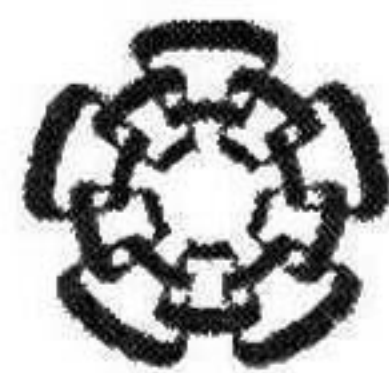


XX (178614.1)



CINVESTAV
BIBLIOTECA CENTRAL

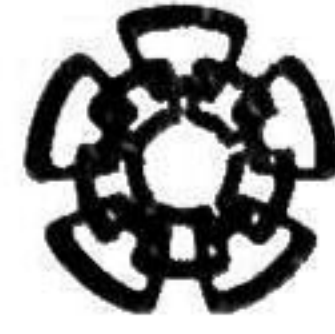
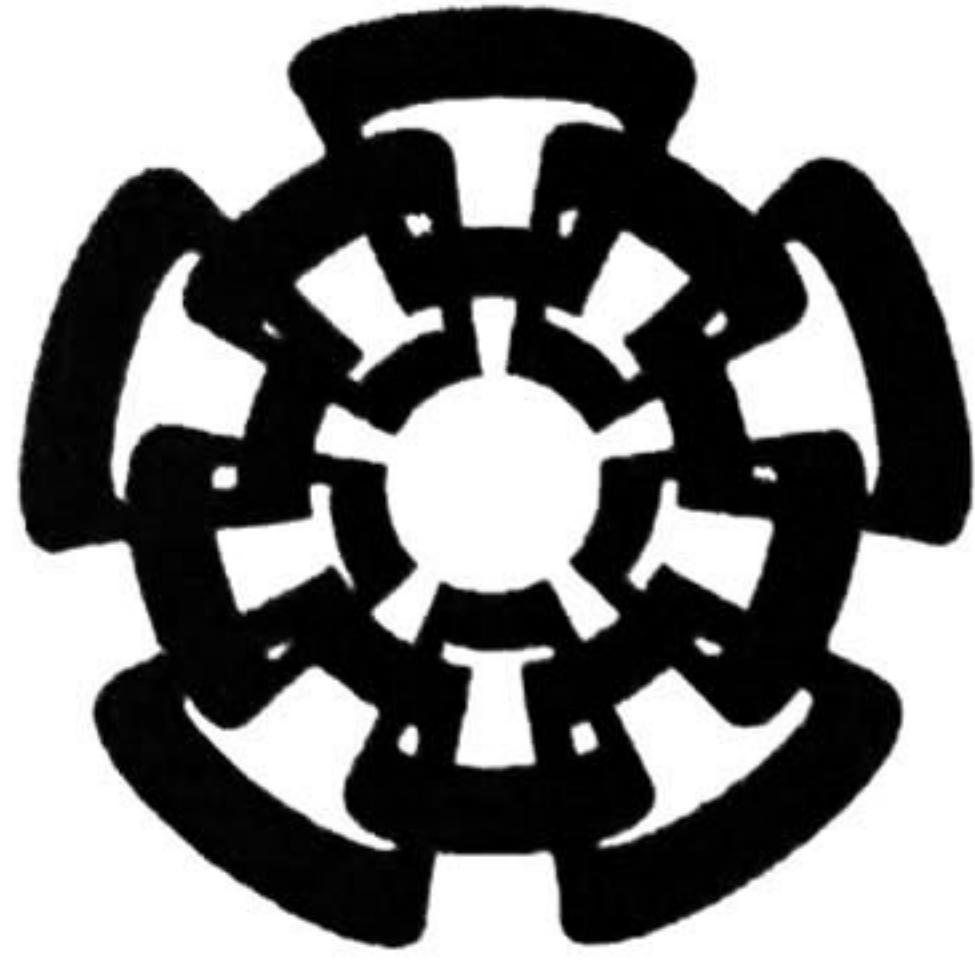


SSIT000004104

TK 165 G8

A75

2009



CENTRO DE INVESTIGACIÓN Y
DE ESTUDIOS AVANZADOS DEL
INSTITUTO POLITÉCNICO
NACIONAL

COORDINACIÓN GENERAL DE
SERVICIOS BIBLIOGRÁFICOS

Centro de Investigación y de Estudios Avanzados del I.P.N.
Unidad Guadalajara

Software para el análisis de QoS en VoIP

Tesis que presenta:

Jesús Antonio Argáez Xool

para obtener el grado de:

Maestro en Ciencias

en la especialidad de:

Ingeniería Eléctrica

Director de Tesis

Dr. Deni Librado Torres Román

**CINVESTAV
IPN
ADQUISICION
DE LIBROS**

Guadalajara, Jalisco, Julio de 2009.

CLASIF.: <u>TR 16568 .A75 2009</u>
ADQUIS.: <u>SSI-569</u>
FECHA: <u>19/11/2010</u>
PROCED.: <u>Don - 2010</u>
\$ <u> </u>

163327-1001

Software para el análisis de QoS en VoIP

**Tesis de Maestría en Ciencias
Ingeniería Eléctrica**

Por:

Jesús Antonio Argáez Xool

Ingeniero en Sistemas Computacionales

Universidad Autónoma de Campeche 2001-2006

Becario de CONACYT, expediente no. 202959

Director de Tesis

Dr. Deni Librado Torres Román

CINVESTAV del IPN Unidad Guadalajara, Julio de 2009.

AGRADECIMIENTOS

A DIOS, quien me permitió estar en esta vida llena de retos, logros y satisfacciones.

A mis PADRES, a quienes les debo la vida, todo lo que ahora soy y he logrado. Siempre han sido el cimiento de todos mis logros y sé que siempre contaré con su apoyo incondicional.

A mi ASESOR, Dr. Deni Librado Torres Román, por su gran apoyo y guía para el desarrollo de este trabajo.

A mi NOVIA, Gloria Margarita Uc Cahuich, porque siempre me apoya y me impulsa a seguir adelante y superarme.

