una conversación con la persona que está a nuestro lado, el volumen tan alto hará que la voz de la persona sea inentendible o imperceptible; la onda del sonido está presente, pero el oído es incapaz de escucharla.

El oído humano percibe sonidos mediante una descomposición en frecuencia realizada en la cóclea: oscilaciones a lo largo de la membrana basilar determinan que frecuencia es audible [29]. En la Figura 2.5 se muestra una ilustración de la cóclea y se indican las localidades en que se perciben ciertas frecuencias. Como se observa las localidades no están dispuestas de manera uniforme respecto a la frecuencia, por lo cual para modelar el oído humano es indispensable utilizar otra escala denominada: Escala de Bark.

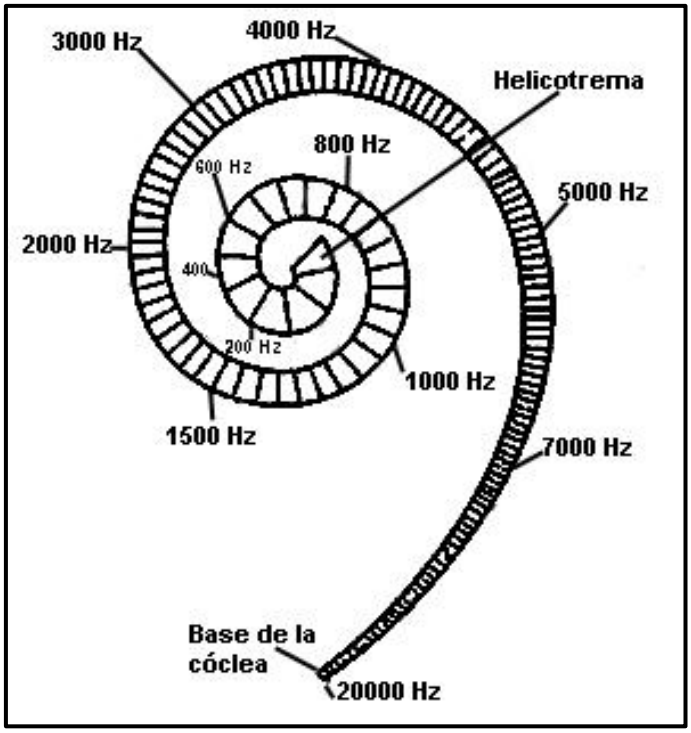


Figura 2.4 La cóclea y localidades audibles

2.2.4.4 Escala de Bark

Es una escala no lineal que agrupa frecuencias en bandas, la cual tiene propiedades casi lineales para bajas frecuencias y logarítmicas para altas, adicionalmente describe una distancia fija a lo largo de la membrana basilar[8]. Si se toma en cuenta que el oído humano difícilmente percibe sonidos con una frecuencia superior a los 20 KHz, esto nos da como resultado 25 bandas críticas que se detallan en la Tabla 2.1.

La unidad establecida para contabilizar el porcentaje de banda crítica (CBR) es un Bark y para transformar de Khz a Bark se utiliza la ecuación (2.5) [28]:

[8] www.otorrinoweb.com/_izquie/glosario/m/membrana_basilar.htm

$$\frac{z(f)}{Bark} = 13\ arctan\left(\frac{0{,}76\ f}{1\ KHz}\right) + \ 3{,}5\ arctan\left(\frac{f}{7{,}5\ KHz}\right)^2 \qquad (2.5)$$

Donde f es la frecuencia en KHz.

Tabla 2.2 Escala y división de bandas Bark [28]

Bark	Frecuencia Central (Hz)	Ancho de Banda (Hz)	Bark	Frecuencia Central (Hz)	Ancho de Banda (Hz)
1	50	0 – 100	**14**	2150	2000 – 2320
2	150	100 – 200	**15**	2500	2320 – 2700
3	250	200 – 300	**16**	2900	2700 – 3150
4	350	300 – 400	**17**	3400	3150 – 3700
5	450	400 – 510	**18**	4000	3700 – 4400
6	570	510 – 630	**19**	4800	4400 – 5300
7	700	630 - 770	**20**	5800	5300 – 6400
8	840	770 – 920	**21**	7000	6400 – 7700
9	1000	920 – 1080	**22**	8500	7700 – 9500
10	1175	1080 – 1270	**23**	10500	9500 – 12000
11	1370	1270 – 1480	**24**	13500	12000 – 15500
12	1600	1480 – 1720	**25**	19500	15500 – 20000
13	1850	1720 – 2000			

2.2.4.5 Enmascaramiento temporal

El enmascaramiento temporal es el que se estudia en el tiempo. Puede aparecer antes y después de la presencia de la señal enmascarante. Si el enmascaramiento sucede antes de la presencia del enmascarante se le denomina pre-enmascaramiento; caso contrario cuando el enmascarante ya no está presente se le conoce como post-enmascaramiento [18].

El origen de este fenómeno radica en el tiempo que ocupan los sentidos para ajustarse y percibir las sensaciones. En la Figura 2.6 se ilustra este efecto:

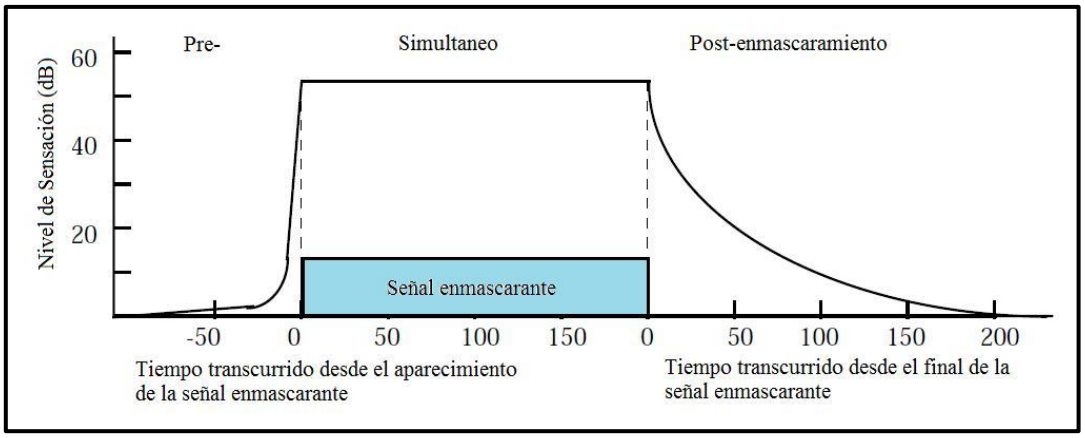


Figura 2.5 Propiedades Temporales de Enmascaramiento [18]

Post-enmascaramiento [3]**:** Se da cuando es el tono de mayor amplitud el que sucede con antelación en el tiempo al de menor amplitud, percibiéndose tan sólo el primer estímulo. Este fenómeno se produce cuando ambos sonidos llegan al oído en un intervalo de tiempo de entre 30 y 60 ms aproximadamente. Esto se debe a que una vez percibido el tono fuerte, el oído necesita un cierto periodo de adaptación.

Pre-enmascaramiento [3]**:** Si se produce primero un estímulo suave y posteriormente un tono intenso, este último enmascarará igualmente al de menor amplitud, siempre y cuando estén separados en el tiempo por una diferencia menor comprendida entre 5 y 10 ms.

2.2.4.6 Umbral de Enmascaramiento [3].

El umbral de enmascaramiento es el nivel de presión sonora (SPL) que se necesita para que el oído humano pueda percibir un sonido, en presencia de una señal enmascarante. Se determina a partir de las curvas de enmascaramiento de los componentes espectrales, enmascaramiento temporal y umbral auditivo.

Una vez que se obtiene la curva de enmascaramiento formada por los enmascaradores individuales esta se combina con el umbral auditivo, para obtener como resultado la curva de umbral global de enmascaramiento.

Como se puede ver en la Figura 2.6, la pendiente del umbral de enmascaramiento es más inclinada hacia las frecuencias bajas; de este modo, las frecuencias mayores son más fácilmente enmascaradas.

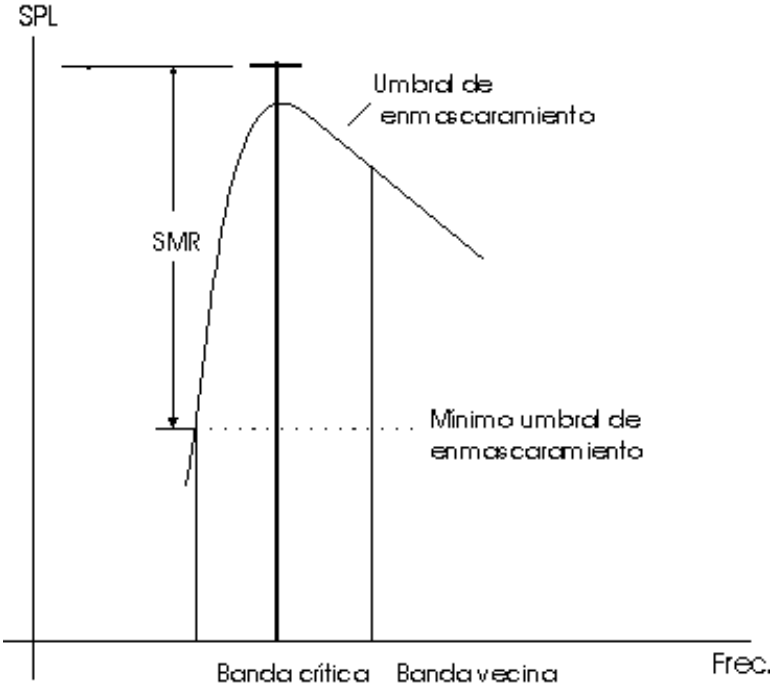


Figura 2.6 Curva del Umbral de Enmascaramiento[9]

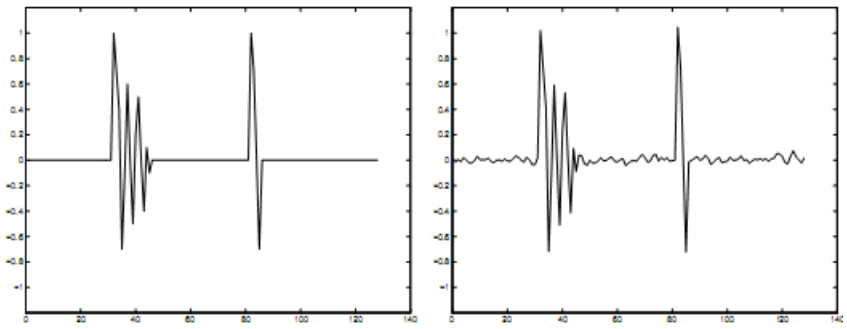
Definimos SMR como: relación señal a máscara. Distancia entre el nivel del enmascarador y el umbral de enmascaramiento. Su máximo valor está en el borde izquierdo de la banda crítica, y su mínimo valor se encuentra en la frecuencia del enmascarador.

2.2.4.7 Pre – Eco

El pre-eco es un fenómeno que se presenta cuando la energía de un sonido transitorio se extiende en el tiempo, esto se produce debido a la cuantización en el dominio de la frecuencia. Cuando se origina esto se puede percibir un pequeño eco antes de la presencia del transitorio [5].

En la Figura 2.8 se ilustra la señal original de un transitorio y la señal resultante luego de aplicar la cuantización, se observa que el ruido de cuantización afecta a toda la señal. En la Figura 2.8 (a) se puede apreciar 3 porciones de silencio y 2 transitorios. Para eliminar el ruido producido en la cuantización Figura 2.8 (b) se hace uso del pre-enmascaramiento logrando que este ruido sea imperceptible. Para que el pre-enmascaramiento sea efectivo es necesario aumentar la resolución en el tiempo cuando se detecte un transitorio.

[9] Tomado de: CODIFICACIÓN PERCEPTIVA DE AUDIO DE ALTA CALIDAD, Rafael Rogriguez.



(a) *Transitorio* (b) *Pre-eco*

Figura 2.7 Efecto de Pre-eco

2.2.5 Vector de Cuantización [7]

El vector de cuantización (VQ) es un método de compresión con pérdida basado en aproximaciones. Una forma obvia de comprimir datos es simplemente reducir el ancho de los bits en los datos. La propuesta para este método sería la cuantización escalar, la cual básicamente viene a decir que solamente los bits más significativos son conservados. Una aproximación mejor y más útil para la codificación de datos que se encuentran altamente correlacionados consiste en llevar los valores dentro de pequeños grupos para codificar datos que son adyacentes.

Este método puede ser aplicado a cualquier conjunto grande de datos, y es particularmente bueno si los puntos consecutivos están relacionados o si la precisión requerida varía sobre el rango de entrada. Si los datos no están relacionados el subconjunto resultante se asemejará bastante a lo que sería una cuantización escalar.

3 Codec propuesto para sonidos estetoscópicos.

3.1 Introducción

El codificador/decodificador tiene como finalidad reducir la cantidad de bits necesarios para transmitir los sonidos cardiacos y respiratorios por una red sin sacrificar la calidad. Para lograr estos requerimientos se utilizará técnicas de compresión con pérdidas debido a la naturaleza del sistema que transmite audio en tiempo real sobre una red IP[10].

3.2 Consideraciones generales

Para diseñar el códec se tomará en cuenta las siguientes consideraciones en base a las características de frecuencia de los sonidos estetoscópicos.

Los algoritmos con pérdidas que trabajan en el dominio de la frecuencia no deben discriminar las siguientes bandas espectrales ya que son de mucha importancia para el correcto diagnostico, como se muestra en la Figura 3.1.

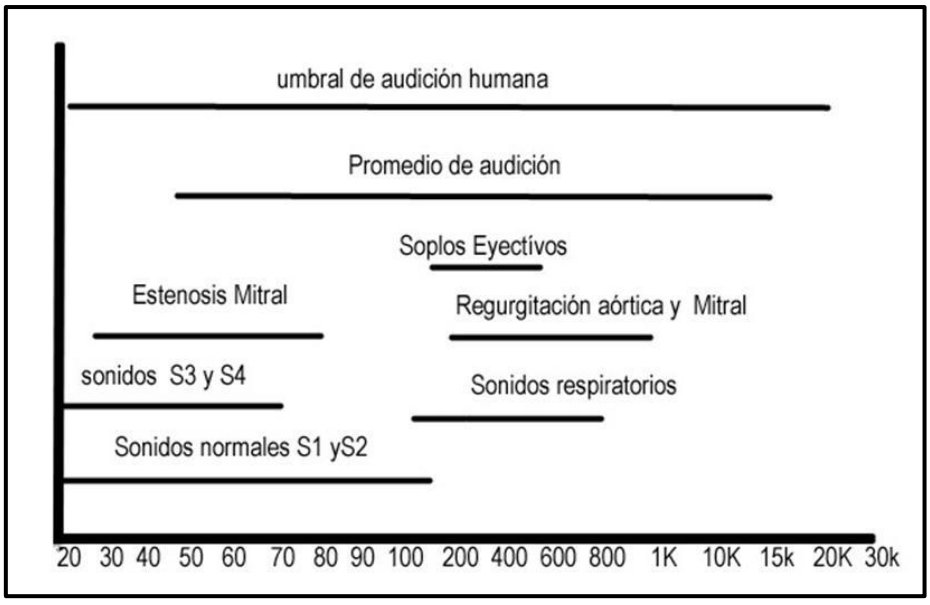


Figura 3.1 Características frecuenciales de los sonidos fisiológicos [13]

Como se puede observar en la figura, las bandas de frecuencia de los diferentes sonidos presentados se encuentran dentro del espectro audible, es así que al hacer una comparación con la máscara del umbral auditivo[11] se considera que es factible aplicar un método de compresión perceptivo debido a que ninguna de las

[10] Véase capítulo 2, “compresión con pérdidas”
[11] Fig 2.2

frecuencias necesarias para el diagnostico se encuentra en las bandas críticas del modelo auditivo humano.
Se debe tomar en cuenta un esquema de codificación de un solo canal debido a que los sonidos del estetoscopio electrónico son monoaurales[12].

El códec tendrá la capacidad para ser transmitido en un flujo de datos en tiempo real, conocido también como streaming, aunque existen varios codecs de compresión de audio pero sólo algunos tienen la capacidad de transmitirse por la red en forma de paquetes y en su destino reproducirse sin necesidad de contar con el archivo completo, así también debe ser soportado por RTP (Real Time Protocol) y UDP (User Datagram Protocol). El ancho de banda que demanda el códec para su transmisión no debe ser superior a 64Kbps, es decir que debe funcionar a bajas tasas de bits por segundo pero sin perjudicar su calidad significativamente.

3.3 Descripción del códec

3.3.1 Codificador

A continuación se detalla la estructura de un códec utilizando la transformada modificada discreta del coseno MDCT para el análisis frecuencial, el modelo psicoacústico (sin perdidas) para el análisis perceptual, y por último para reducir el número de bits redundantes se utilizará codificación Huffman. La figura 3.2 muestra la estructura general del codificador de audio propuesto.

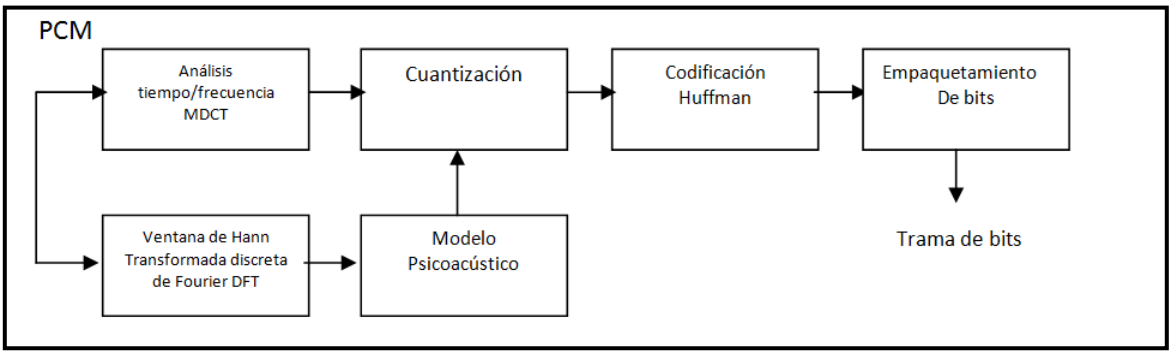


Figura 3.2 Esquema del codificador [13]

[12] Definido por un solo canal.
[13] Realizado en base a la recomendación UIT-R BS.1115

3.3.1.1 MDCT (Transformada discreta de coseno modificada)

La MDCT permite la transformación de una señal temporal al dominio de la frecuencia, esta transformada ha sido adoptada por los códec de alta calidad, debido a que elimina el fenómeno conocido como aliasing[14].

La MDCT es muestreada críticamente. El muestreo crítico mantiene fija la cantidad de datos para reconstruir la señal, es decir, para reconstruir una señal de tamaño N se necesita N coeficientes de la MDCT.

El proceso de la transformada describe dos fases: análisis y síntesis, en la fase de análisis se toman bloques de N muestras con un traslape de 50%, a estos bloques se los multiplica por una ventana de análisis válida que debe cumplir con el principio de Princen-Bradley que dice “La suma al cuadrado de las partes de la ventana que se solapan debe ser igual a 1”, luego se calcula la MDCT a través de la ecuación (3.1) para cada una de los bloques [5].

$$X_i[k] = \sum_{n=0}^{N-1} w_a^i[n] x_i[n] cos\left[\frac{2\pi}{N}(n+n_0)\left(k+\frac{1}{2}\right)\right] \quad (3.1)$$

para $k = 0, \ldots, \frac{N}{2} - 1$

Donde:

$N =$ Número de muestras

$n_0 = \frac{\left(\frac{N}{2}+1\right)}{2}$

w_a^i = ventana de análisis del i-ésimo bloque.