A mi FAMILIA, quien también siempre me ha apoyado y alentado.

Le quiero agradecer al Consejo Nacional de Ciencia y Tecnología (CONACyT) por apoyarme económicamente durante la realización de mis estudios, de igual forma le agradezco al Centro de Investigación y de Estudios Avanzados del IPN (CINVESTAV) por todas las facilidades proporcionadas durante mi estancia en el centro.

RESUMEN

En esta tesis se describe el proceso de desarrollo de un sistema de software, denominado VoIPAS 1.0, que se encarga de realizar una comunicación de VoIP, determinar su calidad de servicio y calcular los siguientes parámetros:

1. Retardos
 - a. Unidireccional(OWD, One Way Delay)
 - b. Bidireccional (RTT, Round Trip Time)

2. Pérdida de Paquetes
 - Acumulativa
 - Ráfagas

3. Estimación del Parámetro de HURST de:
 - OWD
 - RTT

Mediante los siguientes métodos: R/S Statistic, Absolute Moment, Variance Method, Variance of Residuals, Modified Allan Variance, Periodogram, LWhittle, Abry-Veitch Method

4. MOS (Mean Opinion Score) basado en:
 - OWD
 - RTT/2

Una de las principales ventajas de este sistema de VoIP es la automatización de las mediciones de tráfico IP y el ahorro de tiempo y esfuerzo humano que por lo general son invertidas en la realización de este tipo de mediciones.

Esta aplicación es totalmente independiente, ya que una vez que se configuren las mediciones que se deseen realizar esta las hará de forma continua y automática, sin necesidad de intervención humana, y al término de cada una de ellas guardará las trazas de los diferentes parámetros en una ruta previamente configurada.

La aplicación nos permitirá dedicarnos de manera más eficiente al estudio y mejora de la calidad de servicio de VoIP. También podrá servir como plataforma de desarrollo, y permitirá probar modelos de tráfico.

ABSTRACT

In this thesis the development process of a software system, called VoIPAS 1.0 is described, this software system establish a VoIP communication, determinate quality of service and to calculate the following parameters:

5. Delays

a. OWD (One Way Delay)

b. RTT (Round Trip Time)

6. Packet Loss

- Accumulative
- Burst

7. HURST Parameter estimation:

- OWD
- RTT

By means of the following methods: R/S Statistic, Absolute Moment, Variance Method, Variance of Residuals, Modified Allan Variance, Periodogram, LWhittle, Abry-Veitch Method

8. MOS (Mean Opinion Score) based on:

- OWD
- RTT/2

One of the main advantages of this VoIP system is the automatization of IP traffic measurements and the saving time and human effort invested in the accomplishment of this type of measurements.

This application is completely independent, because once you configure to realize measurements, this application will do of continuous and automatic form, without needing human intervention, and at the end of each measurement it will save the traces of the different parameters.

The application will we allow to realize of more efficient way the study and improves of the quality of services on VoIP systems. Also it will be able to serve like development platform, and prove traffic models.

ÍNDICE

1	INTRODUCCIÓN	1
1.1	HERRAMIENTAS DE COMUNICACIÓN	2
2	VOZ SOBRE IP	6
2.1	CODIFICACIÓN DE VOZ	6
2.1.1	CODIFICADORES DE FORMA DE ONDA	6
2.1.2	CODIFICACIÓN PARAMÉTRICA	7
	• VOCODER POR PREDICCIÓN LINEAL [1]	7
	• LPC-10 (FS-1015) [2]	7
2.1.3	CODIFICACIÓN HÍBRIDA	7
	• CODIFICACIÓN RELP [3]	7
	• CODIFICACIÓN MULTIPULSO, MPC [3]	7
	• CODIFICACIÓN CELP (FS-1016) [4]	7
	• CODIFICACIÓN VSELP [5]	7
	• CODIFICACIÓN RPE-LTP [6]	7
2.2	PROTOCOLOS DE INTERNET	8
2.2.1	IP	9
2.2.2	TCP	10
2.2.3	UDP	10
2.2.4	RTP	11
2.2.5	RTCP	11
2.3	PROTOCOLOS DE VOIP	13
2.3.1	H.323	13
2.3.2	SIP	20
2.4	RESUMEN	24
3	QOS EN SISTEMAS VOIP	25
3.1	QoS	26
3.2	METRICAS DE DESEMPEÑO	28
3.2.1	DELAY	28
3.2.2	JITTER	31
3.2.3	PERDIDA DE PAQUETES	33
3.2.4	ANCHO DE BANDA	35
3.3	CALIDAD DE VOZ	38
3.3.1	MOS	38

3.3.2	MODELO E	39
3.4	RESUMEN	42
4	<u>ESPECIFICACIÓN DE REQUERIMIENTOS</u>	43
4.1	ESPECIFICACIÓN DE REQUERIMIENTOS DEL SISTEMA VOIP	44
4.1.1	REQUERIMIENTOS FUNCIONALES	44
4.1.2	REQUERIMIENTOS NO FUNCIONALES	51
5	<u>DISEÑO DEL SISTEMA</u>	52
5.1	CASOS DE USO	53
5.2	ARQUITECTURA DEL SISTEMA DE VOIP	64
5.3	DISEÑO ORIENTADO A OBJETOS	67
5.4	COBERTURA DE REQUERIMIENTOS	70
5.5	INTERFAZ DEL SISTEMA	71
5.6	DOXYGEN Y LA DOCUMENTACIÓN DEL SISTEMA	79
5.7	CONCEPTOS MATEMÁTICOS	80
5.7.1	OWD	80
5.7.2	RTT	80
5.7.3	JITTER	81
5.7.4	ESTIMADORES DEL INDICE DE AUTO-SIMILITUD PARA SERIES DISCRETAS DE TIEMPO	81
5.7.5	MOS	90
6	<u>ANÁLISIS Y EVALUACIÓN DEL SISTEMA</u>	93
6.1	ANÁLISIS DE DESEMPEÑO	94
6.2	ANÁLISIS DE RESOLUCIÓN DE TIEMPO	96
6.2.1	TEMPORIZADORES	96
6.2.2	PRUEBA DE RESOLUCIÓN DE TIEMPO	98
7	<u>CONCLUSIÓN Y TRABAJO FUTURO</u>	106
8	<u>BIBLIOGRAFÍA</u>	108
9	<u>ANEXO A – DOCUMENTACIÓN DEL CÓDIGO</u>	111
10	<u>ANEXO B – PLAN DE PRUEBAS</u>	112
11	<u>ANEXO C – ESPECIFICACIÓN DE OPERACIONES</u>	118
12	<u>ANEXO D – ESPECIFICACIÓN DE DISEÑO DE PRUEBAS</u>	123

1 INTRODUCCIÓN

La importancia que están tomando hoy en día los sistemas de telecomunicaciones basados en IP, obliga a pensar en la implantación de tecnologías que se adapten a las necesidades de una estructura básica existente. Los sistemas de comunicaciones en vez de ser elementos independientes y aislados para atender un determinado tipo de comunicación, ahora tienden a transmitir cualquier tipo de información sobre los medios existentes. Así, sobre la LAN e Internet, mediante la adopción de ciertos estándares y la incorporación de algunos elementos, es posible enviar voz y datos, con la gran ventaja y ahorro que supone el utilizar la infraestructura existente.

Además, el crecimiento y fuerte implantación de las redes IP, en especial Internet, el desarrollo de técnicas avanzadas de digitalización de voz, mecanismos de control y priorización de tráfico, protocolos de transmisión en tiempo real, así como nuevos estándares que permiten la calidad de servicio en redes IP, han creado un entorno donde es posible transmitir voz sobre IP. Si a todo lo anterior, se le suma el fenómeno Internet, junto con el potencial ahorro económico que este tipo de tecnologías puede llevar acarreado, el Protocolo de Voz Sobre Internet es un tema estratégico en la actualidad.

La Red Telefónica Pública Conmutada, PSTN (Public Switching Telephone Network) por sus siglas en inglés, es un sistema de mayor costo que la Telefonía IP, pero esta se lleva a cabo a través de una red de conmutación de circuitos, por lo que nos garantiza cierta calidad de servicio (QoS), a diferencia de la Telefonía IP que se basa en una red de conmutación de paquetes, como lo es el Internet que solo nos puede ofrecer su servicio de mejor esfuerzo (best-effort), en donde tiene que competir por los recursos de la red y no nos garantiza la QoS.

Los factores más importantes que determinan la QoS de las redes de VoIP son:

- Delay o Retardo
- Jitter
- Pérdida de Paquetes
- Ancho de Banda

Por esto es importante el estudio de estos factores, y así elevar el grado de satisfacción de los usuarios de sistemas de VoIP.

1.1 HERRAMIENTAS DE COMUNICACIÓN

Hoy en día existen diversas herramientas que se basan en VoIP, entre las más populares tenemos:

Skype



Figura 1-1 Interfaz de Skype

Es un software para realizar llamadas sobre Internet (VoIP), fundada en 2003 por el sueco Niklas Zennström y el danés Janus Friis, creadores de Kazaa. El código y protocolo de Skype permanecen cerrados y propietarios, pero los usuarios interesados pueden descargar gratuitamente la aplicación del sitio oficial. Los usuarios de Skype pueden hablar entre ellos gratuitamente.

La aplicación también incluye una característica denominada SkypeOut que permite a los usuarios llamar a teléfonos convencionales, cobrándoseles diversas tarifas según el país de destino, incluyendo en algunos los teléfonos móviles, aunque puede llamarse a casi cualquier teléfono del mundo. Otra opción que brinda Skype es SkypeIn, gracias al cual ellos otorgan un número de teléfono para que desde un aparato telefónico en cualquier parte del mundo puedan contactarte a tu ordenador. Además, proveen de un servicio de buzón de voz gratuito. La interfaz de Skype es muy parecida a otro software de mensajería instantánea tales como MSN Messenger o Yahoo! Messenger, y de igual forma que en éstos es posible entablar una conversación de mensajes instantáneos con los usuarios del mismo software. Este programa también te permite enviar mensajes de voz en caso de que el usuario no se encuentre disponible.

Características

- Llamadas gratis a cualquier usuario de skype en el mundo.
- Calidad óptima de sonido en líneas de banda ancha, en líneas conmutadas, funciona pero en menor calidad.
- Funciona a través de firewalls y redes de Internet compartida (usadas en los EE de Enlaces).
- Llamadas encriptadas para mayor seguridad.
- Existe la posibilidad de llamar a teléfonos tradicionales con cobros diferenciados.

Windows Live Messenger



Figura 1-2 Interfaz de Windows Live Messenger

Es un programa de mensajería instantánea que provee herramientas básicas de mensajería instantánea.

El uso de este programa se reduce al chat sin distracciones, sin personalización, sin tantos contenidos y sin muchos otros detalles que caracterizan a su versión más famosa. Además, a diferencia de otras versiones, permite el uso del protocolo SIP.

Características

- Intercambio de mensajes instantáneos de texto y de voz.
- Videoconferencias con otros usuarios.
- Transferencia de archivos.
- Aplicaciones extras como pizarra para compartir dibujos con los contactos.
- Iconos gestuales.
- Juegos.
- Iconos animados.