La fase de síntesis se realiza mediante la aplicación de la transformada inversa IMDCT, que es descrita por la ecuación (3.2) [5]:

$$\tilde{x}_i[n] = w_s^i[n] \frac{4}{N} \sum_{k=0}^{\frac{N}{2}-1} X_i[k] cos\left[\frac{2\,\pi}{N}(n+n_o)\left(k+\frac{1}{2}\right)\right] \quad (3.2)$$

Para $n = 0, \ldots, N-1$

w_s^i = ventana de síntesis para el i-ésimo bloque

[14] Es el efecto que causa que señales continuas distintas se tornen indistinguibles cuando se les muestrea digitalmente.

Como se tiene en las ecuaciones (3.1) y (3.2) los términos $w_a^i[n]$ y $w_s^i[n]$ corresponden a las ventanas de análisis y síntesis, para que estas sean válidas y permiten la reconstrucción exacta de la señal es indispensable cumplir dos condiciones [5]:

1. Las ventanas de análisis y síntesis deben ser inversas en tiempo entre sí en la porción que se traslapan.
2. Cumplir la ecuación(3.3) donde:

   $n = 0, \ldots, \frac{N}{2} - 1$

   $i =$ Indica el número de bloque

$$w_a^i[n]w_s^i[n] + w_a^{i-1}\left[\frac{N}{2} + n\right]w_s^{i-1}\left[\frac{N}{2} + n\right] = 1 \qquad (3.3)$$

Para la fase de análisis y síntesis de la MDCT se utiliza la ventana de tipo senoidal ya que posee una mayor resolución en frecuencia, con lo que se puede tener una mejor apreciación de los sonidos cardiacos y respiratorios cuyas frecuencias son bajas. Además cumplen con las condiciones para que la ventana sea válida.

La ventana seno para señales de tiempo discreto de N muestras se puede implementar mediante la ecuación (3.4) [5]:

$$w_s[n] = sin\left[\frac{\pi\,(n + 1/2)}{N}\right] \qquad (3.4)$$

Para n = 0, … , N-1

Para reconstruir la señal original es indispensable que los bloques a los que se aplique la MDCT estén solapados en 50% para posteriormente sumar las señales generadas por la IMDCT, esta técnica es conocida como cancelación de aliasing en el dominio del tiempo.

Una forma sencilla y rápida de implementar la MDCT es utilizando la FFT[15], para lo cual se modifica la ecuación de la siguiente forma [5]:

$$X_i[k] = Re\left\{e^{-\frac{j\,2\,\pi}{N}n_o\left(k+\frac{1}{2}\right)}\sum_{n=0}^{N-1}\left[w_a^i[n]x_i[n]e^{-\frac{j\,2\,\pi}{2N}}\right]e^{-\frac{j\,2\,\pi}{N}}\right\} \qquad (3.5)$$

[15] Transformada Rápida de Fourier

Para obtener la ecuación (3.5) que permite obtener una transformada rápida de la MDCT se realizan los siguientes pasos [5]:

a. Multiplicar las muestras de la señal de entrada por el factor $e^{-\frac{j\,2\,\pi}{2N}}$
b. Aplicar una ventana de análisis valida de N puntos.
c. Realizar la FFT de N puntos con los datos obtenidos de la multiplicación anterior.
d. Evaluar los datos de la transformada para valores de k comprendidos entre 0 hasta N/2 -1, tomando la parte real de los datos de la transformada el factor de: $e^{-\frac{j\,2\,\pi}{N} n_o \left(k+\frac{1}{2}\right)}$

Para demostrar el procedimiento de análisis y el algoritmo rápido de la MDCT mediante la FFT, tomamos un sonido cardiaco de 768 muestras, Figura 3.3 y se divide esta señal en 4 bloques con N = 512 como se muestra en la Figura 3.4, además cada bloque está solapado un 50%.

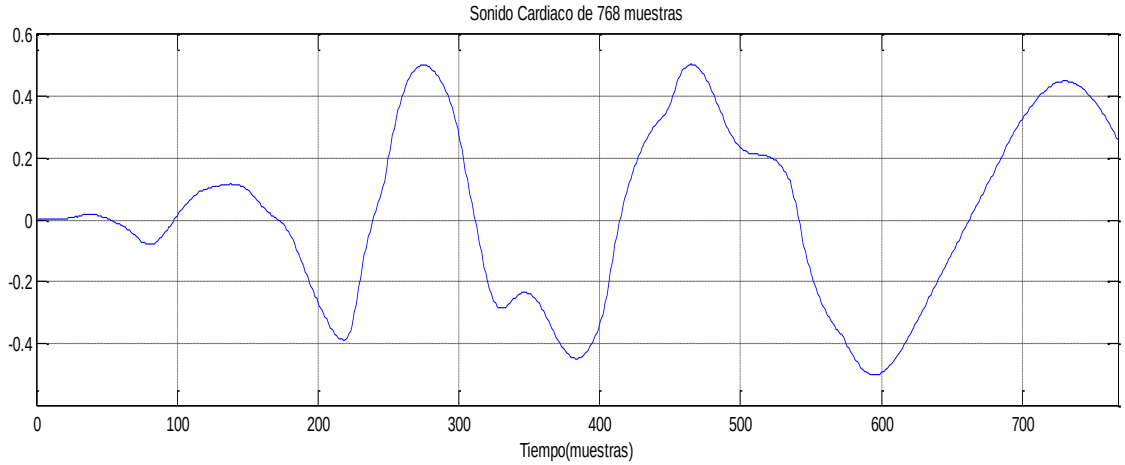


Figura 3.3 Señal Cardiaca de 768 muestras

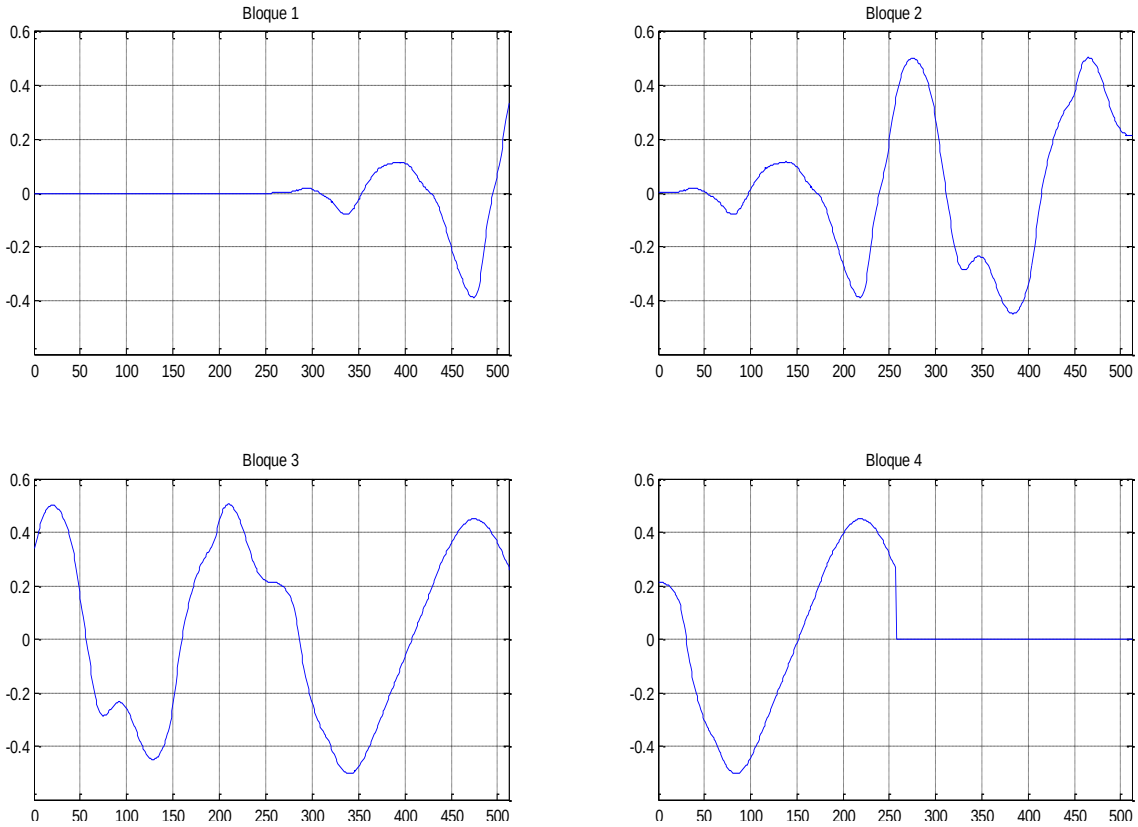


Figura 3.4 Bloques Solapados 50%

Para calcular la MDCT se utiliza la ecuación (3.5) y la ventana de análisis valida descrita en la ecuación (3.4), los cálculos realizados se encuentran en el anexo 1: **Función en Matlab para calcular la MDCT.**

En la Figura 3.5 se observa el espectro de cada bloque en función de la frecuencia, como se aprecia se tendría una reducción de coeficientes de un 50%, con lo cual se debe considera que con N muestras en dominio del tiempo aplicando la MDCT se obtiene N/2 muestras en el dominio de la frecuencia.

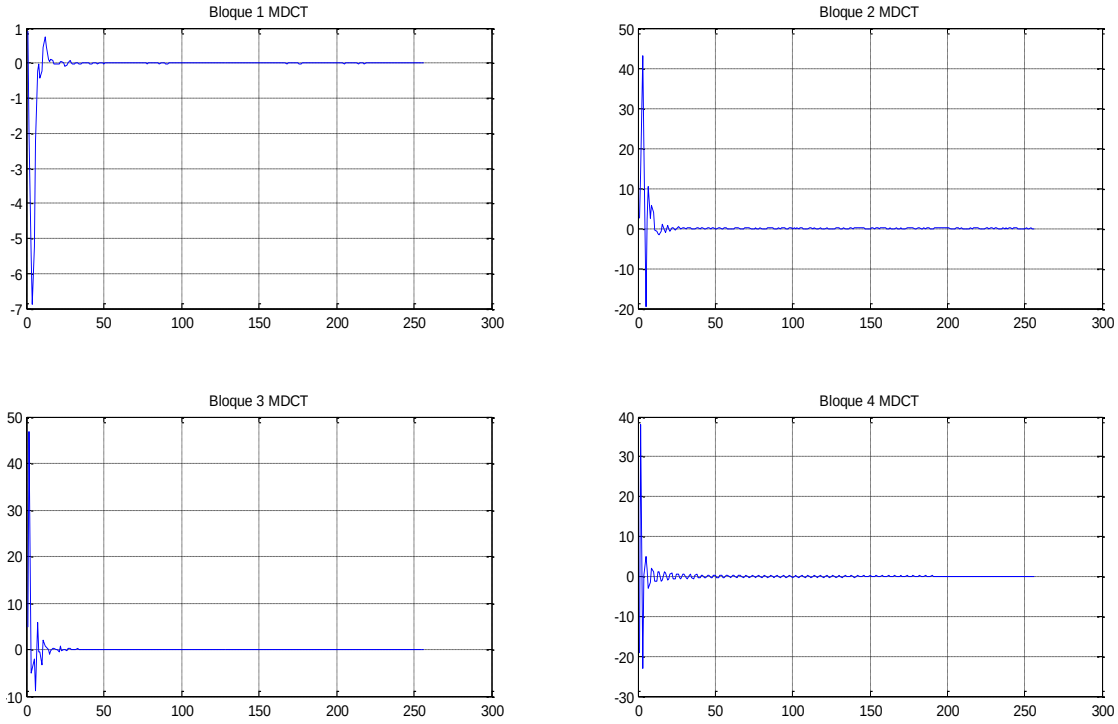


Figura 3.5 Espectro bloques con MDCT

La ecuación (3.6) permite calcular la IMDCT y este proceso se detalla en el anexo 1: **Función en Matlab para calcular la IMDCT**.

$$\tilde{x}_i[n] = 2\, w_s^i[n] Re\left\{ e^{\frac{j\,2\,\pi}{2\,N}(n+n_o)} \frac{1}{N} \sum_{k=0}^{N-1} [X_i[k]\, e^{\frac{j\,2\,\pi}{2} n_0 k}]\, e^{\frac{j\,2\,\pi}{2} n\, k} \right\} \qquad (3.6)$$

Para calcular la IMDCT se realizan los siguientes pasos [5]:

a. Multiplicar las muestras en frecuencia de la señal de entrada por el factor $e^{\frac{j\,2\,\pi}{N} k n_o}$, se debe tener en cuenta que el valor de k = 0,… , N-1, adicionalmente se utiliza X[N-1-k] = -X[k] para ≥ N/2 -1.
b. Realizar la IFTT[16] de N puntos con los datos obtenidos de la multiplicación anterior.
c. Evaluar los datos de la transformada, tomando la parte real de los datos de la transformada inversa el factor de: $e^{\frac{j\,2\,\pi}{2\,N}(n-n_o)}$ y luego multiplicar 2 veces la ventana de síntesis valida.

[16] Transformada Inversa Rápida de Fourier

Continuando con el proceso de síntesis del sonido cardiaco, se utiliza los bloques de la Figura 3.5 para demostrar el algoritmo de la IMDCT. Se aplica la IMDCT a cada bloque y se obtiene los bloques de la Figura 3.6 para reconstruir la señal original se suman cada una de las señales en la parte que se solaparon, esta técnica es conocida como cancelación de aliasing en el dominio del tiempo.

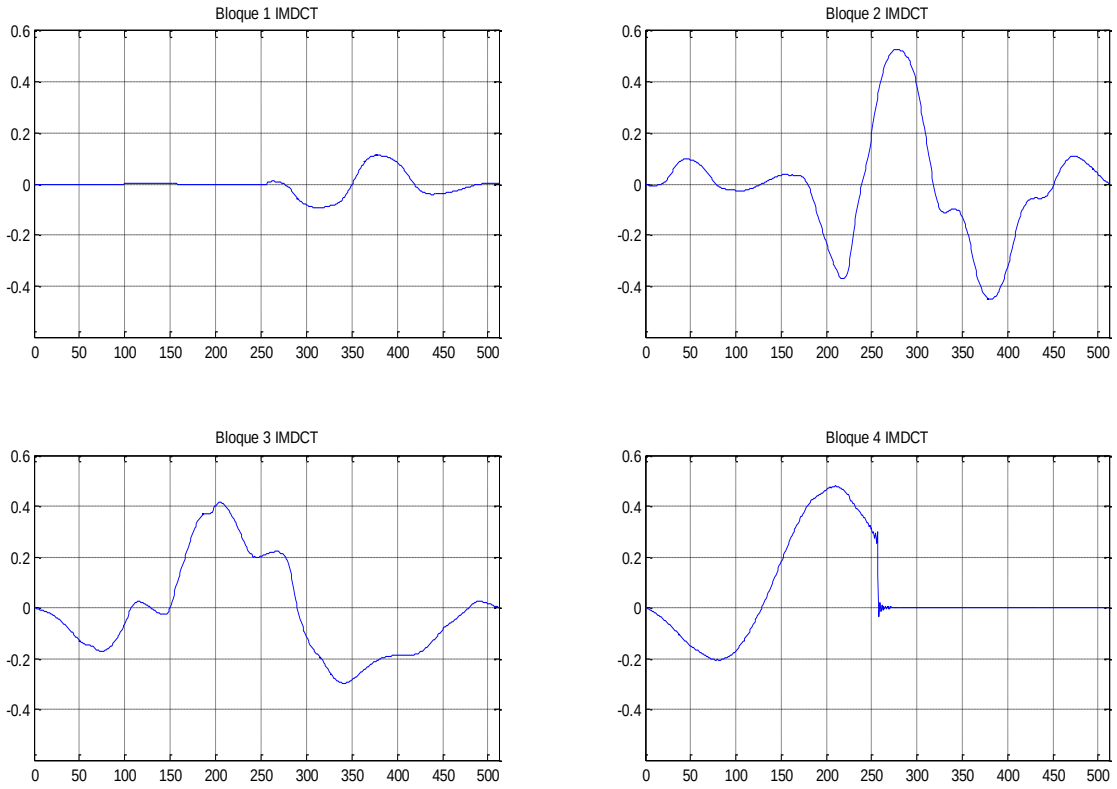


Figura 3.6 Bloques aplicando la IMDCT

3.3.1.2 Transformada discreta de Fourier

Antes de realizar el análisis psicoacústico se debe pasar del dominio del tiempo al de la frecuencia, para lo cual se utiliza la Transformada Discreta de Fourier. Esta se calcula aplicando primeramente una ventana de Hanning a las N muestras de la señal PCM, permite eliminar componentes frecuenciales no deseados. Luego se aplica la implementación rápida de la DFT como es la FFT, se utiliza esta transformada porque facilita el análisis en el dominio de la frecuencia.

3.3.1.3 Modelo Psicoacústico

El objetivo del modelo psicoacústico es obtener el umbral global de enmascaramiento para escoger la precisión con la que serán cuantizados los

coeficientes de la MDCT, mientras mayor sea la precisión menor será el error de la señal reconstruida pero si se utiliza muchos bits la tasa de compresión se verá reducida, al contrario, con poca precisión el error será mayor en la señal pero se logrará mayor compresión.

Al codificar en el dominio de la frecuencia se utiliza el modelo psicoacústico, para determina en que porciones del espectro el oído humano no será capaz de percibir el ruido de cuantización. Además la cantidad máxima de ruido que se pueda localizar en dicha frecuencia sin que esta sea percibida [5].

En este proceso se realizan diferentes operaciones, como transformar los coeficientes DFT a una escala decibel, posteriormente el espectro se convierte a la escala Bark y se calculan las máscaras individuales para obtener el umbral de enmascaramiento.

Para transformar a una escala decibel se determina el Nivel de Presión Sonora (SPL) de la señal de audio PCM. Para datos PCM de 16 bits, se asume que una sinusoide con una amplitud igual al nivel de sobrecarga del cuantizador tendrá un SPL de 96 dB $\left(\frac{6\ dB}{bit} x\ 16\ bits\right)$. Si se define el nivel de sobrecarga del cuantizador igual a 1, el nivel de presión de sonido de una sinusoide con una amplitud A es igual [5]:

$$SPL = 96\ dB + 10\log_{10}(A^2) \qquad (3.7)$$

Para calcular el nivel de presión sonora de la transformada discreta de Fourier se tiene [5]:

$$SPL_{DFT} = 96\ dB + 10\log_{10}\left(\frac{4}{N^2 < w^2 >}|X[k]|^2\right) \qquad (3.8)$$

Para la MDCT el nivel de presión sonora es [5]:

$$SPL_{MDCT} = 96\ dB + 10\log_{10}\left(\frac{8}{N^2 < w^2 >}|X[k]|^2\right) \qquad (3.9)$$

Donde:

$|X[k]|^2$ = Densidad de potencia espectral calculada de la señal de entrada.

$< w^2 >=$ Ganancia de la ventana.

$N =$ Número de muestras de la señal de entrada.

Se tiene que para una ventana seno el valor de la ganancia es 1/2 y para la ventana Hanning este valor es 3/8.