Yahoo Mesenger

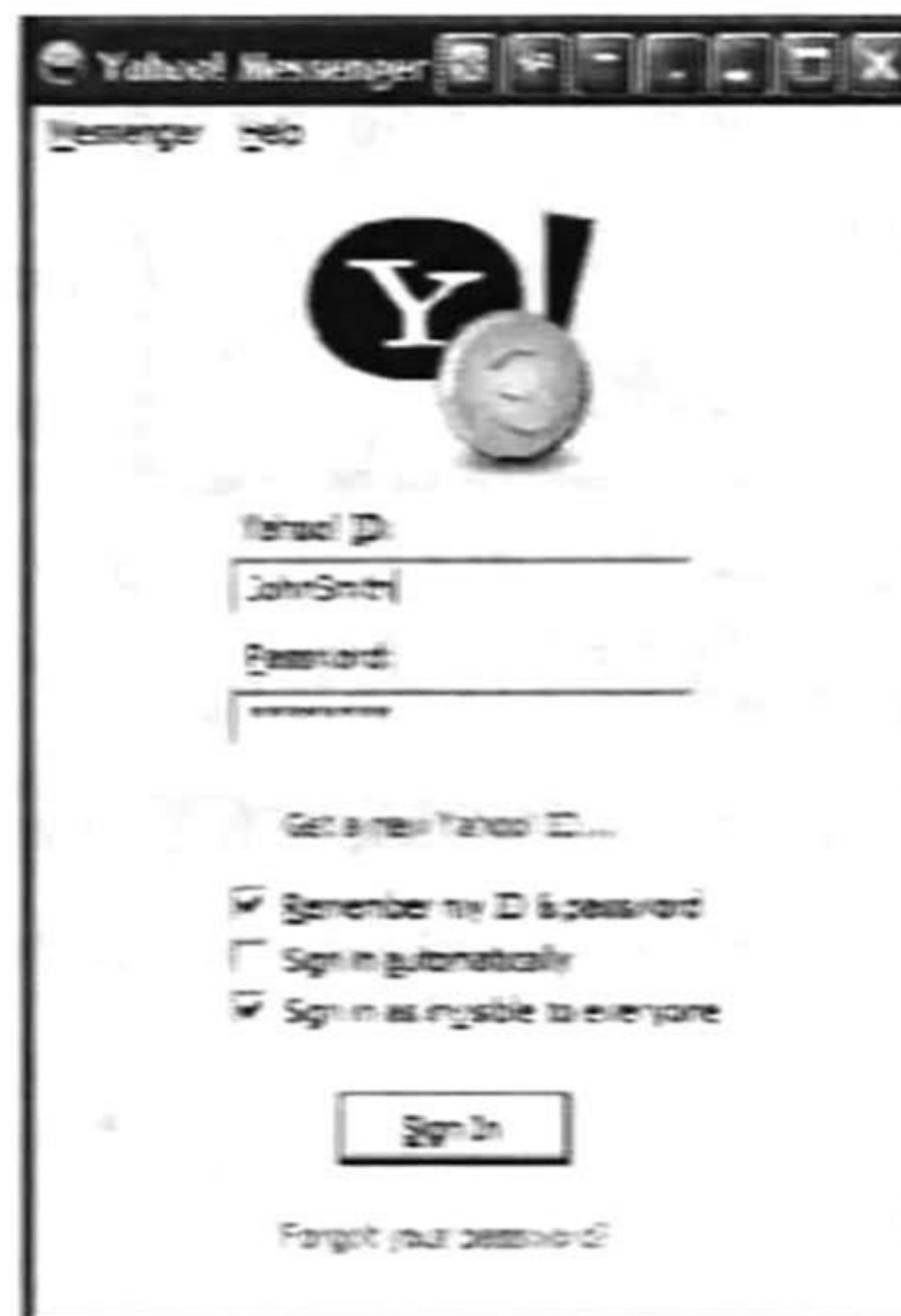


Figura 1-3 Interfaz de Yahoo Messenger

Yahoo! Messenger, también conocido como Yahoo! Instant Messaging y abreviado como YIM, es una aplicación que permite el intercambio en tiempo real de mensajes entre dos o más usuarios en forma de texto.

Características

Es uno de los sistemas de mensajería instantánea más comunes y populares, el servicio se hace cada vez más popular entre usuarios de todas las edades y niveles académicos por la facilidad de comunicación que permite entre dos personas sin

importar la distancia entre las mismas, dado que el servicio es completamente gratuito para cualquier persona con acceso a Internet.

Hoy en día, al igual que otros programas en esta categoría, extiende sus capacidades mucho más allá de un simple intercambio de mensajes de texto en tiempo real, dos usuarios pueden intercambiar también archivos de cualquier tipo por medio del programa, escuchar música mientras conversan e incluso utilizar una cámara web para ver en tiempo real a la persona con quien se conversa, entre otras utilidades.

El mensajero de Yahoo! permite mantener una lista de contactos de las personas que se han agregado a ella, haciendo de esta manera una fácil visualización de quien se encuentra conectado al sistema de mensajería en determinado momento. Además de una conversación usuario-usuario, Yahoo! Messenger permite crear una sala de conversación comúnmente conocida como chat donde se mantendrá una conversación instantánea entre más de dos usuarios.

En su última versión, permite la llamada a teléfonos fijos y móviles de todo el mundo, con una calidad de voz muy similar a la de Skype. A mediados de 2006 tanto la plataforma de MSN Messenger como la de Yahoo! se integraron buscando formar la red de mensajería instantánea más grande del mundo en su momento.

Como podemos observar existen diversas herramientas o software de comunicación que utilizan la tecnología de VoIP. Muchas de ellas solamente usadas para el ocio y otras como negocio.

Los objetivos de la presente tesis son:

- Desarrollar un sistema de comunicación de VoIP.
- Automatizar y Facilitar las mediciones de VoIP.
- Calcular y medir la QoS, así como los diferentes valores que la determinan, para su posterior estudio y mejora, durante la comunicación realizada con el sistema desarrollado.
- Contar con una plataforma para implementar nuevos codecs o teorías para el mejoramiento de la comunicación

Cabe mencionar que no se pretende competir con alguna de las herramientas antes mencionadas, si no desarrollar una herramienta de análisis y estudio para el laboratorio de Telecomunicaciones del Centro de Investigación y de Estudios Avanzados del IPN, Unidad Guadalajara.

2 VOZ SOBRE IP

2.1 CODIFICACIÓN DE VOZ

El proceso de codificación de voz permite transmitir y almacenar la señal de voz en forma digital eficientemente y sin pérdida de calidad. Desde el punto de vista de almacenar la señal de voz en formato digital, la codificación de voz permite minimizar el número de bits necesarios para el almacenamiento, manteniendo un nivel de calidad adecuado.

Existen 3 tipos de codificadores de voz: forma de onda, vocoders e híbridos; caracterizados por su "bit rate" y calidad de voz. La figura 2.1 muestra una comparativa entre los codificadores mencionados.

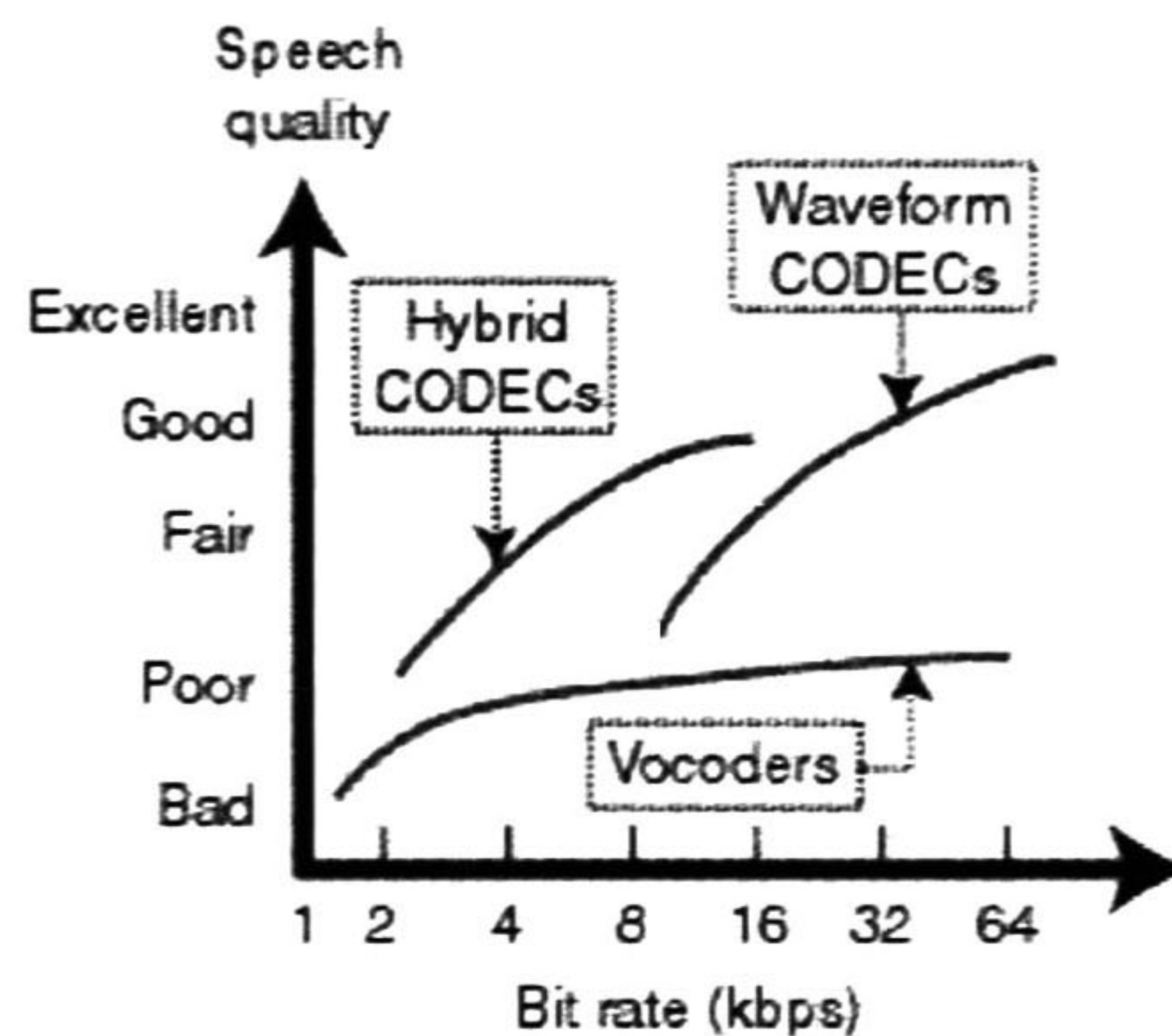


Figura 2-1 Comparativa de los distintos codificadores.

2.1.1 Codificadores de Forma de Onda

Los codificadores de forma de onda intentan reproducir la forma de la onda de la señal de entrada. Generalmente se diseñan para ser independientes a la señal, de tal forma que pueden ser usados para codificar una gran variedad de señales. Presentan una degradación aceptable en presencia de ruido y errores de transmisión. La codificación de forma de onda se divide en dos tipos:

- En el dominio del tiempo: PCM, DPCM y ADPCM
- En el dominio de la frecuencia: Subbandas y Transformadas.

2.1.2 Codificación Paramétrica

Los codificadores de forma de onda no toman en cuenta la naturaleza de la señal a codificar. Sin embargo, si codificamos una señal de voz, podemos aprovechar sus características intrínsecas para que la codificación se realice de forma más eficiente.

Es por eso que la codificación paramétrica intenta producir una señal que suene como la voz original, independientemente de si la forma de onda se parece o no. En el transmisor se analiza la voz y se extraen los parámetros del modelo y la excitación. Esta información se envía al receptor donde se sintetiza la voz. El resultado es que se produce voz inteligible a muy bajo "bit rate", pero tiene el problema de que no suena natural.

Unos ejemplos de vocoder son los siguientes:

- Vocoder por predicción lineal [1]
- LPC-10 (FS-1015) [2]

En resumen, su funcionamiento es el siguiente:

- Estimación de los parámetros del modelo
- Codificación y transmisión de los mismos
- Reconstrucción de la voz en el extremo receptor

2.1.3 CODIFICACIÓN HÍBRIDA

En la codificación híbrida se combinan las técnicas de los codificadores de forma de onda con las de los vocoders, con el propósito de obtener una alta calidad de voz a bajos "bit rates" (inferiores a 8 Kb/s). En estos codificadores, las muestras de la señal de entrada se dividen en bloques de muestras (vectores) que son procesados como si fueran uno solo. Llevan a cabo una representación paramétrica de la señal de voz para tratar que la señal sintética se parezca lo más posible a la original.

También se les conoce como codificadores de análisis-por-síntesis. En el emisor se lleva a cabo un análisis que obtiene los parámetros de la señal para luego sintetizarla y conseguir el mayor parecido a la original.