El SPL obtenido de las ecuaciones (3.7) y (3.8) está en Decibel, el siguiente paso es convertir los valores del espectro a Barks para lo cual se utilizará la ecuación (2.5), al cambiar a escala de Bark el espectro se divide en 25 bandas que corresponden a los 25 Barks descritos en la Tabla 2.2. De cada banda seleccionamos el elemento que posee mayor energía y este llegaría a ser el enmascarador, luego se calcula la máscara individual con la función de propagación propuesta en el ISO/IEC MPEG Psychoacoustic Model 2 [5]:

$$10\log_{10}\big(F(dz)\big) = 15.8111389 + 7.5\,(1.05\,dz + 0.474) - 17.5\sqrt{1 + (1.05\,dz + 0.474)^2} + 8MIN\big(0, (1.05\,dz - 0.5)^2 - 2(1.05\,dz - 0.5)\big) \qquad (3.10)$$

Donde $dz = z(f_{enmascarado}) - z(f_{enmascarador})$

Ahora debemos encontrar el umbral global de enmascaramiento, para lo cual es posible que se desee sumar, sobre-sumar o elegir el valor mayor de las máscaras en una frecuencia dada. Estas operaciones se pueden describir de acuerdo a la siguiente fórmula [5]:

$$I_N[f] = \left(\sum_{n=0}^{N-1} I_n^{\alpha}\,[f]\right)^{\frac{1}{\alpha}} \qquad (3.11)$$

Donde:

$I_N[f] =$ Intensidad de la curva de enmascaramiento resultante de combinar N curvas de enmascaramiento con intensidades $I_n[f]$ en una frecuencia f, ademas I_n es calculado con la función de enmascaramiento dada.

$\alpha =$ Define la forma con que se combinan las curvas de enmascaramiento individuales.

α = 1, la ecuacion equivale a sumar las intensidades.

$\alpha = +\infty$, equivale a utilizar el máximo de la curva de enmascaramiento.

$\alpha < 1$, sobre-suma, se obtiene una intensidad mayor que la suma de sus componentes.

Se sugiere un valor de $\alpha = 0.33$ [11] para enmascaradores de intensidad comparable, implicando que al combinar dos curvas de igual intensidad resulta una sola curva con una intensidad 8 veces mayor.

Después que se tiene la curva de enmascaramiento se combina con el umbral auditivo, ecuación (2.4), se selecciona el valor máximo de las 2 curvas para finalmente obtener el umbral global de enmascaramiento.

3.3.1.4 Cuantización

Cada una de las bandas de la escala Bark es cuantizada de acuerdo a la relación señal a máscara de la banda, que se determina mediante la ecuación [5]:

$$SMR_b = \max(b) - \min(mask_b) \quad (3.12)$$

Donde

$b =$ Banda

$mask_b =$ Porción del umbral global de enmascaramiento que corresponde a la banda b.

La asignación de bits se hace para cada banda, es decir cada banda es cuantizada con un cuantizador de tamaño diferente. Para calcular el número de bits necesarios para que el ruido de cuantización dentro de cada banda no sea percibido se calcula mediante la ecuación (3.13) [5]:

$$R_b = \frac{P}{K_p} + \frac{ln10}{20ln2}\left(SMR_b - \frac{1}{K_p}N_b SMR_b\right) \quad (3.13)$$

Donde:

$R_b =$ Tamaño de bits del cuantizador para la banda *b*.

$P =$ Número de bits disponibles sin factor de escalamiento.

$K_p =$ Número de coeficientes diferentes de cero.

$N_b =$ Número de coeficientes en la banda *b*.

Una vez que se conoce el tamaño en bits del cuantizador de cada banda se procede a hacer la cuantización, además como los valores que van a ser

almacenados son discretos, es necesario que el factor de escalamiento codifique utilizando pocos bits.

3.3.1.5 Codificación Huffman

Luego de completarse el proceso de cuantización con un modelo psicoacústico, la información se empaqueta en un flujo de bits específico ya que se ha concluido con éxito la codificación con pérdidas, es posible eliminar datos redundantes empleando una técnica de codificación sin pérdidas, en este caso se utilizará la codificación Huffman[17] para no perder la calidad de la señal obtenida en los bloques anteriores. Este tipo de codificación reserva códigos de tamaño menor para los caracteres o grupos de ellos, con más probabilidad de ocurrir, y asigna códigos más largos para aquellos menos frecuentes. Su empleo reduce el tamaño del código usado para representar los símbolos de un alfabeto.

3.3.2 Decodificador

El proceso de decodificación es mucho más simple que el de codificación debido a que el análisis de tipo psicoacústico se realiza en el codificador, además sólo se efectúa la lectura de los datos codificados y se aplica los procesos matemáticos y probabilísticos inversos, como se muestra en la Figura 3.7.

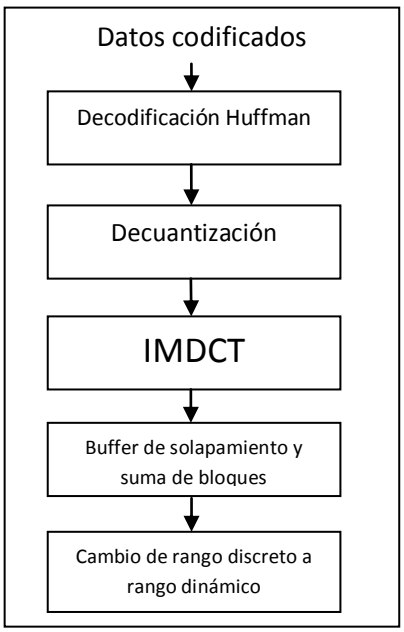


Figura 3.7 Esquema del decodificador[18]

[17] Véase Capitulo 2: "Codificación Huffman"
[18] Realizado en base a la recomendación UIT-R BS.1115

3.4 Selección del codec

El códec propuesto es de naturaleza perceptiva, en el mercado existen varios códecs de este tipo y con prestaciones similares debido a que sus algoritmos de codificación manejan herramientas como la MDCT, el enmascaramiento psicoacústico y codificación sin perdidas Huffman.

Los codecs que han sido considerados y que cumplen con los requerimientos propuestos son los siguientes:

- MP3 (MPEG Audio Layer 3)
- WMA (Windows media audio)
- AAC (Advanced audio coding)
- Ogg Vorbis

Aunque el formato ATRAC3 contiene las mismas características de los formatos nombrados no se considera debido a que es utilizado exclusivamente por dispositivos fabricados por la compañía SONY[19].

En la tabla 3.1 se contrastan las características de los diferentes codecs nombrados.

Tabla 3.1 Comparación formatos de audio[20]

Formato	Tasas de bits [Kbps]	Transformada	Codificación	Modos	Licencia
MP3	32 – 320	MDCT	Huffman	*CBR- **VBR	Privada
WMA	48 – 192	MDCT	Huffman	CBR	Privada
AAC	12 – 320	MDCT	Huffman	CBR	Privada
Ogg	8 – 512	MDCT	Huffman	CBR-VBR	Pública

*CBR: Tasa de Bits Constante

** VBR: Tasa de Bits Variable

[19] Ver: "Estandar de compresión de audio ATRAC3" documento compilado por Ing. Ivan Lobato, Universidad Central de Venezuela

[20] Tabla realizada en base a las especificaciones generales de los formatos nombrados, algunas características pueden variar dependiendo del software codificador

3.4.1 MP3

El MP3 fue creado por el Motion Pictures Expert Group (MPEG) en 1992, es un codec de audio con pérdidas de tipo psicoacústico, su popularidad creció debido a su extendida utilización en Internet. La utilización de este formato demanda que se solicite una licencia de utilización al instituto Fraunhofer. Contiene cuantificación no uniforme, codificación Huffman, soporta codificación con tasa de bits variable, y máximo dos canales de audio[21].

3.4.2 WMA

Es el formato por defecto del reproductor de Windows a lo que debe su utilización aunque menos extendida que MP3, se han creado nuevas versiones para poder competir con AAC y Ogg, WMA 8 y WMA 9 son los más populares aunque la versión más actual es WMA 10 que trabaja con hasta 5.1 canales de audio y se puede implementar sobre streaming, trabaja con frecuencias de muestreo de 44.1kHz y 96kHz[22].

3.4.3 AAC

Desarrollado por el Instituto Fraunhofer juntamente con AT&T, Nokia, Sony y Dolby y diseñado para reemplazar al MP3. Para un mismo número de impulsos por segundo (bit rate) y un mismo tamaño de archivo MP3, el formato AAC es más estable y tiene más calidad que MP3, maneja algunas extensiones como son (.m4a, .m4b, .m4p, .m4v, .m4r, .mp4, .3gp, y .aac), soporta hasta 48 canales, la frecuencia de muestreo alcanza resoluciones de hasta 96kHz y trabaja correctamente con tasas de bits bajas sin agregar ruido extra. Al igual que MP3 está vinculado al pago de una licencia por su utilización.[23]

3.4.4 Ogg Vorbis

Este formato se crea en respuesta a la demanda de codecs de compresión de licencia gratuita, especialmente para integrarse a sistemas operativos como Linux, soporta hasta 255 canales, las frecuencias de muestreo en que trabaja son 44.1 Khz y 48KHz y más de 16 bits de resolución, su código es abierto, razón por la cual

[21] http://www.mpeg3.com
[22] http://www.microsoft.com/windows/windowsmedia/windows_media_audio_contents.mht
[23] http://www.apple.com/quicktime/technologies/aac.mht

no es posible delinear sus características con exactitud ya que puede adaptarse a las necesidades de uso.

Su popularidad está en asenso principalmente porque su versión definitiva es reciente, y ya es soportado por los principales reproductores del mercado, además consta en los paquetes de codecs que se pueden instalar en Windows, Linux y en MAC OS[24].

3.5 Comparación de los formatos considerados

La comparación entre formatos de audio es complicada debido a que cada codec tiene características propias e indicadores de calidad diferentes, es decir, factores que indican a un oyente la calidad de una pista de audio, por ejemplo, se sabe que un archivo MP3 con calidad cercana a la de un CD tiene 128Kbps, generalmente se relaciona la velocidad de bits con la calidad de sonido no solo en MP3 sino en WMA y AAC, pero no ocurre lo mismo en Ogg donde la calidad de compresión se determina con una escala del 0 al 10, así un archivo Vorbis con calidad 0 correspondería a un MP3 de 64Kbps y la calidad 10 a uno de 499Kbps.

Esto indica que la calidad es subjetiva para cada persona y cada quien prefiere un tipo de codec diferente basándose en la popularidad del formato como sucede con MP3, tamaño del archivo, o por que posee hardware o software que manejan predeterminado formato por ejemplo los populares iPods que trabajan con .aac.

En el presente trabajo la comparación se realizará basándose en la observación del comportamiento de cada codec con relación al archivo original no comprimido en formato WAV. El archivo de prueba es una pista de 4 segundos que corresponde a un sonido cardiaco captado por un estetoscopio digital de la marca 3M[25].

3.5.1 Tamaño de los archivos

La señal original en .wav ha sido comprimida en los formatos WMA, AAC, MP3 y Ogg, el software utilizado para la codificación y la decodificación es el plugin de Winamp versión 5.552 [ml_transcode.dll] disponible en su página oficial.

[24] http://www.vorbis.com/faq/#what.mht
[25] Enlace del archivo: http://solutions.3mchile.cl/wps/portal/3M/es_CL/Littmann-WW/stethoscope/

Los archivos se han establecido en CBR (tasa constante de bits) de 64Kbps para WMA, MP3 y AAC y con un solo canal (mono), Ogg se ha establecido en calidad 10 debido a que produce archivos cercanos a 64Kbps.

Pista 1: Sonido Cardiaco (S1 y S2).

Tabla 3.2 Comparación de tamaño de distintos formatos Calidad 64Kbps

Formato	Velocidad [Kbps]	Duración [s]	Tamaño [KB]	Porcentaje de compresión[26]
.WAV (archivo original)	176	4	97	--------
.WMA	64	4	50	48.45%
.AAC	64	4	39	59.79%
.MP3	64	4	36	62.89%
.Ogg	64	4	30	69.07%

Tabla 3.3 Comparación de tamaño de distintos formatos Calidad 48Kbps

Formato	Velocidad [Kbps]	Duración [s]	Tamaño [KB]	Porcentaje de compresión
.WAV (archivo original)	176	4	97	--------
.WMA	48	4	43	55.67%
.AAC	48	4	30	69.07%%
.MP3	48	4	30	69.07%
.Ogg	48	4	28	71.13%

Pista 2: Sonido respiratorio (estridor)

La segunda pista corresponde a un sonido respiratorio de cuatro segundos de duración y codificada con el proceso anterior.

Tabla 3.4 Comparación de tamaño de distintos formatos Calidad 64 Kbps de estridor

Formato	Velocidad [Kbps]	Duración [s]	Tamaño [KB]	Porcentaje de compresión
.WAV (archivo original)	176	14	110	--------
.WMA	64	14	126	--------
.AAC	64	14	67	39.9%
.MP3	64	14	115	--------
.Ogg	64	14	65	40.9%

[26] Tasa de compresión $=100-\left(\frac{Tamaño\ archivo\ comprimido[KB]}{Tamaño\ archivo\ original[KB]}*100\right)$

3.5.2 Comparación temporal.

A continuación se realizará una comparación en el dominio del tiempo entre dos señales, la señal original en .wav y la señal comprimida, cabe recalcar que esta última es transformada de nuevo a .wav para observar posibles variaciones sufridas en el proceso de compresión[27].

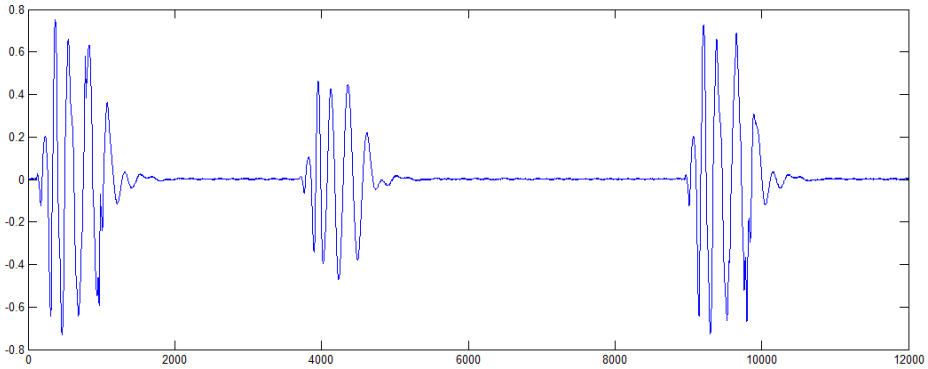


Figura 3.8 Señal original

Las señales utilizadas para la prueba corresponden a la Tabla 3.1

3.5.2.1 MP3

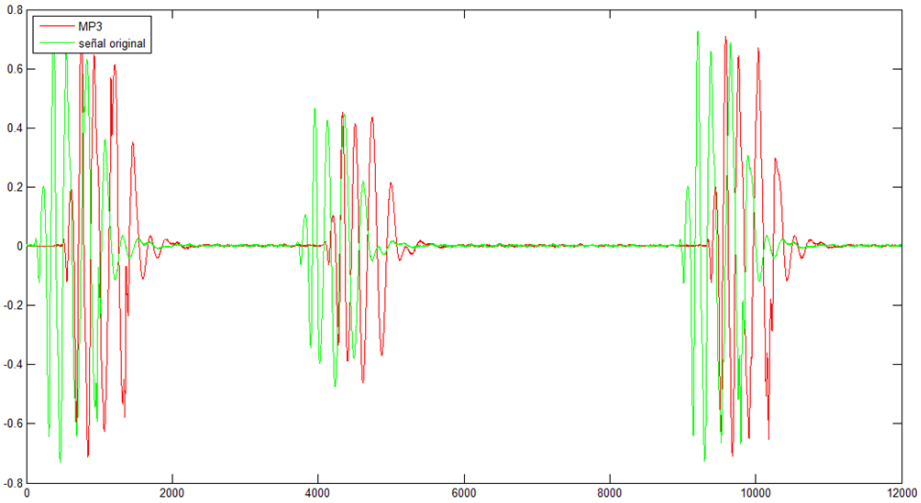


Figura 3.9 Comparación con señal MP3 reconstruida

[27] Las gráficas mostradas en esta sección han sido realizadas en Matlab versión 2007, en el anexo 1 se describe el código utilizado para este efecto.

La señal en rojo es la señal MP3 codificada de nuevo en .wav y se observa un retraso de 400 muestras pero con una reconstrucción casi exacta con respecto a la original.

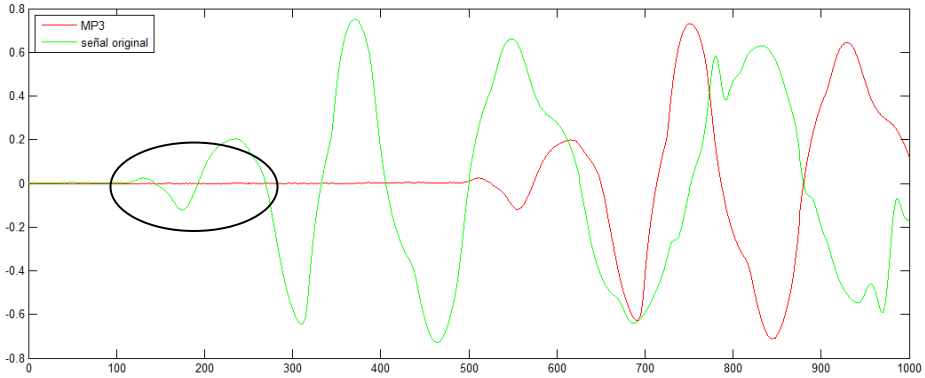


Figura 3.10 Inicio de la pista

En la Figura 3.10 se puede observar el principio de la pista, en el circulo se indica los efectos que genera la codificación (en rojo) al inicio de la señal, sin embargo este ruido es imperceptible.

3.5.2.2 HE- AAC

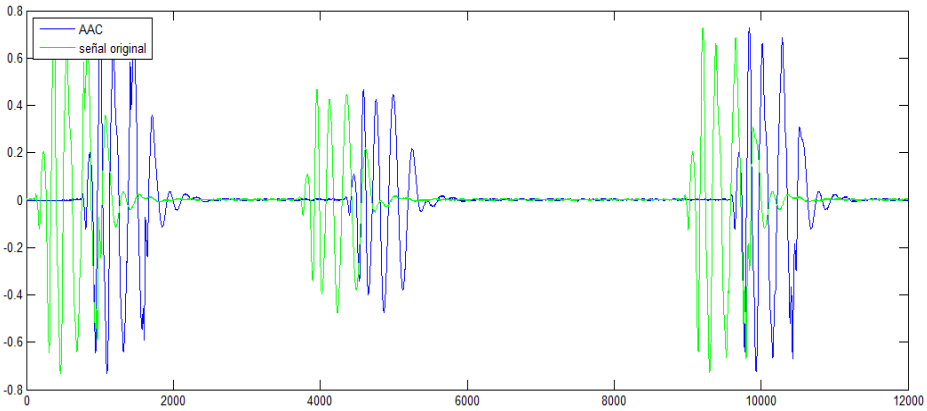


Figura 3.11 Comparación con señal AAC reconstruida

En este caso el retraso de la señal en AAC que se ve en azul es de 464 muestras, la reconstrucción es muy aproximada a la original y al principio de la pista no se agrega ruido.

3.5.2.3 WMA versión 10

En la Figura 3.12 no se observa ningún retraso en la señal y una reconstrucción casi perfecta de la señal.

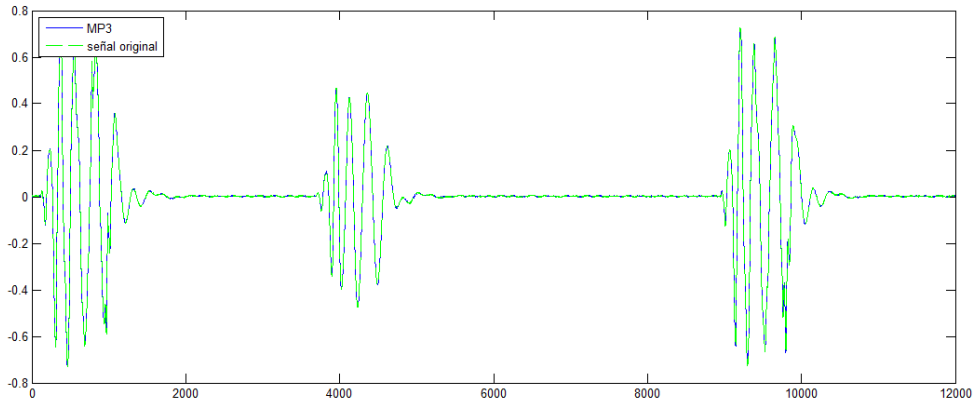


Figura 3.12 Comparación con WMA

3.5.2.4 Ogg

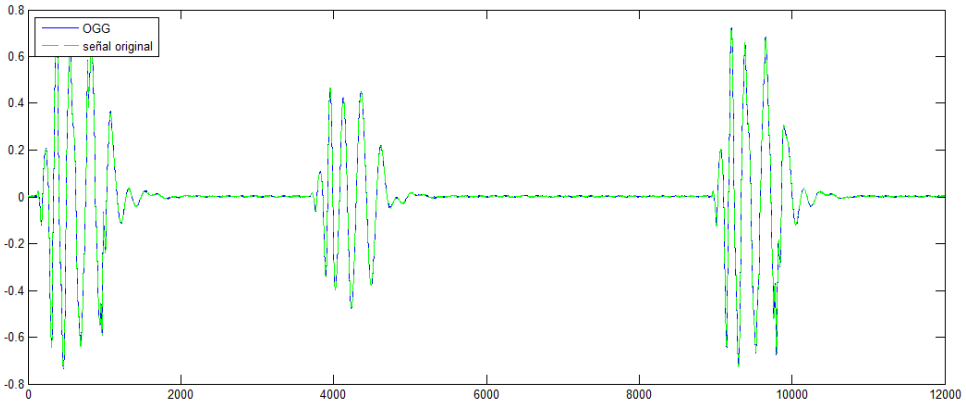


Figura 3.13 Comparación con Ogg reconstruido

Al igual que con WMA Vorbis no presenta retraso en la señal y la reconstrucción es exacta.

Similares resultados se observan con tasas de bits de 48Kbps Figura 3.14. Nótese que las señales Ogg y WMA no se distinguen debido a que están solapadas con la señal original.

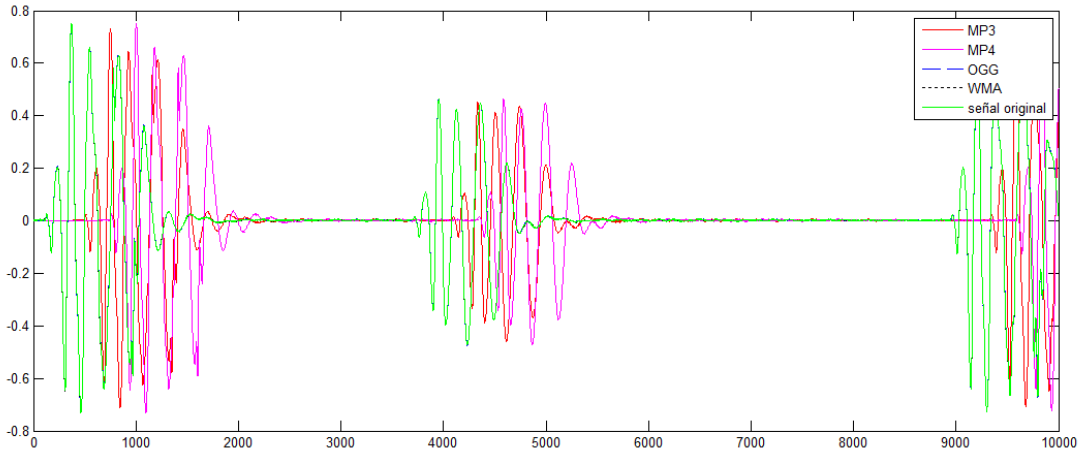


Figura 3.14 Señales comprimidas comparadas con la señal original.

3.5.3 Comparación en frecuencia

En la Figura 3.15 se observa el espectro que producen las señales comprimidas y la señal original.

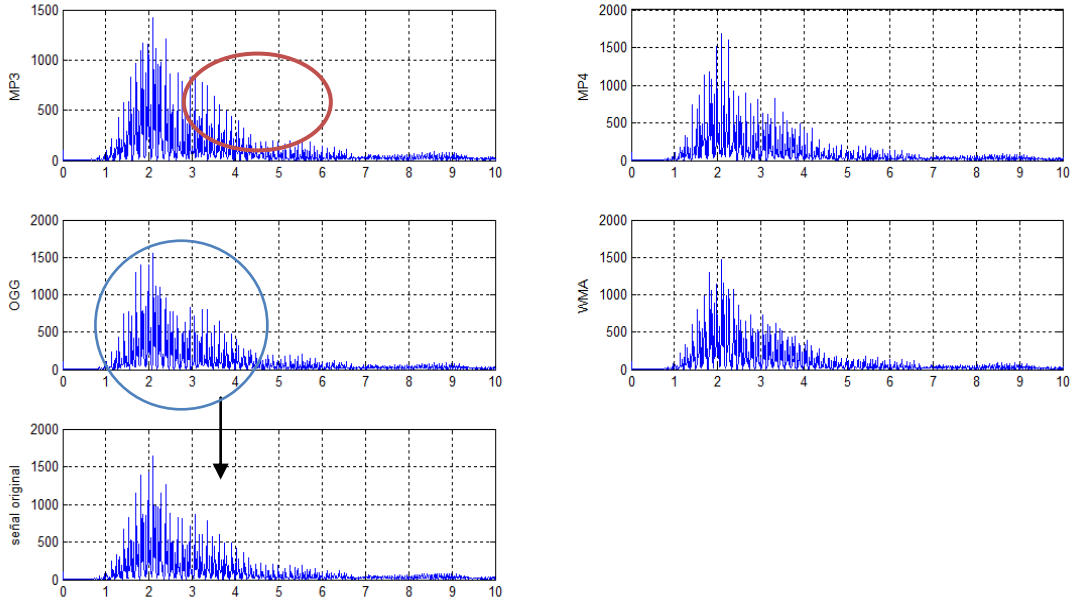


Figura 3.15 Comparación frecuencial de las señales comprimidas con la señal original

Como se observa en la figura superior las mejores aproximaciones en cuanto a espectro se logran con Ogg vorbis, AAC o MP4 y WMA, en el caso del MP3 se

aprecia un incremento en la energía de algunos componentes frecuenciales (circulo magenta), Ogg permite obtener reconstrucciones espectrales más precisas debido al enventanamiento de varias longitudes para la transformada MDCT.

3.5.4 Comparación en fase

Otra importante característica para tomar en cuenta es la respuesta en fase que se obtiene con los distintos formatos de compresión, en la Figura 3.16 se aprecia esta característica contrastándose los cuatro codecs en cuestión con la señal original.

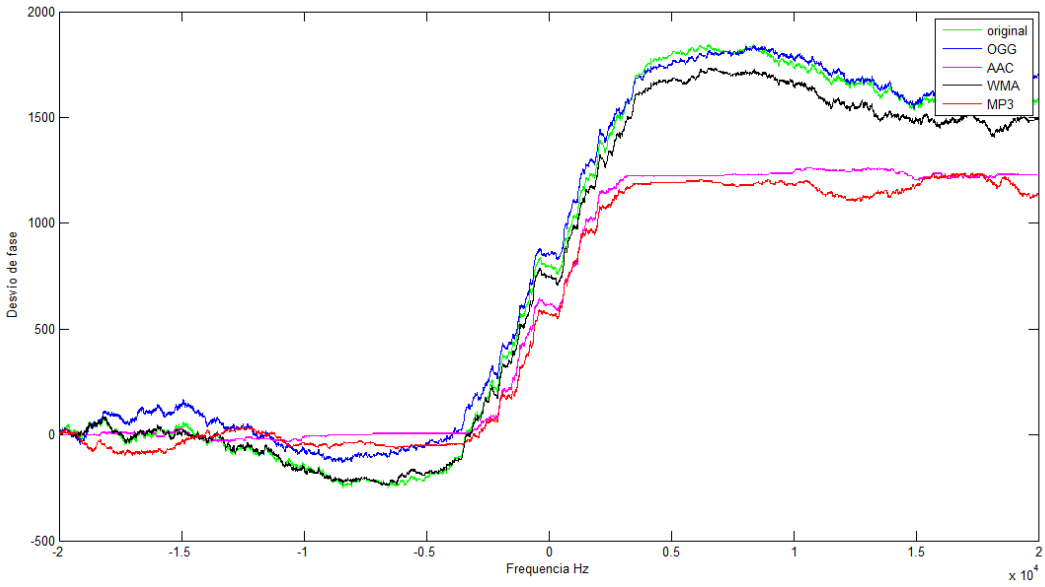


Figura 3.16 Desvío de fase de los codecs de compresión

Todos los codecs agregan una desviación de fase con respecto a la señal original pero según se observa los códecs MP3 y AAC tienen un desvío mucho más amplio que los formatos Ogg y WMA.

Según las características de los diferentes formatos de codificación se considera que AAC y Ogg son los que proveen mejores prestaciones debido a que el sistema de transmisión en tiempo real demanda que se utilicen bajas tasas de bits menores a 48 Kbps sin deformar la señal estetoscópica, en cuanto a calidad ambos

demuestran superioridad con respecto a MP3 en estudios de campo realizados en los últimos años[28] especialmente Ogg Vorbis.

En cuanto a la reconstrucción exacta de la señal se destacan Ogg y WMA, lo que provee seguridad y confianza al sistema donde se implementará el codec, se puede decir que en este caso existe un relativo empate entre los dos formatos de compresión.

Considerando el tamaño del archivo comprimido el que logra una mayor compresión es Ogg, con tasas mayores al 70% en la calidad más baja, es una gran ventaja sobre los demás ya que el objetivo es mejorar la calidad de los sonidos disminuyendo su tamaño para la transmisión.

Se considera que este último es el más conveniente para el sistema de transmisión debido a que su licencia es de libre uso y sobre todo a que el código es abierto y se puede modificar acorde a las necesidades que surjan en el proceso, sin las ataduras de los codecs comerciales.

[28] Véase www.rjamorim.com/test/index.html y (AVILA 2005)

4 Análisis del códec Ogg Vorbis

4.1 Introducción

Vorbis es un proyecto de la fundación Xiph.org fundada por Christopher Montgomery, forma parte de otros proyectos de código abierto como Theora para video, FLAC que es un códec de compresión de audio sin pérdidas, Speex diseñado para voz, y Ogg que consiste en un formato de almacenamiento capaz de albergar audio y video[29].

En el presente capítulo se describe el códec de audio Vorbis el cual es embebido en una trama Ogg, de ahí que la extensión de los archivos de este tipo sea [.ogg]. Vorbis es un códec de uso general de tipo perceptivo y de código abierto, libre de patente y está basado en la licencia pública GNU[30], su calidad es comparable con codecs de audio de última generación como MP3Pro, WMA V8 y AAC, además soporta tasas de bits tan bajas como 16Kbps, aunque teóricamente puede llegar hasta 8Kbps[31].

Su ventaja sobre otros formatos como el MP3 radica en que Vorbis además de ofrecer una mejor calidad a bajas tasas de bit, es completamente gratuito, su código está abierto a posibles modificaciones y mejoras por parte de cualquier programador, esto implica que su aplicación no se limita a comprimir música, sino a adaptarse a diferentes necesidades, por ejemplo, el codificador de Vorbis puede modificarse para comprimir eficientemente sonidos estetoscópicos que tienen particularidades en comparación con sonidos musicales utilizando técnicas de transformación como wavelet.

Sin embargo Vorbis aún es un formato desconocido por la mayoría del público pero esto se debe a que es relativamente nuevo en este mercado en comparación con el MP3, pero es actualmente soportado por varias aplicaciones como: icecast, winamp, Realnetworks, Mozilla firefox.

---

[29] Más información sobre los proyectos de Xiph: http://www.xiph.org

[30] GNU es el nombre que recibe un proyecto que data del año 1984 cuyo objetivo era el desarrollo de un sistema operativo basado en Unix y con la calificación de software libre. Estos sistemas son hoy en día muy usados bajo el nombre de Linux.

[31] http://vorbis.com/documentation.mht

4.2 Codificador

Ogg Vorbis sólo define su decodificador, esto quiere decir que cualquier codificador que produzca una trama decodificable por Vorbis es considerado un codificador Vorbis, esto permite que se implementen mejoras o cambios en el codificador sin cambiar el decodificador. A continuación se define el proceso para codificar una trama de audio [27].

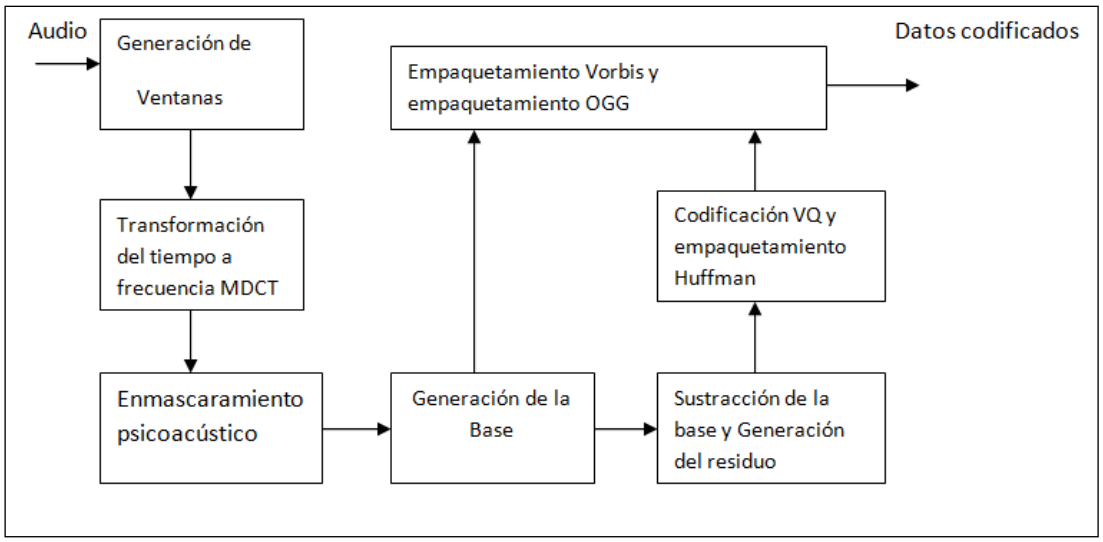


Figura 4.1 Codificación Ogg Vorbis[32]

4.2.1 Generación de ventanas

La trama de audio entrante PCM es dividida en bloques llamados ventanas, esto se hace para efectos de reducción del pre-eco producido por la posterior transformación de dominio con MDCT, la extensión de las ventanas deben ser una potencia de dos entre 64 y 8192 muestras, generalmente se utiliza dos tamaños comunes de ventanas, las grandes con 2048 y las pequeñas con 256, la razón de utilizar varios tamaños de ventanas se debe a que el uso de ventanas pequeñas reduce el efecto del pre-eco y las ventanas grandes permiten una mejor resolución del espectro, estas a su vez son creadas con un solapamiento del 50% para cumplir con los requerimientos de la MDCT como se muestra en la Figura 4.2. [27]

[32] Esquema realizado en base a la descripción de un codificador Vorbis general descrito en la especificación Vorbis 1

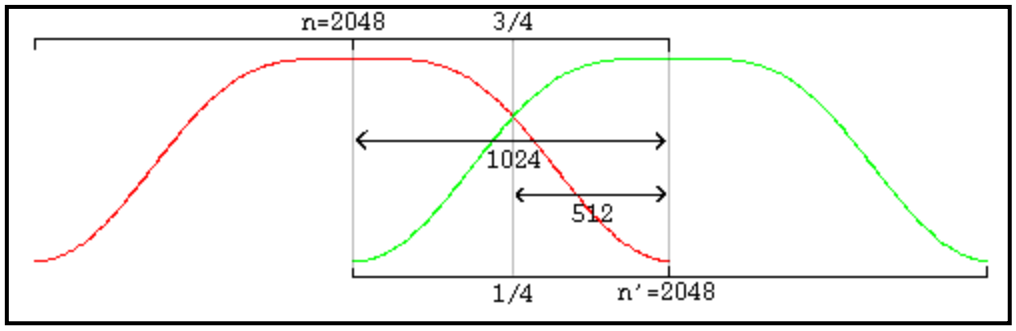


Figura 4.2 Solapamiento de dos ventanas con la misma extensión[33]

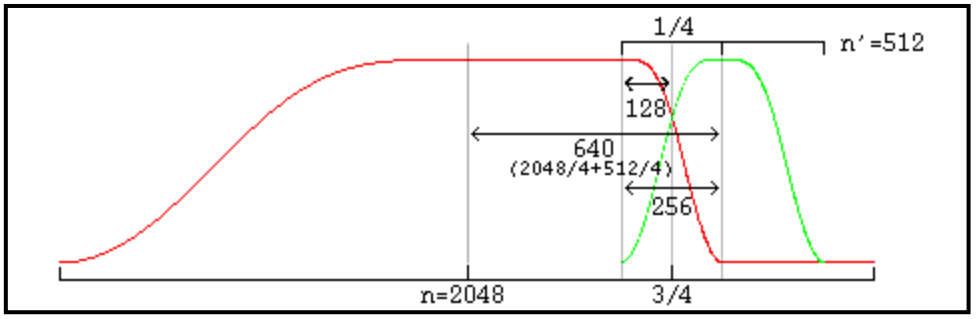


Figura 4.3 Solapamiento de una ventana grande y una pequeña[33]

Vorbis utiliza el tipo de ventana descrito en la ecuación 4.1 para el solapamiento [27].

$$y = \operatorname{sen}\left(\frac{\pi}{2}.\operatorname{sen}^2\left(\frac{x+0.5}{n}\pi\right)\right) \quad (4.1)$$

4.2.2 Transformación de dominio MDCT

Debido a que el estándar del codificador en Vorbis I es abierto, es posible utilizar cualquier tipo de transformada, en el caso de la primera versión de Vorbis se utiliza la MDCT.

4.2.3 Enmascaramiento Psicoacústico.

Vorbis I utiliza el modelo psicoacústico humano para descartar información no audible, característica que lo ubica en el grupo de codec con pérdidas, pero a diferencia de los otros codecs que utilizan un modelo con un volumen fijo durante la duración de la trama de audio, Vorbis asume que el volumen se ajusta dinámicamente con un máximo situado en el umbral del dolor [27].

[33] Tomado de la especificación Vorbis 1

4.2.4 Generación de la Base

La base, llamada floor por Xiph.org, es una versión del espectro de la señal de baja resolución, la señal base puede ser de dos tipos, Floor 0 que utiliza LSP (line spectral pair), que permite lograr un envolvente suavizado de la señal original en dB y en la escala Bark, y Floor 1 que utiliza una función lineal a trozos que codifica la envolvente del espectro, la cual se representa en un eje una frecuencia lineal y en el otro eje una escala logarítmica [27].