Estos son ejemplos de algunos codificadores:

- Codificación RELP [3]
- Codificación MULTIPULSO, MPC [3]
- Codificación CELP (FS-1016) [4]
- Codificación VSELP [5]
- Codificación RPE-LTP [6]

La familia de protocolos de Internet es un conjunto de protocolos de red en la que se basa Internet y que permiten la transmisión de datos entre redes de computadoras. En ocasiones se le denomina conjunto de protocolos TCP/IP, en referencia a los dos protocolos más importantes que la componen: Protocolo de Control de Transmisión (TCP) y Protocolo de Internet (IP), que fueron los dos primeros en definirse, y que son los más utilizados de la familia. Existen tantos protocolos en este conjunto que llegan a ser más de 100 diferentes, entre los cuales se encuentran UDP, RTP, HTTP, FTP, TELNET, SMTP, POP, POP3, etc.

La familia de protocolos de Internet puede describirse por analogía con el modelo OSI[7] el cual consta de diferentes capas, donde cada una de estas capas es responsable de una función diferente para comunicación, cada capa tiene una función bien definida:

1.- *El nivel Físico.*- Describe las características físicas de la comunicación, como las convenciones sobre la naturaleza del medio usado para la comunicación (como las comunicaciones por cable, fibra óptica o radio), y todo lo relativo a los detalles como los conectores, código de canales, modulación, potencias de señal, longitudes de onda, sincronización, temporización y distancias máximas. Este nivel asegura el transporte físico de los bits.

2.- *El nivel de Enlace de datos.*- Especifica cómo son transportados los paquetes sobre el nivel físico, incluyendo los delimitadores (patrones de bits concretos que marcan el comienzo y el fin de cada trama). Ethernet, por ejemplo, incluye campos en la cabecera de la trama que especifican que máquina o máquinas de la red son las destinatarias de la trama. Ejemplos de protocolos de nivel de enlace de datos son Ethernet, Wireless Ethernet, SLIP, Token Ring y ATM.

3.- *El nivel de Red.*- Maneja el movimiento de los paquetes en la red. Por ejemplo, el enrutamiento de los paquetes se lleva a cabo en esta capa. IP provee las funcionalidades de esta capa en la familia de protocolos de Internet.

4.- *El nivel de Transporte.*- Proporciona un flujo de datos entre dos sistemas que contienen las aplicaciones de la capa superior. Existen dos protocolos de transporte, TCP y UDP. TCP provee un flujo de datos confiable entre dos hosts. Le conciernen cosas como fragmentar los datos que le son pasados desde la aplicación a un tamaño apropiado para su manejo en la capa de red inferior, reconocimiento de los paquetes recibidos, y establecer tiempos de espera para asegurarse que el otro extremo de la comunicación pueda reconocer los paquetes enviados. Debido a que la confiabilidad del flujo de datos es controlado por la capa de transporte, la capa de aplicación puede

ignorar todos estos detalles. UDP, por otra parte, proporciona un servicio mucho más simple a la capa de aplicación. Envía paquetes de datos llamados *datagramas* de un host a otro, pero no existe garantía de que el datagrama alcance el otro extremo de la comunicación. Cualquier fiabilidad deseada debe ser añadida por la capa superior.

5.- *El nivel de Aplicación.*- El nivel de aplicación es el nivel que los programas más comunes utilizan para comunicarse a través de una red con otros programas. Los procesos que acontecen en este nivel son aplicaciones específicas que pasan los datos al nivel de aplicación en el formato que internamente use el programa y es codificado de acuerdo con un protocolo estándar.

2.2.1 IP

El Protocolo de Internet (IP, de sus siglas en inglés Internet Protocol) es un protocolo no orientado a conexión usado tanto por el origen como por el destino para la comunicación de datos a través de una red de conmutación de paquetes, este protocolo se encuentra descrito en el RFC 791 [8].

Los datos en una red basada en IP son enviados en bloques conocidos como paquetes o datagramas (en el protocolo IP estos términos se suelen usar indistintamente).

El Protocolo de Internet provee un servicio de datagramas no fiable, también llamado "best-effort", hará lo mejor posible pero la QoS no está garantizada. IP no provee ningún mecanismo para determinar si un paquete alcanza o no su destino, únicamente proporciona seguridad (mediante checksums o sumas de comprobación) de sus cabeceras y no de los datos transmitidos [7]. La tabla 2.1 nos muestra la estructura del protocolo IP.

0-3	4-7	8-15	16-18	19-31
Versión	Tamaño Cabecera	Tipo de Servicio	Longitud Total	
Identificador			Indicadores	Posición de Fragmento
Time To Live		Protocolo	Checksum Cabecera	
Dirección IP de Origen				
Dirección IP de Destino				
Opciones				Relleno

Tabla 2-1 Estructura de la Cabecera de Paquete IP

2.2.2 TCP

Es uno de los protocolos fundamentales en Internet de la capa de transporte orientado a la conexión [9]. Muchos programas dentro de una red de datos compuesta por computadoras pueden usar TCP para crear *conexiones* entre ellos a través de las cuales puede enviarse un flujo de datos. El protocolo garantiza que los datos serán entregados en su destino sin errores y en el mismo orden en que se transmitieron. También proporciona un mecanismo para distinguir distintas aplicaciones dentro de una misma máquina, a través del concepto de puerto. Su estructura se muestra en la tabla 2.2.

Bits 0 – 3	4 – 7	8 – 15	16 – 31	
0	Puerto Origen		Puerto Destino	
32	Número de Secuencia			
64	Número de Acuse de Recibo (ACK)			
96	longitud cabecera TCP	Reservado	Flags	Ventana
128	Suma de Verificación (Checksum)		Puntero Urgente	
160	Opciones + Relleno (opcional)			
224	Datos			

Tabla 2-2 Estructura de la Cabecera de Paquete TCP

2.2.3 UDP

Definida en el RFC 760 [10], UDP (User Datagram Protocol) es un protocolo del nivel de transporte, no orientado a conexión, basado en el intercambio de datagramas. Permite el envío de datagramas a través de la red sin que se haya establecido previamente una conexión, ya que el propio datagrama incorpora suficiente información de direccionamiento en su cabecera. Tampoco tiene confirmación, ni control de flujo, por lo que los paquetes pueden adelantarse unos a otros; y tampoco se sabe si ha llegado correctamente, ya que no hay confirmación de entrega o de recepción.