4.2.5 Generación del residuo

El residuo es el producto de restar la base a la señal espectral del audio, se codifica utilizando el vector de cuantización y se empaqueta con Huffman, los códigos resultantes de esta operación se empaquetan en la trama Ogg en la cabecera de configuración, esta característica diferencia a Vorbis debido a que los códigos no residen en el decodificador o en una ROM, lo que agrega kilobytes extra al archivo final y una demora de un segundo al inicio de la decodificación [27].

4.2.6 Empaquetamiento Vorbis

Finalmente se agrega la información codificada a la trama Vorbis Fig 4.4, en la cabecera de identificación se incluye el número de versión de Vorbis, el número de canales de audio y la tasa de bits, luego en la cabecera de comentarios se ubica información como autoría del archivo, organización, fecha, lugar, entre otros, en la cabecera de configuración yacen los códigos, la señal base, los residuos y el modo que indica el tipo de ventanas y transformada, por último se agrega los paquetes de audio [27].

Los paquetes Vorbis son de longitud no definida, así un típico paquete Vorbis varía de 2–3 bytes a 8-12 kilobytes. Cada paquete comienza con una cabecera de dos octetos que es utilizada para representar el tamaño en bytes de los datos siguientes. Luego este paquete vorbis se encapsula en una trama Ogg para su utilización multimedia[34].

En el caso de que Vorbis se utilice sobre RTP la encapsulación en Ogg que provee sincronización es innecesaria, en este caso se usan paquetes Vorbis directamente en la carga útil[35].

[34] Ver RFC 3533
[35] Ver RFC 5215

4.3 Decodificación

El trabajo del decodificador es identificar correctamente las cabeceras y procesarlas, a continuación se describe la forma en que se decodifica estos paquetes, Figura 4.4.

Identificación	Comentarios	Configuración	Audio

Figura 4.4 Trama Ogg[36]

Todas las cabeceras inician con un número de identificación de acuerdo al tipo de paquete, la cabecera de identificación es “tipo 1”, la cabecera de comentarios es “tipo 3” y la de configuración es “tipo 5”, estos números son impares debido a que un paquete con un número menos significativo 0 es un paquete de audio [27].

4.3.1 Cabecera de identificación

La cabecera de identificación contiene campos que son usados para declarar la trama como Vorbis y proveer información acerca de la trama de audio como el número de canales, la tasa de bits (máxima, nominal y mínima), la longitud de cada campo se define a continuación [27]:

1) [vorbis_version] = 32 bits enteros sin signo, versión de Vorbis.
2) [audio_channels] = 8 bits enteros sin signo, número de canales de audio.
3) [audio_sample_rate] = 32 bits enteros sin signo, tasa de muestreo
4) [bitrate_maximum] = 32 bits enteros con signo, máxima tasa de bits
5) [bitrate_nominal] = 32 bits enteros con signo, tasa de bits normal
6) [bitrate_minimum] = 32 bits enteros con signo, mínima tasa de bits
7) [framing_flag] = 1 bit, bandera señalando el fin de la cabecera

4.3.2 Cabecera de comentarios

Esta cabecera consta de ocho campos de 32 bits cada uno, es utilizada para proveer información sobre el archivo de audio como el autor, nombre del archivo, año de producción, etc., estos campos están abiertos para que se ubique cualquier información que se crea conveniente, a continuación se describe el proceso de decodificación de esta cabecera [27].

1) [vendor_length] = 32 bits enteros sin signo, longitud del campo del nombre del proveedor.

[36] Esquema basado en la descripción de la especificación Vorbis 1

2) [vendor_string] = leer vector UTF-8, nombre o comentario del proveedor o creador.
3) [user_comment_list_length] = 32 bits enteros sin signo, espacio para comentarios
4) [user_comment_list_length] veces {
5) [length] = leer 32 bits enteros sin signo
6) [comment] = leer un vector UTF-8 }
7) [framing_flag]= 1 bit, bandera de fin de trama.

4.3.3 Cabecera de configuración

La cabecera de configuración contiene toda la información necesaria para el proceso de decodificación, esta cabecera contiene lo siguiente: la lista de códigos, configuraciones de la transformada, configuraciones de la base, configuraciones del residuo, y configuraciones de los canales y del modo [27].

En este caso los campos son de longitud variable, así el campo de los códigos contiene ocho bits sin signo por paquete, estos se almacenan en una matriz llamada [vorbis_codebook_configurations].
El campo de la base o floor [vorbis_floor_types] contiene 16 bits enteros sin signo, indicando el tipo de base a decodificar (0 o 1), esta configuración se almacena en un campo llamado [vorbis_floor_configurations], si el campo lee un número mayor a uno la trama no es decodificable.
El campo del residuo [vorbis_residue_types] indica el tipo de residuo a decodificar tal como el caso anterior [vorbis_residue_configurations] almacena las configuraciones.
El campo modo contiene lo siguiente:
a) [vorbis_mode_blockflag] = 1 bit, es la bandera de inicio
b) [vorbis_mode_windowtype] = 16 bits, indica el tipo de ventana
c) [vorbis_mode_transformtype] = 16 bits, indica el tipo de transformada

4.3.3.1 Proceso de decodificación.

Las tramas se decodifican a través del siguiente proceso, la figura 4.5 muestra la síntesis del proceso.

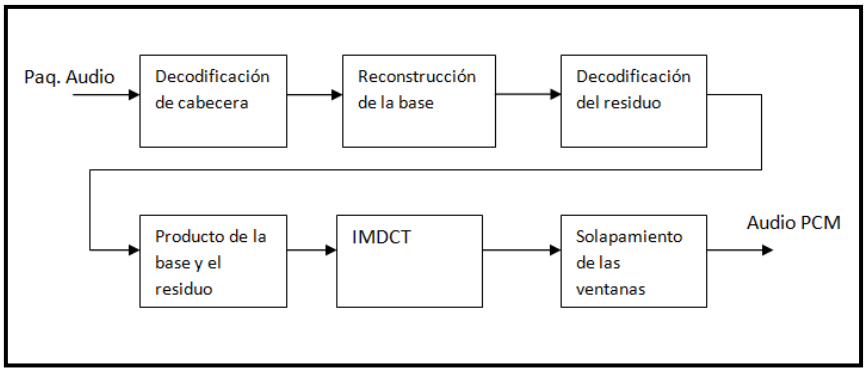


Figura 4.5 Proceso de decodificación[37]

1. Decodificación de cabecera: En este bloque se decodifica la información necesaria para el proceso de reconstrucción del audio, se realiza los siguientes pasos:
    - Decodificación de la bandera que señala el tipo de paquete.
    - Decodificación del número de modo.
    - Decodificación del tipo de ventana.
2. Reconstrucción de la base: Se decodifica los vectores correspondientes a la base y se la reconstruye a través del algoritmo de Bresenham.
3. Decodificación de los residuos en vectores de residuos.
4. Cálculo del producto de la base y de los residuos generando un vector del espectro del audio.
5. Transformada inversa monolítica del vector del espectro de audio, siempre de tipo MDCT en Vorbis I.
6. Solapamiento de las ventanas: Finalmente se realizan los siguientes pasos:
    - Solapar/adicionar la salida de parte izquierda de los bloques de la transformada con la salida de la parte derecha del anterior bloque.
    - Guardar los datos de la parte derecha del bloque actual para el siguiente solapamiento.
    - Si no es el primer bloque, devuelve los resultados del proceso de solapar/adicionar como audio resultante del bloque actual.

4.3.3.2 Códigos

Vorbis no tiene un modelo probabilístico estático, es así que se debe empaquetar toda la configuración de decodificación para descifrar la información codificada con VQ y Huffman, en la cabecera de configuración [27].

[37] Realizado en base a la especificación Vorbis I

4.3.3.3 Base

Vorbis usa dos tipos de base llamadas floor type 0 y floor type 1, en este campo se debe especificar el tipo de base aunque en Vorbis 1 se utiliza el tipo 1 [27].

Floor 0 utiliza LSP (Line Spectral Pair) codifica la envolvente del espectro como la respuesta en frecuencia de un filtro LSP. El filtro LSP es una representación en el dominio de la frecuencia de un filtro IIR. Esto es equivalente a un LPC (codificación predictiva lineal) y puede ser convertida a dicha representación. La decodificación de la curva base se realiza en dos etapas. Primero se extraen la amplitud de la curva y los coeficientes del filtro del flujo de bits, y después la curva base se calcula como está definida la respuesta en frecuencia de un filtro LSP decodificado. Este tipo de base no se utiliza actualmente, debido a las pobres prestaciones que obtiene y a la alta complejidad del decodificador comparado con la función floor1. La función Floor 0 fue reemplazada en una versión beta temprana, pero forma todavía parte del estándar ya que debe ser contemplada por cualquier decodificador [27].

Floor 1 codifica una curva espectral en una serie de segmentos lineales, la curva se construye usando predicción iterativa. Para reconstruirla se dibuja una línea entre el punto de inicio y de fin, y después de forma iterativa se van cambiando una serie de valores. El algoritmo para dibujar líneas de Bresenham[38] es el que se utiliza para realizar la interpolación de la línea. Los coeficientes se codifican a través de Huffman. En el documento de la especificación Vorbis se muestra el siguiente ejemplo [27]:

Se asume una configuración de base con n = 128, los valores en orden ascendente es 0,16,32,48,64,80,96,112 y 128 en X.

Los valores tal como se decodifican de un paquete son:

Y_ = 110, 20, -5, -45, 0, -25, -10, 30, -10

X_ = 0, 128, 64, 32, 96, 16, 48, 80, 112

El primer segmento se traza desde X_0, Y_0 y X_1,Y_1 es decir desde [0,110] hasta [128,20].

[38] Información detallada véase (Weitzenfeld).

A continuación se genera un nuevo punto en X_3 [64], corrigiendo el valor original en la recta con el valor de Y_3 [-5], como se observa en la Figura 4.6.

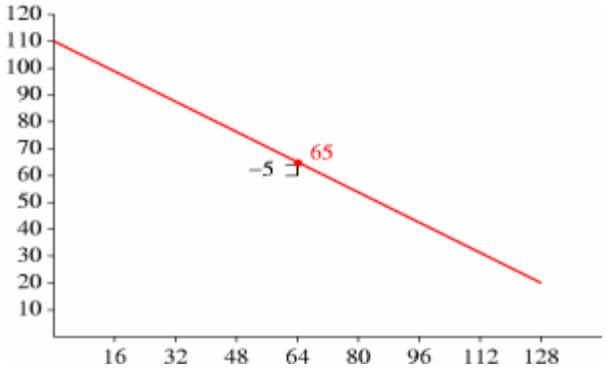


Figura 4.6 Reconstrucción de la base[39]

Ahora se reflejan nuevas líneas lógicas para reflejar la corrección de la nueva posición de Y, y se prosigue con X_4 y con X_5 y se corrige el valor de Y con Y_4 y Y_5 respectivamente, Figura 4.7:

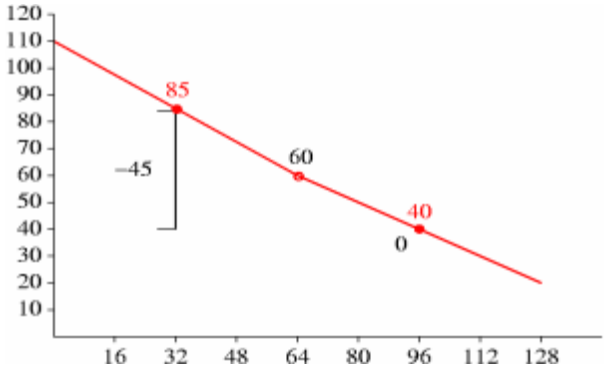


Figura 4.7 Reconstrucción de la base

A pesar de que el nuevo valor de Y a la posición X_5 no cambia es usada posteriormente para afinar la curva, se completa la curva como sigue:

[39] Tomado de Especificación Vorbis 1

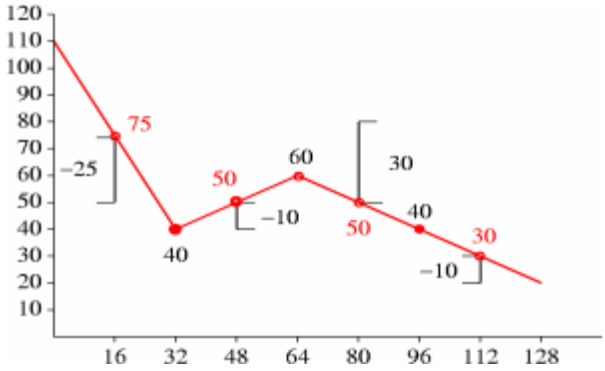


Figura 4.8 Reconstrucción de la base

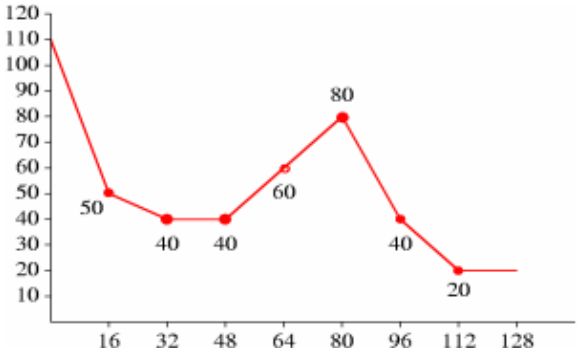


Figura 4.9 Reconstrucción final de la base

Vorbis actualmente utiliza algoritmos más eficientes que permiten que se redondee la curva para lograr una curva más limpia además se maneja de mejor manera el ruido producido por la acción del redondeo y el truncamiento.

4.3.3.4 Residuos

Un residuo representa el fino detalle de un espectro de audio luego que la curva base es restada de la señal espectral inicial, un vector residuo puede representar líneas espectrales, magnitudes espectrales o fase espectral, Vorbis hace uso de tres tipos de variantes de codificación. Los residuos son codificados mediante VQ de acuerdo a uno de los tres posibles modos de empaquetamiento, los cuales difieren en como los valores son intercalados. Aunque los vectores deben ser codificados de forma monolítica, a menudo son codificados como una suma aditiva de varias pasadas sobre el vector de residuos utilizando más de una codificación VQ con el objetivo de permitir un diseño eficiente de los códigos. Los códigos VQ tiene el mismo formato que los códigos para la base, y algunos códigos se utilizan

en ambas etapas. La decodificación con varias pasadas es la razón principal de la alta complejidad en tiempo de esta etapa [27].

4.3.3.5 Reconstrucción del espectro.

La reconstrucción se logra a través del producto entre el vector residuo y la base, luego de que la curva de la base haya sido reconstruida se añade a esta curva los residuos, lo que permite que el espectro se asemeje ya al espectro original, la figura 4.12 muestra este proceso.

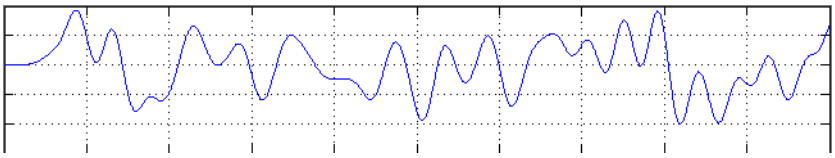


Figura 4.10 Señal base reconstruida

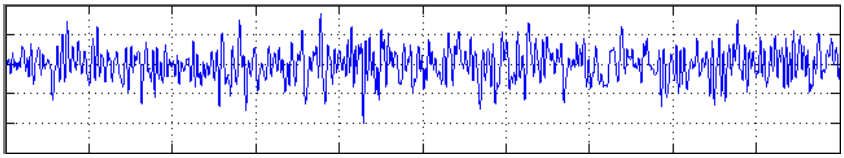


Figura 4.11 Vector residuo

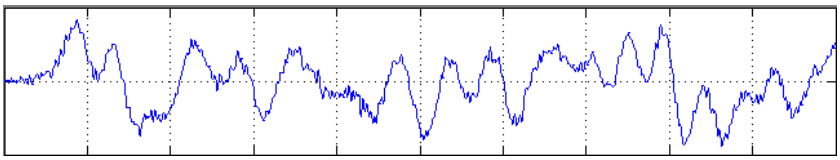


Figura 4.12 Espectro reconstruido

Vorbis realiza este proceso en cada canal de audio que exista, multiplicando elemento por elemento.

4.3.3.6 Reconstrucción del audio

El espectro de audio es transformado al dominio del tiempo en PCM mediante la IMDCT, la señal PCM producida por la IMDCT no es todavía el audio final, la señal aún está dividida en bloques producto de la operación del enventanamiento, estos bloques deben ser solapados de nuevo entre ellos para lograr la reconstrucción correcta del audio. Cabe recalcar que este proceso demanda la utilización de un buffer para preservar los datos de la parte derecha de los bloques para que se pueda solapar con la parte izquierda de los nuevos bloques entrantes. Este proceso se puede observar con detalle en la sección 3.3.1.

5 Transmisión de audio en tiempo real sobre redes IP.

5.1 Introducción

La transmisión en tiempo real se refiere al hecho de que el punto A y el punto B se encuentran en comunicación directa a través de un medio, en este caso una red de datos con protocolo IP. Casos típicos son la teleconsulta, la teleasistencia y la teleeducación interactiva[40]. Esto permite una interacción entre los dos actores, que puede ser más eficaz que si se hiciera en tiempo diferido. Los datos a transmitirse en vivo son multimedia, lo que demanda gran ancho de banda de las redes de comunicación. Para aprovechar al máximo los recursos de las redes además de comprimir los datos es necesario implementar tecnologías y estándares de comunicación eficientes. En el presente capítulo se explican los protocolos utilizados en la transmisión de audio sobre redes IP.

5.2 Protocolo IP

El protocolo IP (protocolo de Internet) proporciona servicios de transmisión de datos en forma de datagramas (paquetes de información), desde un origen a un destino, posiblemente atravesando varias redes, está implementado en cada sistema final (servidores, PCs) e intermedio (routers, switch) y proporciona conectividad, direccionamiento y fragmentación. IP fue diseñado como un sistema independiente de los medios físicos y de mejor esfuerzo (best effort), en el que no se asegura ni la ruta, ni el tiempo de transmisión de un datagrama [22].

5.2.1 Protocolos de transporte

El propósito de la capa de transporte es segmentar los datos y brindar el control necesario para que los datos puedan ser reensamblados en su destino. Existen dos protocolos de trasporte: TCP y UDP. TCP (Protocolo de control de transporte) es un protocolo orientado a la conexión y que realiza control de flujo de datos y recuperación de paquetes perdidos, lo que agrega bits adicionales a cada segmento, este protocolo se utiliza para transportar datos en los que la integridad de la información es primordial como por ejemplo correo electrónico, páginas web, y transacciones bancarias, entre otros; por otro lado para aplicaciones en las que se puede soportar la perdida de paquetes como video y audio en tiempo real se utiliza el protocolo de transporte UDP (protocolo de datagrama de usuario) que es un

[40] Aplicaciones de las tecnologías de información: www.fiap.org.es/revista3_2.htm

protocolo simple que envía datos sin acuses de recibo o garantía de entrega. Este protocolo no es orientado a la conexión, requiere de aplicaciones para la retransmisión y la corrección de errores [22].

5.2.2 RTP Protocolo de transporte en tiempo real [17]

RTP es un protocolo basado en IP que proporciona soporte para el transporte de datos en tiempo real.

Este protocolo en un principio fue diseñado para emisiones multicast de tráfico en tiempo real, adicionalmente soporta emisiones unicast y se utiliza para video bajo demanda y servicios interactivos tales como videoconferencia en Internet.

RTP ha sido diseñado para funcionar junto con el protocolo de control auxiliar RTCP para mantener la calidad en la transmisión de datos y proporcionar información sobre los participantes al iniciarse la sesión. Este protocolo permite identificar el tipo de información transportada, añadir marcas temporales y números de secuencia de la información de transporte y controlar la llegada de los paquetes.

Los paquetes RTP y RTCP son transmitidos normalmente usando un servicio UDP/IP. Sin embargo, permiten un transporte independiente pudiendo utilizar CLNP (Connectionless Network Protocol), IPX (InternetWork Packet Exchange), AAL5/ATM u otros protocolos[41].

El paquete RTP se encapsula en un paquete UDP/IP, Figura 5.1

20 Bytes	8 Bytes	12 Bytes	Variable
Cabecera IP	**Cabecera UDP**	**Cabecera RTP**	**Datos de Audio y Video digital**

Figura 5.1 Empaquetamiento con RTP

Los paquetes RTP encapsulan la información de audio y video, este a su vez se encapsula en un segmento UDP y después pasa el segmento a IP. El lado receptor extrae el paquete RTP del segmento UDP y después pasa al reproductor de medios para su decodificación y procesamiento.

[41] http://cp.literature.agilent.com/litweb/pdf/5989-5245EN.pdf, Network communication protocols

5.2.3 Cabecera RTP [17].

El paquete RTP está formado por la información de audio, y la cabecera, que posee cuatro campos principales: el tipo de carga, número de secuencia, marca de tiempo, e identificador de fuente.

7 bits	16 bites	32 bits	bits	Variable
Tipo de carga	**Número de secuencia**	**Marca de tiempo**	**Identificador de la fuente de sincronización**	**Varios Campos**

Figura 5.2 Cabecera RTP

Tipo de carga: En este campo se identifica el tipo de codificación de audio que se emplea.

Número de secuencia: es de 16 bits, se incrementa en uno para cada paquete RTP que es enviado y puede ser utilizado por el receptor para detectar pérdida de secuencias en los paquetes.

Marca de tiempo: es de 32 bits y refleja el instante de muestreo del primer byte del paquete de datos RTP, el receptor utiliza las marcas de tiempo para eliminar la fluctuación de paquetes introducida en la red.

Identificador de la fuente de sincronización (SSRC): es de 32 bits, identifica la fuente del flujo RTP, cada flujo en una sesión RTP tiene un SSRC distinto. El SSRC no es la dirección IP del emisor sino que es un número que asigna la fuente aleatoriamente cuando comienza un nuevo flujo.

5.2.4 RCTP: Protocolo de Control de RTP [17]

Está diseñado para proveer realimentación sobre la calidad de servicio a los participantes de la sesión RTP, este protocolo es utilizado para enviar informes del receptor al transmisor en el proceso de transmisión de paquetes. En base a la información del retardo de paquetes en el dominio del tiempo o jitter, ancho de banda y perdida de paquetes, el remitente puede enviar los paquetes o adaptar la transmisión de acuerdo a los mensajes, con la sincronización del flujo de datos multimedia, admisión de sesión e información de descripción de la fuente. Provee los siguientes servicios:

Realimentación de QoS: es la función principal del RTCP, la información se envía a través de reportes de remitente y reportes de receptor.
Identificación del participante: la fuente puede ser identificada por el campo SSRC en la cabecera RTP.
Los paquetes RTCP llevan información de control, calidad de los datos transmitidos, control de flujo, congestión e informes estadísticos y anchos de banda adecuados y pueden ser:
SR (Sender Report): ofrece estadísticas de transmisión y recepción de los participantes que son emisores activos.
RR (Receiver Report): ofrece estadísticas de recepción de los participantes que no son emisores activos.
SDES (Source Description): lo utilizan los emisores para anunciarse, estos paquetes contienen información del usuario: teléfono, e-mail y otros.
BYE: indica el final de la participación.
Con la información de RTCP, los emisores pueden ajustar el flujo de datos según el estado de la red. [17]

5.3 Protocolo RTSP

Protocolo de streaming en tiempo real (RTSP, Real Time Streaming Protocol), trabaja a nivel de aplicación y control de la sesión para la realización de streaming de medios sobre Internet. Una de las funciones principales de RTSP es el soporte de funciones como: parada, pausa, resumir, avance rápido y retroceso rápido. RTSP funciona tanto en difusión punto a punto como en multidifusión, permite controlar múltiples sesiones y escoger protocolos de transporte a utilizar, como UDP o TCP.

El protocolo soporta las siguientes operaciones:
Petición de medios: el cliente pide una descripción de presentación vía HTTP u otro método. Si la presentación es multicast, la descripción contiene las direcciones multicast y los puertos que pueden ser usados.
Invitación a un servidor de medios para una conferencia: un servidor puede ser invitado a unirse a una conferencia existente, bien como participante o simplemente para grabar parte de la conferencia, útil para las aplicaciones de enseñanza distribuida.

Adición de medios a una presentación existente: particularmente en presentaciones en directo, el servidor avisa al cliente de que nuevos medios están disponibles. [21]

5.4 Ancho de banda de la red y tamaño del paquete.

El ancho de banda necesario para transportar audio sobre IP dependerá de las características de la pista de audio como el tipo de algoritmo, número de canales, tasa de muestreo etc., para conexiones síncronas se considera que el ancho de banda necesario es igual a la velocidad de transporte de bits del códec utilizado. Pero en IP se debe agregar un encabezado que es necesario para empaquetar los datos de audio [2].

Tabla 5.1 Relación entre ancho de banda, empaquetamiento, retardos y velocidad de datos [2].

Tasa de Datos de Audio	Tamaño del Paquete de Audio (bytes)	Tamaño del Paquete IP (bytes)	Paquete IP/sec	Retardo de Empaquetamiento (ms)	Tasa de Datos IP
64 kbps	128	194	62,5	16	97 kbps
	256	322	31,25	32	80,5 kbps
	512	578	15,625	64	72,3 kbps
	1280	1346	6,25	160	67,3 kbps
128 kbps	128	194	125	8	194 kbps
	256	322	62,5	16	161 kbps
	512	578	31,25	32	144,5 kbps
	1280	1346	12,5	80	134,6 kbps
256 kbps	128	194	250	4	388 kbps
	256	322	125	8	322 kbps
	512	578	62,5	16	289 kbps
	1280	1346	25	40	269,2 kbps
384 kbps	128	194	375	2,7	582 kbps
	256	322	187,5	5,3	483 kbps
	512	578	93,75	10,7	433,5 kbps
	1280	1346	37,5	26,7	403,8 kbps
576 kbps	128	194	562,5	1,8	873 kbps
	256	322	281,25	3,6	724,5 kbps
	512	578	140,625	7,1	650,3 kbps
	1280	1346	56,25	17,8	605,7 kbps

Las cabeceras que contienen los datos se deben incluir en cada paquete que se origina en una red IP, es decir que existe una relación entre el tamaño del paquete y el ancho de banda, en la tabla 5.1 se detalla la relación entre la calidad del audio el tamaño del paquete IP, el retardo y el ancho de banda necesario.

Elegir tamaños de paquetes IP largos permite reducir el ancho de banda y el jitter, pero también significa que si se pierde un paquete, un mayor segmento de audio se perderá perjudicando la calidad del servicio. Si se opta por paquetes IP más

pequeños se requiere un mayor ancho de banda debido al incremento de bits que aportan las cabeceras IP, pero se reduce el retardo por empaquetamiento. [2]

5.5 Jitter en Audio sobre IP

Es una característica de las redes con conmutación de paquetes, los paquetes pueden tomar cualquier ruta desde su origen al destino, el jitter ocurre cuando los paquetes llegan fuera del tiempo en el que eran esperados y el codec del receptor no es capaz de reconstruir la señal en tiempo real. [2]

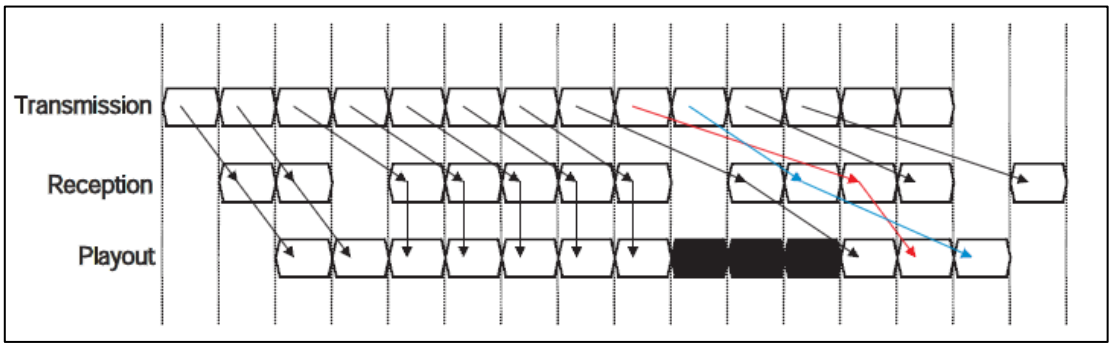


Figura 5.3 Efecto del Jitter en la reconstrucción de la señal [2]

Los efectos del jitter se pueden compensar introduciendo un buffer para almacenar suficientes paquetes para compensar los paquetes que no lleguen a tiempo. La Figura 5.3 muestra el efecto del jitter en la reconstrucción y la reproducción de un archivo de audio, en este caso se utiliza un buffer de dos paquetes si el jitter es bajo el sistema no se afectará, cabe recalcar que si el jitter rebasa la capacidad del buffer los paquetes llegarán después del tiempo de espera y se perderán lo que resulta en una mala calidad de audio. [2]

5.6 Retardo en Audio sobre IP

Las redes de telecomunicaciones por estar sujetas a leyes físicas sufren retardo en la transmisión de las señales eléctricas, este retardo no puede ser removido.

En una red IP existen dos retardos, el retardo estándar y el retardo producido por efecto del empaquetamiento, esta latencia será típicamente equivalente de 10-30ms, a esto se puede agregar el efecto de la longitud del paquete y el buffer de jitter.

La figura de latencia representa el retardo que sufre la señal cuando pasa por los switches, routers, etc., pero no incluye el retardo sufrido por el tiempo en que se

demora el codec en comprimir la señal, todo retardo producto del algoritmo de compresión escogido deberá ser añadido al retardo global del sistema. La elección del correcto algoritmo de compresión es crítica para determinar la latencia del sistema, en el caso de tiempo real se debe escoger codecs con técnicas de compresión de bajo retardo. [2]

5.7 Ancho de banda utilizado por Vorbis en streaming

Vorbis tiene su propio sistema de empaquetamiento que es Ogg (RFC 3533), pero para el efecto de transmitir un sonido codificado con Vorbis a través de una red se empaqueta las tramas salientes del codificador directamente sobre RTP (RFC 5215).

Para calcular el ancho de banda que utiliza Vorbis es necesario conocer que para la transmisión en tiempo real la trama RTP con su carga útil (tramas Vorbis) corre sobre UDP y este protocolo a su vez se empaqueta en IP Figura 5.4, es así, que para el cálculo del ancho de banda se necesita conocer el tamaño total de bits de cada paquete a transmitirse, la tasa de datos de audio y cada cuanto tiempo se transmiten dichos paquetes, ecuación 5.1. En la tabla 5.2 se muestra la longitud en bits de las tramas y protocolos utilizados en la transmisión.

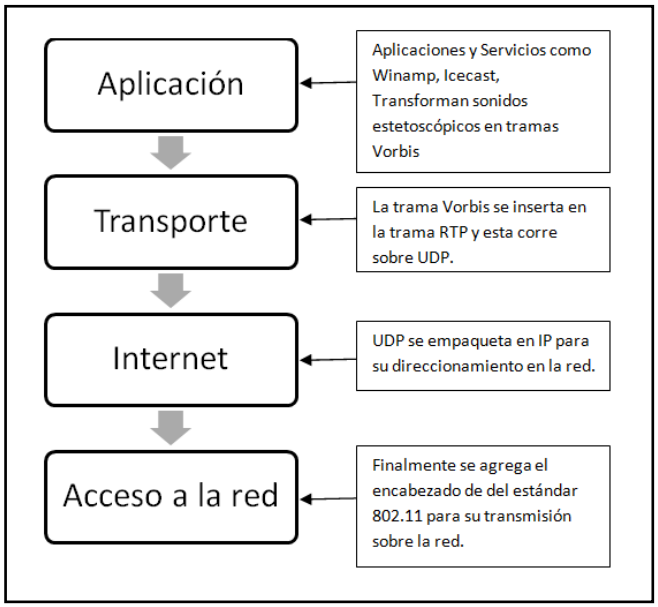


Figura 5.4 Vorbis en TCP/IP

Tabla 5.2 Longitud del paquete

Protocolo	Tamaño [bytes]
Vorbis	<800 (variable)
RTP	12 (variable)
UDP	8
IP	20 ~ 60
802.11	42

Según la RFC 5215 el número máximo de tramas Vorbis encapsuladas por cada paquete RTP es de 15, y el tiempo de retardo entre cada trama Vorbis es variable dependiendo del codificador y la capacidad del procesador que realiza esta operación, así, se podría considerar un valor promedio de 10ms con una tasa de audio aproximada de 30Kbps.

La ecuación 5.1 describe el proceso del cálculo propuesto[42].

$$AB = \frac{\big(Lp + (Lt * N)\big) * Tb}{Tt} \qquad (5.1)$$

$AB =$ Ancho de banda
$Lp =$ Longitud total de los encabezados de los protocolos involucrados
$Lt =$ Longitud de la trama Vorbis
$N =$ Número de tramas Vorbis por cada paquete RTP
$Tb =$ Tasa de bits de audio
$Tt =$ Retardo entre cada trama Vorbis

Por ejemplo, si consideramos la transmisión de audio en Vorbis sobre una red inalámbrica con 3 tramas Vorbis por paquete y que cada trama se transmite cada 10ms el ancho de banda total será la suma total de los encabezados de cada protocolo, esto da un total de 82 bytes, a esto se suma el tamaño de las tres

[42] Basado en el artículo "Cálculo de Ancho de Banda en VoIP" realizado por Julián María Ganzábal, www.lairent.com.ar

tramas de Vorbis, si consideramos un codificador que produzca tramas de 10 bytes tendremos 30 bytes de carga útil en RTP, esto se suma a los 82 bytes de los encabezados teniendo por resultado 112 bytes. Finalmente este valor se multiplica por la tasa de bits de audio y se divide para el retardo de las tramas Vorbis es decir 10ms x 3, así, el ancho de banda total es 112 x 8/30ms = 29.9 kbps.

Se debe señalar que este valor se incrementa por ejemplo si se introduce un protocolo RTCP en un 5% del ancho de banda total.

6 Conclusiones

1. El códec que demostró mejores características en cuanto al porcentaje de compresión como reconstrucción de la señal es el Ogg Vorbis y se comprobó que para comprimir señales estetoscópicas este formato es superior a los de MP3, AAC, y WMA.

2. Se eligió el formato Ogg Vorbis por la ventaja de que su código no tiene licencia comercial, razón por la cual se considera el mejor frente a los demás formatos de audio permitiendo que cualquier programador modifique el codificador acorde a sus necesidades.

3. Las pruebas realizadas demuestran que las señales estetoscópicas comprimidas con el códec Ogg Vorbis, no sufren alteraciones espectrales, ni temporales, y a la vez conservan la fase de la señal original.

4. El tipo de ventana que utiliza el códec Ogg Vorbis, para la fase análisis y síntesis de la MDCT es de longitud variable con lo se obtiene una mayor resolución en frecuencia, logrando así que las frecuencias bajas que comprenden los sonidos cardiacos no sean discriminadas al momento del análisis frecuencial.

5. El cálculo del ancho de banda en el códec Vorbis en una transmisión en tiempo real resulta complicado, debido a que la tasa de bits de audio es variable y la longitud del encabezado de la trama Vorbis no es constante, porque depende del diseño que se emplee al desarrollar el codificador; sin embargo se puede tener una aproximación utilizando valores estándar de la longitud del encabezado y del retardo en la transmisión de la trama Vorbis.

6. El ancho de banda aproximado utilizando el códec Ogg Vorbis sobre una red inalámbrica es 30 Kbps, que es un valor aceptable para transmitir sonidos estetoscópicos, dejando suficiente ancho de banda para otros dispositivos y aplicaciones que demandan mayores recursos como: la videoconferencia, la teleradiología, entre otros, utilizados en los sistemas de telemedicina.

7 Recomendaciones

1. Al momento de escoger un codec para comprimir señales cardiacas o respiratorias se debe tener cuidado en que este no modifique la señal, ya que pequeños desplazamientos en frecuencia o ruido extra pueden enmascarar posibles patologías y emitir un diagnóstico errado.

2. Para implementar el codec Ogg Vorbis en transmisión en tiempo real se recomienda utilizar el servidor Icecast debido a que soporta dicho formato y permite escoger entre varias tasas de bits.

3. En el caso de modificación del codificador se recomienda eliminar la característica multicanal ya que este formato soporta hasta 255 canales, los estetoscopios electrónicos contienen un solo canal, además se pueden descartar los campos de comentarios para ahorrar bytes en la trama ogg.

4. Se recomienda comprimir las señales estetoscópicas a un máximo de 64Kbps debido a que tasas de bits superiores no demuestran una diferencia importante en la calidad de la señal.

8 Bibliografía

[1]. Ambardar, Ashok. Procesamiento de señales analógicas y digitales. Thomson, segunda edición, 2002.

[2]. APT. The practical guide to audio over IP for broadcast. 2008.

[3]. Arribas, Juan Ignacio, Ph.D. Enmascaramiento Sonoro. Junio de 2009 http://www.lpi.tel.uva.es/~nacho/docencia/ing_ond_1/trabajos_06_07/io1/public_html/enascaramiento.htm.

[4]. Avila Calvillo, Sergio. ANÁLISIS COMPARATIVO DE LOS FORMATOS DE SONIDO DIGITAL: MP3, OGG VORBIS Y YAMAHA VQF. 2005.

[5]. Bosi, Mariana y Ricard E. Goldberg. Introduction to Digital Audio Coding and Standards. Publicaciones Kluwer Academic, 2003.

[6]. Cerquera, Edwin Alexander. Caracterización de estados funcionales en fonocardiografía empleando análisis acústico y técnicas de dinámica no lineal. 2005.

[7]. Data-Compression.com. Vector Quantization. <http://data-compression.com/vq.shtml>.

[8]. David, Salomon. A Concise Introdution to video and Audio Compression. Springer, 2008.

[9]. Jacob, S.W. Anatomía y Fisiología Humana. McGraw Hill, 1982.

[10]. Long, Marshall. Architectural Acoustics. Elsevier Academic Press, 2006.

[11]. Lufti, R. A. Additivity of Simultaneos Masking. Vol. 73. J. Acost. Soc Am., 1983. 162-267.

[12]. Martínez, Jose M y Javier Molina. Practica 1: Codificación Huffman. 2009. http://arantxa.ii.uam.es/~jms/tdatos/2009-2010/practicas/enunciados20092010/p1/Guion_1_Huffman_2009-2010.pdf.

[13]. Martínez-Alajarín J, Ruiz-Merino R. Wavelet and wavelet packet compression of phonocardiograms. Electronics Letters 40 (2004): 1040-1041.