Su uso principal es para protocolos como DHCP, BOOTP, DNS y demás protocolos en los que el intercambio de paquetes de la conexión/desconexión son mayores, o no son rentables con respecto a la información transmitida, así como para la transmisión de audio y vídeo en tiempo real, donde no es posible realizar retransmisiones por los estrictos requisitos de retardo que se tiene en estos casos [7]. La estructura de los paquetes UDP [10] se muestran en la tabla 2-3.

+	Bits 0 - 15	16 - 31
0	Puerto origen	Puerto destino
32	Longitud del Mensaje	Suma de verificación
64	Datos	

Tabla 2-3 Estructura de los paquetes UDP

2.2.4 RTP

RTP son las siglas de Real-time Transport Protocol (Protocolo de Transporte de Tiempo Real) y definida en el RFC 1889 [11]. Es un protocolo de nivel de transporte utilizado para la transmisión de información en tiempo real, como por ejemplo audio y vídeo en un video-conferencia. En este protocolo se introducen los paquetes de VoIP. Su estructura se observa en la tabla 2-4 [12].

Byte 0		Byte 1		Byte 2		Byte 3	
V	P	X	CC	M	PT	Número de Secuencia	
Estampa de Tiempo							
Recurso de Sincronización (SSRC)							
Recurso de Contenido (CSRC)							
Cabecera de Extensión (EH - opcional)							
Datos							

Tabla 2-4 Estructura de los paquetes RTP.

2.2.5 RTCP

RTP Control Protocol (RTCP), definido en el RFC 3550 [12] es un protocolo de comunicación que proporciona información de control que está asociado con un flujo de datos para una aplicación multimedia (flujo RTP). Trabaja junto con RTP en el transporte y empaquetado de datos multimedia, pero no transporta ningún dato por sí mismo. Se usa habitualmente para transmitir paquetes de control a los participantes de una sesión multimedia de "streaming"

La función principal de RTCP es informar de la calidad de servicio proporcionada por RTP. Este protocolo recoge estadísticas de la conexión y también información como: bytes enviados, paquetes enviados, paquetes perdidos, jitter, entre otros. Una aplicación puede usar esta información para incrementar la calidad de servicio (QoS),

ya sea limitando el flujo o usando un códec de compresión más baja. En resumen, RTCP se usa para proveer información respecto a la QoS (Quality of Service).

2.3 PROTOCOLOS DE VOIP

La red telefónica pública conmutada (PSTN) está diseñada para la transmisión de voz, sin embargo tiene sus limitaciones tecnológicas. ¿Qué limitaciones?

Sin embargo los protocolos de las redes IP originalmente no fueron diseñados para el transporte en tiempo real de audio o cualquier otro tipo de medio de comunicación.

El objetivo de VoIP es transportar flujos de voz sobre paquetes bajo el protocolo IP en redes IP.

Es por lo anterior que se crean los protocolos para la transmisión de VoIP, cuyo mecanismo de comunicación consiste en un conjunto de mensajes de señalización entre terminales y canales lógicos para el flujo voz en tiempo real. Algunos protocolos utilizados en VoIP son:

- H.323 - Protocolo definido por la ITU-T
- SIP - Protocolo definido por la IETF
- IAX - Protocolo original para la comunicación entre PBXs Asterisk
- Skype - Protocolo propietario peer-to-peer utilizado en la aplicación Skype
- MGCP- Protocolo propietario de Cisco

2.3.1 H.323

Es un estándar creado por el grupo de estudio 16 de la ITU-T para la transmisión de voz, vídeo y datos multimedia a través de redes basadas en conmutación de paquetes sin calidad de servicio (QoS) garantizada, como las redes IP. H.323 es el estándar más utilizado actualmente en la telefonía IP [7][8].

La arquitectura de H.323 define un conjunto de componentes, protocolos, señalización, CODECs, etc., para llevar a cabo la comunicación y garantizar así la compatibilidad entre dispositivos [15].

H.323 soporta los siguientes protocolos [16]:

- H.225 para el direccionamiento (RAS: registro, admisión y estado) y señalización de llamada (H.225.0).
- H.235 para la seguridad y cifrado.
- H.245 para la señalización de control y la apertura/cierre de canales lógicos.
- H.450 para los servicios suplementarios.
- RTP/RTCP para el transporte de medios en tiempo real.
- T.120 como protocolo de datos para conferencia multimedia.
- CODECs de voz: G.711, G.722, G.723, G.728 y G.729.
- CODECs de video: H.261, H.263, H.264.