[14]. Molina, Rafael. Introduccion a la Compresion de Datos. Universidad de Granada: Depto. de Ciencias de la Computación e Inteligencia Artificial, s.f.

[15]. Molina, Rafael y Javier Mateos. Codificación y Compresión de Datos: Codificación Aritmética. 2006. http://decsai.ugr.es/ccd/practicas/guion5.pdf.

[16]. Montejo Zarco, José y Pedro Nogales Aguado. Medynet.com. 2003. http://www.medynet.com/elmedico/aula2002/tema3/cardiacab1.htm.

[17]. Nico VanHaute, Julien Barascud, Jean-Roland Conca. kioskea.net. 2009. http://es.kioskea.net/contents/internet/rtcp.php3.

[18]. Painter, Ted y Spanias Adreas. Perceptual Coding of Digital Audio. Arizona: Telecommunications Research Center Arizona State University, s.f.

[19]. Pasterkamp, H. Respiratory Sounds. 1997.

[20]. Sánchez, Ignacio. «Aplicaciones clínicas del estudio objetivo.» 74.3 (2003).

[21]. Schulzrinne, H. Real Time Streaming Protocol (RTSP). 1998.

[22]. Tanenbaum, Andrew S. Redes de computadoras. Pearson, Cuarta Edición, 2003.

[23]. Terhardt, E. Calculating Virtual Pitch. Vol. I. Hearing Res., 1979.

[24]. Watkinson, J. An Introduction to Digital Audio. 1994.

[25]. Weinrauch, Larry A. clinicadam. 2007. <http://www.clinicadam.com/salud/5/003266.html>.

[26]. Weitzenfeld, Alfredo. «Cannes.» 2006. http://cannes.itam.mx. <http://cannes.itam.mx/Alfredo/Espaniol/Cursos/Grafica/Linea.pdf.

[27]. Xiph.org Foundation. Xiph open source community. 2 de Junio de 2009. www.xiph.org. Agosto de 2009 http://xiph.org/vorbis/doc/.

[28]. Zwicker, E y H Fastl. Psychoacoustics: Facts and Models. Berlin Heidelberg: Springer-Verlag, 1990.

[29]. Zwicker, H.F. Eberhard. Psychoacoustics: Facts and Models. Segunda Edición. Springer: Information Sciences, 1999.

Glosario de Términos.

Termino o Abreviaturas	Definición
A/D	Analógico/Digital
ADC	Analog to Digital converter
Bluetooth	Tecnología de transmisión de datos por radio ITU 802.
Carótida	Cada una de las dos arterias, que por uno y otro lado del cuello llevan la sangre a la cabeza.
Códec	Codificador/decodificador
Distensibilidad	Tensión violenta de membranas
Doppler	El ultrasonido o Doppler se emplea para medir el flujo de un líquido corporal, por ejemplo, el flujo sanguíneo.
ECG	Electrocardiograma
ELF	Frecuencias extremadamente bajas
Estenosis	Estrechamiento de los conductos de las arterias
Eyectivo	Impulsar con fuerza
FCG	Fonocardiograma
FPGA	Field Program Gate Array
Miocardio	Parte musculosa del corazón de los vertebrados, situada entre el pericardio y el endocardio.
Mitral	válvula
Pericardio	Envoltura del corazón, que está formada por dos membranas, una externa y fibrosa, y otra interna y serosa.
Prolapso	Caída o descenso de una víscera, o del todo o parte de un órgano.
Protodiástole	Primera fase de la diástole cardiaca
Proxy	Punto intermedio entre un ordenador conectado a Internet y el servidor que está accediendo.
RDSI	Red Digital de Servicios Integrados
Sistólico	Movimiento de contracción del corazón y de las arterias para empujar la sangre que contienen
Streaming	Tecnología que permite reproducir contenidos en tiempo real en internet.
Tricúspide	válvula
Ultrasonido (medicina)	Técnica diagnóstica en la que un sonido de frecuencia muy alta es dirigido hacia el organismo; se conoce como ecografía.
Ventricular	Cavidad del corazón que recibe sangre de una aurícula y la impulsa por el sistema arterial

ANEXO 1

Función en Matlab para calcular la MDCT (Ver ecuación 3.5)

```
function y = fast_mdct(x)
x=x(:);
N=length(x);
n0 = (N/2+1)/2;
wa = sin(([0:N-1]'+0.5)/N*pi);
y = zeros(N/2,1);
x = x .* exp(-j*2*pi*[0:N-1]'/2/N) .* wa;
X = fft(x);
y = real(X(1:N/2) .* exp(-j*2*pi*n0*([0:N/2-1]'+0.5)/N));
y=y(:);
```

Función en Matlab para calcular la IMDCT (Ver ecuación 3.6)

```
function y = fast_imdct(X)
N = 2*length(X);
ws = sin(([0:N-1]'+0.5)/N*pi);
n0 = (N/2+1)/2;
Y = zeros(N,1);
Y(1:N/2) = X;
Y(N/2+1:N) = -1*flipud(X);
Y = Y .* exp(j*2*pi*[0:N-1]'*n0/N);
y = ifft(Y);
y = 2*ws .* real(y .* exp(j*2*pi*([0:N-1]'+n0)/2/N));
```

Código para graficar una muestra de sonidos cardiacos dividido en 4 bloques.

```
sonido = wavread('Mediawebserver2.wav');


%muestras ya solapadas
t = sonido(1:768);
a = [zeros(256,1);t(1:256)];
b = t(1:512);
c = t(256:768);
d = [t(512:768);zeros(256,1)];

%transformada mdct
j = fast_mdct(a);
k = fast_mdct(b);
l = fast_mdct(c);
m = fast_mdct(d);

%Transformada inversa
o = fast_imdct(j);
p = fast_imdct(k);
q = fast_imdct(l);
r = fast_imdct(m);
```

```
s = o(257:512);
t2= p(1:256);
u = p (257:512);
v = q (1:256);
w = q (257:512);
y = r (1:256);

% Suma Solapada de los Bloques

sum = [(s + t2); (u+v);(w +y)];


% Figura 3.4 Bloques Solapados 50%

figure (1);
subplot (2,2,1);
plot(a), grid on
axis([0  512  -0.6  0.6])
title('Bloque 1')

subplot(2,2,2);
plot (b), grid on
axis([0  512  -0.6  0.6])
title('Bloque 2')

subplot(2,2,3);
plot(c),  grid on
axis([0  512  -0.6  0.6])
title('Bloque 3')

subplot(2,2,4);
plot(d),  grid on
axis([0  512  -0.6  0.6])
title('Bloque 4')



figure (2);
subplot(2,1,1);
plot(t), grid on
title('Figura 3.3 Sonido Cardiaco de 768 muestras')
xlabel('Tiempo(muestras)')
axis([0  768  -0.6  0.6])

subplot(2,1,2);
plot (sum),grid on
title('Señal reconstruida')
axis([0  768  -0.6  0.6])

%Figura 3.5 Espectro bloques con MDCT


figure (3);
subplot (2,2,1);
plot(j), grid on
title('Bloque 1 MDCT')
```

```
subplot(2,2,2);
plot (k), grid on
title('Bloque 2 MDCT')

subplot(2,2,3);
plot(l), grid on
title('Bloque 3 MDCT')


subplot(2,2,4);
plot(m), grid on
title('Bloque 4 MDCT')


% Figura 3.6 Bloques aplicando la IMDCT

figure (4);
subplot (2,2,1);
plot(o), grid on
title('Bloque 1 IMDCT')
axis([0  512  -0.6  0.6])

subplot(2,2,2);
plot (p), grid on
title('Bloque 2 IMDCT')
axis([0  512  -0.6  0.6])

subplot(2,2,3);
plot(q), grid on
title('Bloque 3 IMDCT')
axis([0  512  -0.6  0.6])


subplot(2,2,4);
plot(r), grid on
title('Bloque 4 IMDCT')
axis([0  512  -0.6  0.6])
```

Código para comparar temporalmente de señales codificadas en 4 diferentes formatos de audio. (Ver sección 3.5)

```
%Figura 3.14 Señales comprimidas comparadas con la señal original

%lectura de señales de audio
x = wavread('WAV_OR.wav');
x1 = wavread('MP4_64');
x2 = wavread('OGG_64.wav');
x3 = wavread('WMA_64.wav');
x4 = wavread('MP3_64.wav');

n = 10000;%número de muestras tomadas de cada señal

y = x(1:n);
y1 = x1(1:n);
```

```
y2 = x2(1:n);
y3 = x3(1:n);
y4 = x4(1:n);

%Gráficas temporales
plot(y,'r-')
hold on
plot(y1,'m-')
hold on
plot(y2,'b--')
hold on
plot(y3,'k:')
hold on
plot(y4,'g-')

h = legend('señal original','MP4','OGG','WMA', 'MP3',5);
set(h,'Interpreter','none')
```

Código para obtener los espectros de las señales codificadas. (Ver sección 3.5.3)

```
%Figura 3.15 Comparación en el dominio de la frecuencia


%Lectura de las señales y obtención de la fft
x = wavread('MP3_64.wav');
fre = abs(fft(x));
f = (0:length(fre)-1)'*500/length(fre);
subplot(321);
plot(f,fre);
grid on
zoom on
ylabel('MP3');

x1 = wavread('MP4_64.wav');
fre1 = abs(fft(x1));
f1 = (0:length(fre1)-1)'*500/length(fre1);
subplot(322);
plot(f1,fre1);
grid on
zoom on
ylabel('AAC');

x2 = wavread('OGG_64.wav');
fre2 = abs(fft(x2));
f2 = (0:length(fre2)-1)'*500/length(fre2);
subplot(323);
plot(f2,fre2);
grid on
zoom on
ylabel('OGG');

x3 = wavread('WMA_64.wav');
fre3 = abs(fft(x3));
```

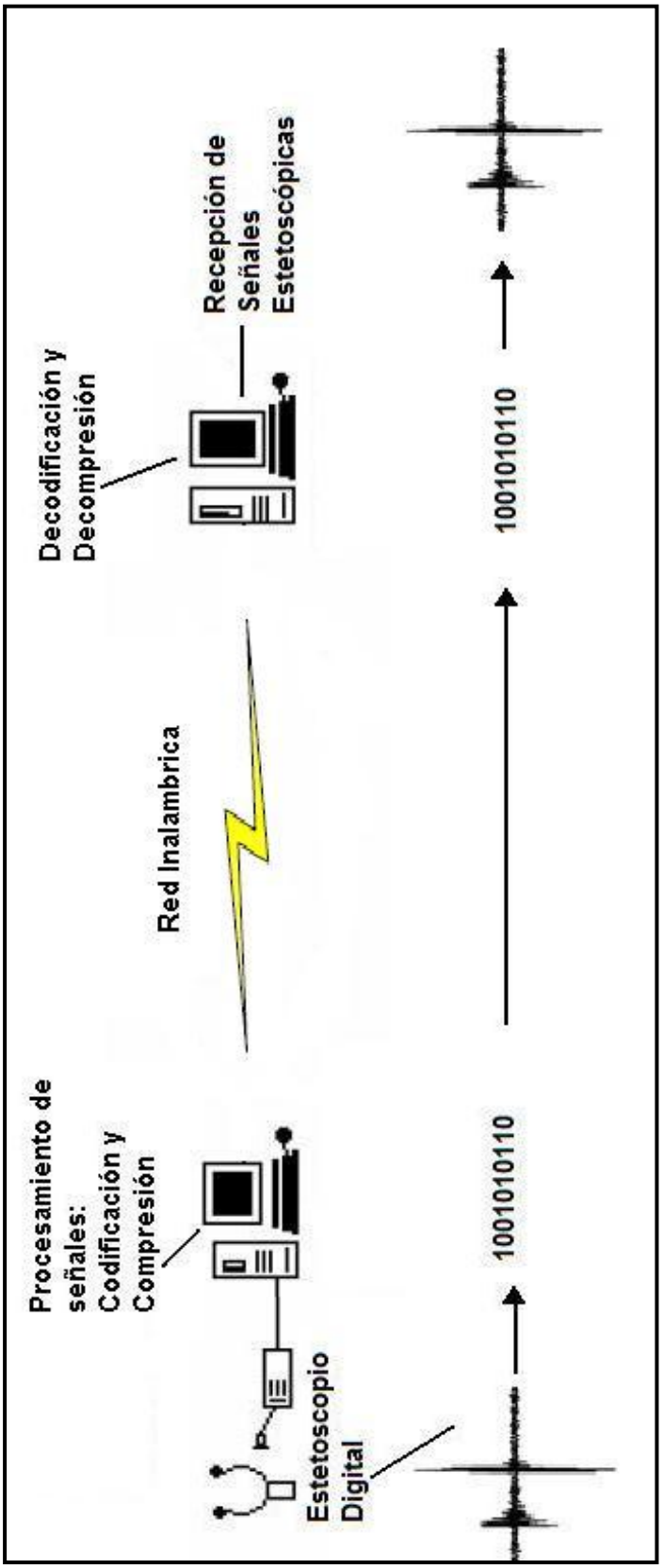
```
f3 = (0:length(fre3)-1)'*500/length(fre3);
subplot(324);
plot(f3,fre3);
grid on
zoom on
ylabel('WMA');

x4 = wavread('WAV_OR.wav');
fre4 = abs(fft(x4));
f4 = (0:length(fre4)-1)'*500/length(fre4);
subplot(325);
plot(f4,fre4);
grid on
zoom on
ylabel('señal original');
```

Código para obtener la respuesta en fase de los diferentes formatos de audio codificando un sonido cardiaco. (Ver sección 3.5.4)

```
%Figura 3.16 Desvío de fase de los codecs comparados

df = [ 0:40000 - 1 ] - (40000/2) ;
orig = wavread('WAV_OR.wav');
ogg = wavread('OGG_64.wav');
AAC = wavread('MP4_64.wav');
WMA = wavread('WMA_64.wav');
MP3 = wavread('MP3_64.wav');

FFTlength = 40000;

%Función para obtener la fase de cada señal
origphase = unwrap(angle(fftshift(fft(orig(:,1),FFTlength))));
oggphase = unwrap(angle(fftshift(fft(ogg(:,1),FFTlength))));
AACphase = unwrap(angle(fftshift(fft(AAC(:,1),FFTlength))));
WMAphase = unwrap(angle(fftshift(fft(WMA(:,1),FFTlength))));
MP3phase = unwrap(angle(fftshift(fft(MP3(:,1),FFTlength))));

plot(df, origphase(:,1), 'g' ) ; ylabel('Original')
hold on
plot(df, oggphase(:,1) , 'b' ) ; ylabel('OGG')
hold on
plot(df, AACphase(:,1), 'm' ) ; ylabel('AAC')
hold on
plot(df, WMAphase(:,1), 'k' ) ; ylabel('WMA')
hold on
plot(df, MP3phase(:,1), 'r' ) ; ylabel('MP3')
hold on
ylabel('Desvío de fase')
xlabel('Frecuencia Hz')

h = legend('original','OGG','AAC','WMA','MP3',5);
set(h,'Interpreter','none')
```

ANEXO 2

Esquema General del Sistema



Estudio de un Códec de Compresión para mejorar la Calidad de Servicio de Sonidos Estetoscópicos sobre una Red IP

Rohoden Katty [1], Castillo Tuesman [2], y Romero Jaime [3]
[1]*karohoden@gmail.com*, [2]*tuesman@gmail.com*, [3]*jasrom12@gmail.com*

Resumen.- *El presente trabajo enfoca la selección de un códec de compresión que permita disminuir el ancho de banda que demanda una red IP para la transmisión de sonidos estetoscópicos, partiendo del estudio y análisis de las principales características y naturaleza de los sonidos fisiológicos, con el objeto de delinear el tipo de compresión y códec a utilizar, para lo cual se propone un códec modelo. Luego se seleccionará un códec existente en el mercado que se adapte a los requerimientos planteados; finalmente se presenta un acercamiento a la transmisión en tiempo real con redes IP, que permita la auscultación remota de los sonidos cardiacos y respiratorios, como una técnica innovadora en el campo de la telemedicina.*

Abstract.- *The present work focuses the selection of a compression códec that allows to decrease the band width that demands an IP network for the transmission of sounds stethoscopic sounds, begining with the study and characteristic analysis of the main ones and nature of physiologic sounds, in order to delineating the compression type and códec to use. Then an existent códec will be selected in the market that adapts to the outlined requirements; finally an approach is presented to streaming with IP networks that allows the remote auscultation of the heart and breathing sounds, like an innovative technique in the field of the telemedicine.*

Palabras Clave.- *Códec, compresión, estetoscopio.*

I. INTRODUCCIÓN

En las últimas décadas el desarrollo tecnológico, tanto en la electrónica como en el procesamiento de señales, ha permitido el aparecimiento de dispositivos de diagnóstico médico más precisos y complejos en comparación con el tradicional estetoscopio. Este dispositivo se ha visto parcialmente reemplazado en los modernos centros de salud por técnicas como el electrocardiograma y el ultrasonido, sin embargo la simplicidad y la eficacia que provee el estetoscopio al momento de realizar una valoración médica inmediata, ha hecho que evolucionen en forma de estetoscopios electrónicos o digitales para integrarse a nuevos escenarios y tecnologías como la construcción de fonocardiogramas en tiempo real, la grabación y posterior reproducción de sonidos en el mismo dispositivo para su análisis y, cómo no, la telemedicina. Los sistemas de telemedicina aprovechan las bondades de las telecomunicaciones para implementar nuevos métodos de consulta a distancia, valiéndose de dispositivos electrónicos de diagnóstico y de captura de información médica que deben operar de forma remota. Una de las características importantes de la telemedicina es la de transmitir y controlar el abundante flujo de información que se produce en una teleconsulta, el ancho de banda de un sistema de comunicaciones siempre será limitado ya sea por el elevado costo de implementar una nueva red, porque se tiene que operar sobre redes alquiladas o por un proveedor de servicios de internet. Es así, que es necesario contar con técnicas que permiten aprovechar de mejor manera el ancho de banda disponible en una red IP, como la compresión de datos, sonido e imágenes y protocolos de comunicación eficientes.

II. SONIDOS CARDIACOS Y RESPIRATORIOS

Estos sonidos son producidos en la caja torácica y son utilizados en medicina para examinar el funcionamiento de estructuras que conforman el corazón y el aparato respiratorio con el objeto de detectar si estos se producen en el momento, intensidad y duración normal. En la Tabla 1 se muestra la duración y frecuencia de los sonidos cardiacos tanto normales, sonidos uno y dos (S1 y S2) y patológicos o anormales, sonidos 3 y 4 (S3 y S4). [1].

TABLA 1
RANGO DE FRECUENCIAS DE SONIDOS CARDIACOS

Ruido	Duración [s]	Rango frecuencial [Hz]
S1	0.1-0.12	20-150
S2	0.08-0.14	50-60
S3	0.04-0.05	20-50
S4	0.04-0.05	<25

En cuanto a los sonidos respiratorios su caracterización según [3] se muestra en la Tabla 2, se pueden diferenciar los sonidos normales como son el Pulmonar normal y Traqueal normal y los sonidos adventicios o anoramales entre los que se encuentran los Roncus, Sibilancias y Estridor.

TABLA 2 CARACTERÍSTICAS FRECUENCIALES DE LOS SONIDOS RESPIRATORIOS

Sonido	Rango frecuencial [Hz]
Pulmonar normal	100 - 800
Traqueal normal	200 - 1500
Roncus	< 300
Sibilancias	100 - 1000
Estridor	200 - 1500

III. ESQUEMA DE UN CÓDEC DE AUDIO

El codec tiene como objetivo reducir la cantidad de bits necesarios para transmitir los sonidos cardiacos y respiratorios por una red sin sacrificar la calidad, para lo cual se elegirá el esquema de un codificar sin pérdidas debido a la naturaleza del sistema que se desea transmitir audio en tiempo real sobre IP. Además, se tomara en cuenta las siguientes consideraciones:

- Los algoritmos con perdidas no deben discriminar las bandas espectrales de las Tablas 1 y 2.
- Se aplicará una compresión de tipo perceptivo debido a que las frecuencias necesarias para el diagnostico se encuentran dentro del umbral auditivo humano.
- Se elegirá un esquema de codificación de un solo canal debido a que los sonidos del estetoscopio electrónico son monoaurales.
- El códec tendrá la capacidad para ser transmitido en un flujo de datos en tiempo real, conocido también como streaming,
- Debe ser soportado por los protocolos RTP (Real Time Protocol) y UDP (User Datagram Protocol).
- El ancho de banda que demanda el códec para su transmisión no debe ser superior a 64Kbps.

A. Codificador

A continuación se detalla la estructura del codificador Fig. 1 utilizando la transformada modificada discreta del coseno MDCT para el análisis frecuencial, el modelo psicoacústico (sin perdidas) para el análisis perceptual, y por último para reducir el número de bits redundantes se utilizará codificación Huffman.

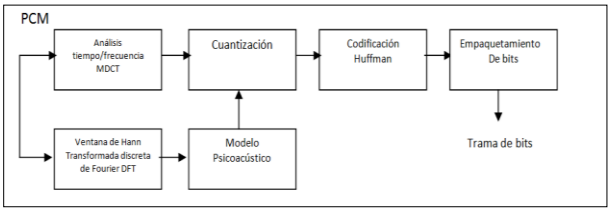


FIG. 1 ESQUEMA DE DECODIFICADOR[4]

B. Decodificador

El proceso de decodificación es mucho más simple que el de codificación debido a que el análisis de tipo psicoacústico se realiza en el codificador, además sólo se efectúa la lectura de los datos codificados y se aplica los procesos matemáticos y probabilísticos inversos. Fig. 2.

C. Selección del codec

El códec propuesto es de naturaleza perceptiva, en el mercado existen varios códecs de este tipo y con prestaciones similares debido a que sus algoritmos de codificación manejan herramientas como la MDCT, el enmascaramiento psicoacústico y codificación sin perdidas Huffman.

Los codecs que han sido considerados y que cumplen con los requerimientos son:

- *MP3* (MPEG Audio Layer 3)
- *WMA* (Windows media audio)
- *AAC* (Advanced audio coding)
- *Ogg Vorbis*

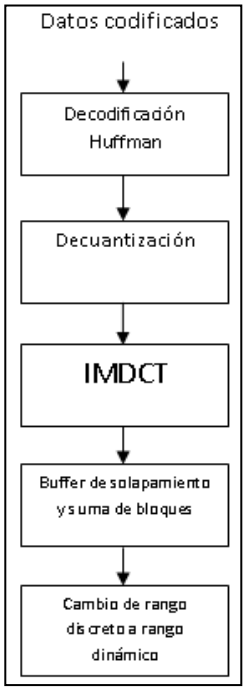


FIG. 2 ESQUEMA DEL DECODIFICADOR[4]

La comparación entre formatos de audio es complicada debido a que cada codec tiene características propias e indicadores de calidad diferentes. En el presente trabajo la comparación se realizará basándose en el comportamiento de cada codec con relación al archivo original no comprimido en formato WAV. El archivo de prueba es una pista de 4 segundos que corresponde a un sonido cardiaco captado por un estetoscopio digital de la marca 3M[1].

1) Tamaño de los archivos

La señal original en .wav ha sido comprimida en los formatos WMA, AAC, MP3 y Ogg, el software utilizado para la codificación y la decodificación es el plugin de Winamp versión 5.552 [ml_transcode.dll].

Los archivos se han establecido en CBR (tasa constante de bits) de 64Kbps para WMA, MP3 y AAC y con un solo canal (mono), Ogg se ha establecido en calidad 10 debido a que produce archivos cercanos a 64Kbps. En la Tabla 3 se presenta la tasa de compresión de un sonido cardiaco utilizando los diferentes codecs que satisfacen las necesidades del sistema:

[1] Enlace del archivo: http://solutions.3mchile.cl/wps/portal/3M/es_CL/Littmann-WW/stethoscope/

TABLA 3
COMPARACIÓN PORCENTAJE DE COMPRESIÓN A 64KBPS

Formato	Velocidad [Kbps]	Duración [s]	Tamaño [KB]	Porcentaje Compresión
.WAV (archivo original)	176	4	97	--------
.WMA	64	4	50	48.45%
.AAC	64	4	39	59.79%
.MP3	64	4	36	62.89%
.Ogg	64	4	30	69.07%

2) Comparación temporal.

Se realiza una comparación en el dominio del tiempo entre dos señales, la señal original Fig. 3 en .wav y la señal comprimida, además esta última es transformada de nuevo a .wav para observar posibles variaciones sufridas en el proceso de compresión.

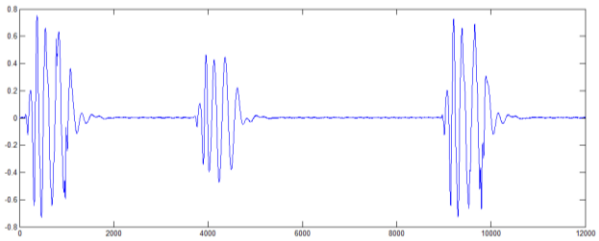


FIG. 3 SEÑAL ORIGINAL

En la Fig. 4 se observa que de los códec utilizados para comprimir la señal original, las señales Ogg y WMA no presentan alteraciones en el tiempo y son indistinguibles en la grafica.

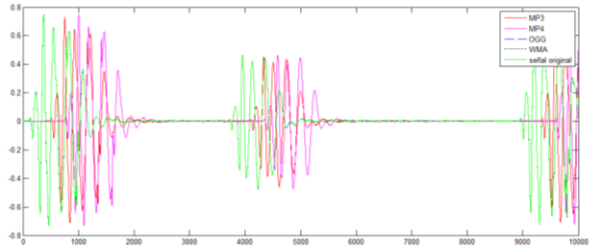


FIG. 4 COMPARACIÓN SEÑALES COMPRIMIDAS Y SEÑAL ORIGINAL

3) Comparación en frecuencia

En la Fig. 5 se observa el espectro que producen las señales comprimidas y la señal original.

Como se observa en la Fig. 5 las mejores aproximaciones en cuanto a espectro se logran con Ogg Vorbis, AAC o MP4 y WMA, en el caso del MP3 se aprecia un incremento en la energía de algunos componentes frecuenciales (circulo magenta), Ogg permite obtener reconstrucciones espectrales más precisas debido al enventanamiento de varias longitudes para la transformada MDCT.

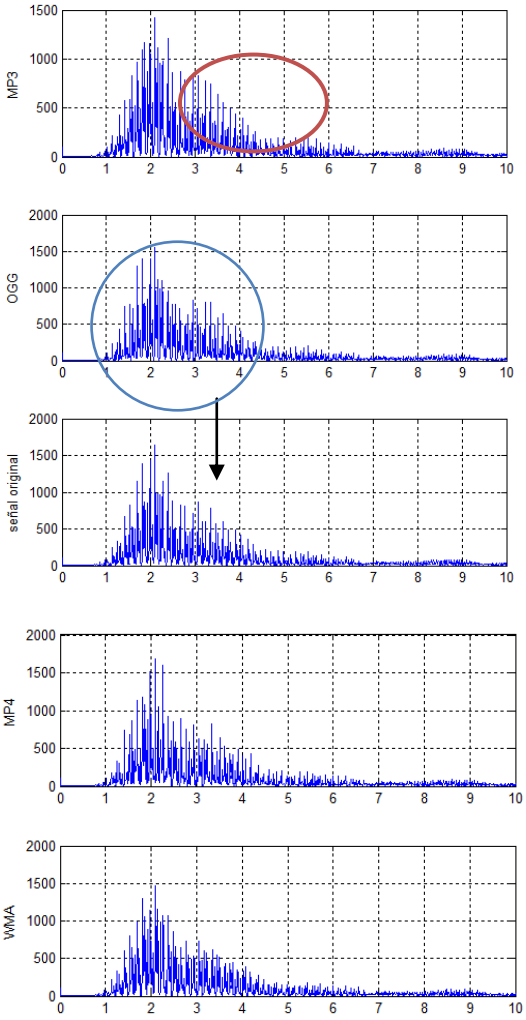


FIG. 5 COMPARACIÓN FRECUENCIAL DE LAS SEÑALES COMPRIMIDAS CON LA SEÑAL ORIGINAL

4) Comparación en fase

Todos los codecs agregan una desviación de fase con respecto a la señal original pero según se observa los códecs MP3 y AAC tienen un desvío mucho más amplio que los formatos Ogg y WMA.

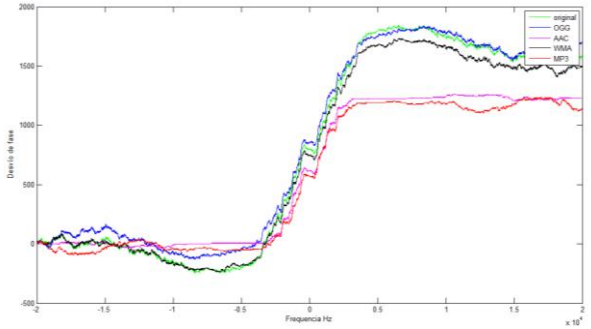


FIG. 6 DESVÍO DE FASE DE LOS CODECS DE COMPRESIÓN

Según las características de los diferentes formatos de codificación se considera que AAC y Ogg son los que proveen mejores prestaciones debido a que el sistema de transmisión en tiempo real demanda que se utilicen bajas tasas de bits menores a 48 kbps.

Finalmente, se considera que Ogg es el más conveniente para el sistema de transmisión debido a que su licencia es de libre uso y sobre todo a que el código es abierto y se puede modificar acorde a las necesidades que surjan en el proceso, sin las ataduras de los codecs comerciales.

IV. CÓDEC OGG VORBIS

Vorbis es un códec de uso general de tipo perceptivo y de código abierto, libre de patente y está basado en la licencia pública GNU, su calidad es comparable con codecs de audio de última generación como MP3Pro, WMA V8 y AAC, además soporta tasas de bits tan bajas como 16Kbps, aunque teóricamente puede llegar hasta 8Kbps según [2]. Su ventaja sobre otros formatos como el MP3 radica en que Vorbis además de ofrecer una mejor calidad a bajas tasas de bit, es completamente gratuito, su código está abierto a posibles modificaciones y mejoras por parte de cualquier programador.

A. *Codificador*

Ogg Vorbis sólo define su decodificador, esto quiere decir que cualquier codificador que produzca una trama decodificable por Vorbis es considerado un codificador Vorbis, en la Fig. 7 se muestra un esquema de un codificador Vorbis base.

1) *Generación de ventanas:* La trama de audio entrante PCM es dividida en bloques llamados ventanas, esto se hace para efectos de reducción del pre-eco producido por la posterior transformación de dominio con MDCT.
2) *Transformación de dominio MDCT:* Debido a que el estándar del codificador en Vorbis I es abierto, es posible utilizar cualquier tipo de transformada, en el caso de la primera versión de Vorbis se utiliza la MDCT.

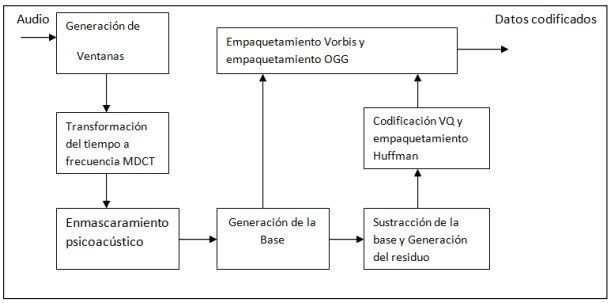


FIG. 7 CODIFICACIÓN OGG VORBIS

3) *Enmascaramiento Psicoacústico:* Vorbis I utiliza el modelo psicoacústico humano para descartar información no audible, característica que lo ubica en el grupo de codec con pérdidas, pero a diferencia de los otros codecs que utilizan un modelo con un volumen fijo durante la duración de la trama de audio, Vorbis asume que el volumen se ajusta dinámicamente con un máximo situado en el umbral del dolor.

4) *Generación de la base:* La base, llamada floor por Xiph.org, es una versión del espectro de la señal de baja resolución.
5) *Generación del residuo:* El residuo es el producto de restar la base a la señal espectral del audio, se codifica utilizando el vector de cuantización y se empaqueta con Huffman.
6) *Empaquetamiento Vorbis:* Finalmente se agrega la información codificada a la trama Vorbis, se incluye el número de versión de Vorbis, el número de canales de audio y la tasa de bits, luego en la cabecera de comentarios se ubica información como autoría del archivo, organización, fecha, lugar, entre otros, en la cabecera de configuración yacen los códigos, la señal base, los residuos y el modo que indica el tipo de ventanas y transformada, por último se agrega los paquetes de audio.

B. *Decodificador*

Las tramas se decodifican a través del siguiente proceso, la Fig. 8 muestra la síntesis del proceso.

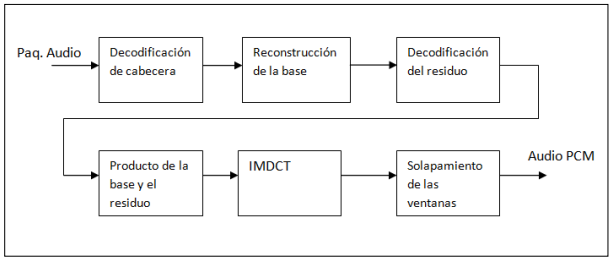


FIG. 8 PROCESO DE DECODIFICACIÓN

1) Decodificación de cabecera: En este bloque se decodifica la información necesaria para el proceso de reconstrucción del audio, se realiza los siguientes pasos:
2) Decodificación de la bandera que señala el tipo de paquete.
3) Decodificación del número de modo.
4) Decodificación del tipo de ventana.
5) Reconstrucción de la base: Se decodifica los vectores correspondientes a la base y se la reconstruye a través del algoritmo de Bresenham.
6) Decodificación de los residuos en vectores de residuos.
7) Cálculo del producto de la base y de los residuos generando un vector del espectro del audio.
8) Transformada inversa monolítica del vector del espectro de audio, siempre de tipo MDCT en Vorbis I.
9) Solapamiento de las ventanas: Finalmente se realizan los siguientes pasos:
10) Solapar/adicionar la salida de parte izquierda de los bloques de la transformada con la salida de la parte derecha del anterior bloque.
11) Guardar los datos de la parte derecha del bloque actual para el siguiente solapamiento.
12) Si no es el primer bloque, devuelve los resultados del proceso de solapar/adicionar como audio resultante del bloque actual.

V. ANCHO DE BANDA UTILIZADO POR VORBIS EN STREAMING[2]

Vorbis tiene su propio sistema de empaquetamiento que es Ogg [RFC 3533], pero para el efecto de transmitir un sonido codificado con Vorbis a través de una red se empaqueta las tramas salientes del codificador directamente sobre RTP [RFC 5215].

Para calcular el ancho de banda que utiliza vorbis es necesario conocer que para la transmisión en tiempo real la trama RTP con su carga útil (tramas Vorbis) corre sobre UDP y este protocolo a su vez se empaqueta en IP Fig 3, es así, que para el cálculo del ancho de banda se necesita conocer el tamaño total de bits de cada paquete a transmitirse, la tasa de datos de audio y cada cuanto tiempo se transmiten dichos paquetes, ecuación #. En la Tabla 3 se muestra la longitud en bits de las tramas y protocolos utilizados en la transmisión.

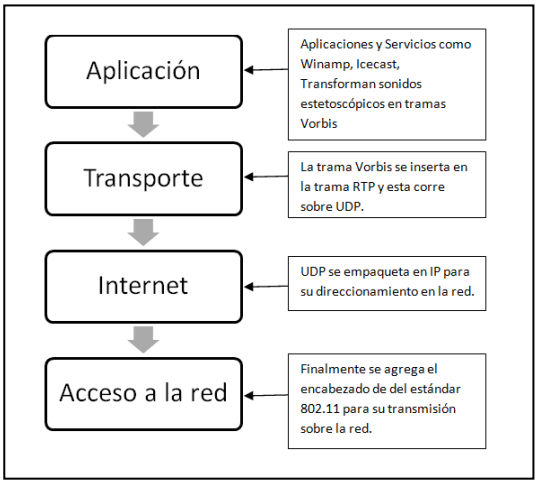


FIG. 9 VORBIS EN TCP/IP

TABLA 4
LONGITUD DEL PAQUETE

Protocolo	Tamaño [bytes]
Vorbis	<800 (variable)
RTP	12 (variable)
UDP	8
IP	20 ~ 60
802.11	42

Según [RFC 5215] el número máximo de tramas Vorbis encapsuladas por cada paquete RTP es de 15, y el tiempo de retardo entre cada trama vorbis es variable dependiendo del codificador y la capacidad del procesador que realiza esta operación, así, se podría considerar un valor promedio de 10ms con una taza de audio aproximada de 30Kbps.

La ecuación 1 describe el proceso del cálculo propuesto.

$$AB = \frac{(Lp+(Lt*N))*Tb}{Tt} \qquad (1)$$

AB = Ancho de banda

Lp = Longitud total de los encabezados de los protocolos involucrados
Lt = Longitud de la trama Vorbis
N = Número de tramas Vorbis por cada paquete RTP
Tb = Tasa de bits de audio
Tt = Retardo entre cada trama vorbis

Se debe señalar que este valor se incrementa si se introduce un protocolo RTCP en un 5% del ancho de banda total.

VI. CONCLUSIONES

- El codec que demostró mejores características en cuanto al porcentaje de compresión como reconstrucción de la señal es el Ogg Vorbis y se comprobó que para comprimir señales estetoscópicas este formato es superior a los de MP3, AAC, y WMA.
- Se eligió el formato Ogg Vorbis por la ventaja de que su código no tiene licencia comercial, razón por la cual se considera el mejor frente a los demás formatos de audio permitiendo que cualquier programador modifique el codificador acorde a sus necesidades.
- Las pruebas realizadas demuestran que las señales estetoscópicas comprimidas con el códec Ogg Vorbis, no sufren alteraciones espectrales, ni temporales, y a la vez conservan la fase de la señal original.
- El ancho de banda aproximado utilizando el códec Ogg Vorbis sobre una red inalámbrica es 30 Kbps, que es un valor aceptable para transmitir sonidos estetoscópicos, dejando suficiente ancho de banda para otros dispositivos y aplicaciones que demandan mayores recursos como: la videoconferencia, la teleradiología, entre otros, utilizados en los sistemas de telemedicina.

REFERENCIAS

[1]. Jacob, S.W. "Anatomía y Fisiología Humana". McGraw Hill, 1982.

[2]. Nico VanHaute, Julien Barascud, Jean-Roland Conca. kioskea.net. 2009. http://es.kioskea.net/contents/internet/rtcp.php3

[3]. Pasterkamp, H. "Respiratory Sounds". 1997.

[4]. Recomendación UIT-R BS.1115

[5]. Xiph.org Foundation. «Xiph open source community.» 2 de Junio de 2009. www.xiph.org. Agosto de 2009 <http://xiph.org/vorbis/doc/>.

[2] Transmisión de datos en tiempo real