2.3.1.1 COMPONENTES DE UNA RED H.323

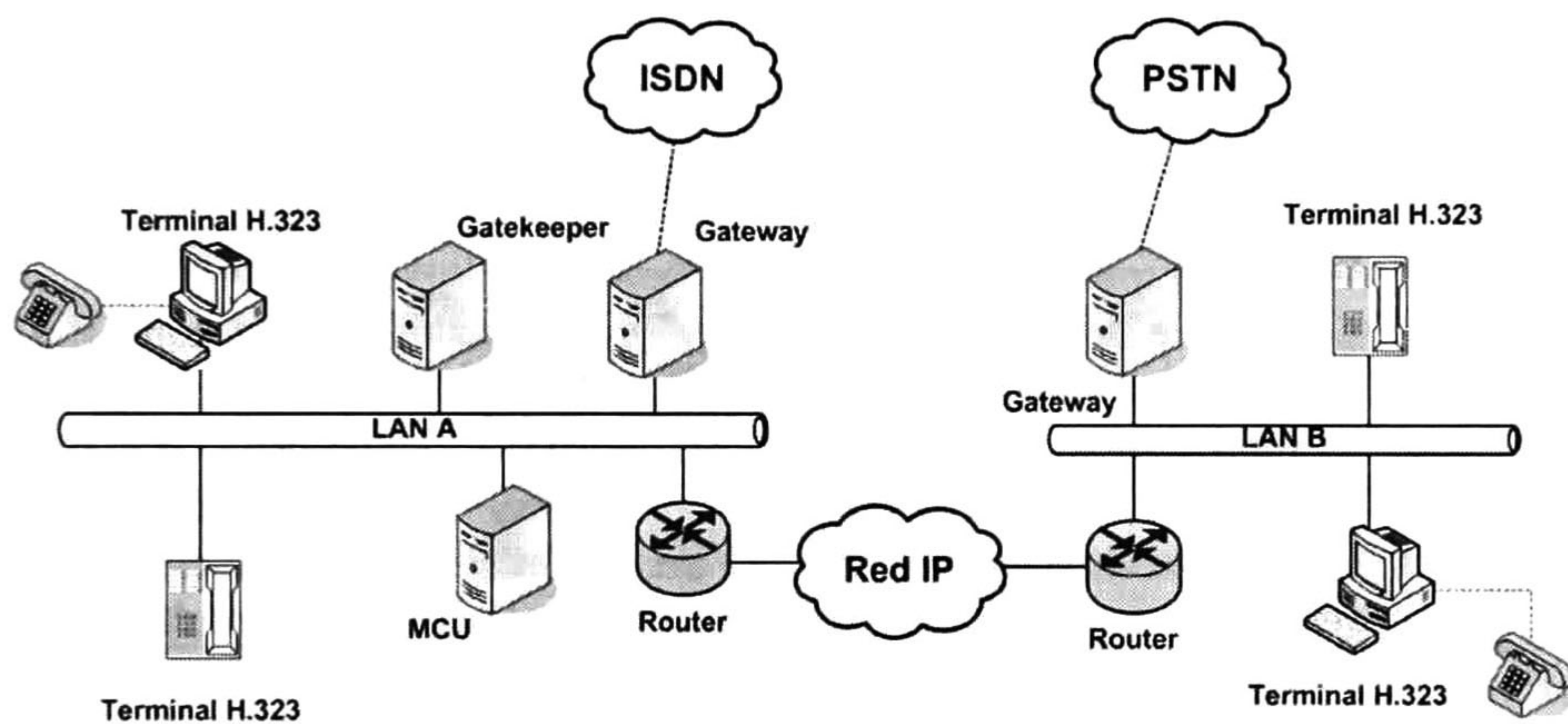


Figura 2-2 Arquitectura de una red H.323.

Las redes basadas en la recomendación H.323 constan principalmente de cuatro componentes:

- Los **terminales** H.323 permiten establecer comunicación bidireccional de audio y, opcionalmente, vídeo y datos. Cada terminal H.323 debe, como mínimo, soportar la codificación /decodificación de los formatos de audio empleados en las líneas telefónicas tradicionales (leyes a y μ) según G.711 (PCM, 8KHz@64kbps). El soporte de G.722, G.723.1, G.728 y G.729 es opcional. El soporte de vídeo no es imprescindible, aunque de existir, debe soportar al menos el códec H.261. Otras funcionalidades que opcionalmente puede incluir un terminal son las indicadas en el protocolo T.120 (transferencia de archivos y pizarra electrónica compartida, entre otras).

- Los **“gatekeepers”** se pueden considerar el punto central en la topología de una red H.323 y definen el concepto de zona H.323:

Una zona es un conjunto de MCUs, **“Gateway”** y terminales gestionados principalmente por un único **“gatekeeper”**; no obstante, en una zona pueden existir **“gatekeepers”** secundarios por si el **“gatekeeper”** primario fallara. Los **“gatekeepers”** no son necesarios para llamadas entre terminales H.323 dentro de una misma red, aunque sí lo son cuando se desea compatibilidad con las redes de telefonía. Es conveniente el uso de **“gatekeepers”**, puesto que proporcionan:

- Control del ancho de banda disponible en la red H.323, con el que el administrador puede limitar el número de conexiones simultáneas y así evitar problemas de congestión en la red que puedan reducir la calidad del servicio.
 - Conversión de números de teléfono estándar E164 a direcciones nativas H.323. Esta funcionalidad es imprescindible cuando se pretende establecer comunicación (a través de un **“Gateway”**) con la red telefónica tradicional.
 - Control de admisión de **“Gateway”** y terminales en una zona H.323, gestionado a través de mensajes H.225.0/RAS.
 - Gestión de componentes (terminales, **“Gateway”** y MCUs) pertenecientes a la zona. Opcionalmente, un **“gatekeeper”** puede ofrecer la señalización indirecta de llamadas entre terminales H.323 (enrutada a través del **“gatekeeper”**), restringir las llamadas que se pueden realizar, e incluso mantener una lista de las llamadas en espera.
- La **MCU** o unidad multipunto es un punto final encargado de dar soporte a las conferencias entre tres o más puntos finales H.323. Una MCU consta de un controlador multipunto (MC) y uno o más procesadores multipunto (MP).
 - Los **MCs**, que también pueden encontrarse integrados en terminales, **“Gateway”** o **“gatekeepers”**, se encargan de transmitir información de los CODECs soportados por los distintos terminales para poder así negociarlos y puedan ser utilizados durante la conferencia.
 - Los **MPs**, por su parte, distribuyen los flujos de audio / datos / vídeo entre los distintos terminales que participan en una multiconferencia.
 - Un **“Gateway”** permite conectar una red de conmutación de paquetes con una red de conmutación de circuitos. Sus dos funciones básicas son las de traducir los distintos protocolos de establecimiento y fin de llamada empleados por las distintas redes, y realizar la conversión de formatos de audio / vídeo oportuna.

2.3.1.2 PROTOCOLOS ESPECIFICADOS POR H.323

H.323 especifica los protocolos para el direccionamiento, señalización y control de llamada, CODECs de audio/video y normas de otros organismos que especifican la transmisión en tiempo real de los paquetes de voz. Estos protocolos funcionan sobre un nivel de transporte basado en TCP y UDP y/o (tras la 5ª revisión de H.323) SCTP. La figura 2.3 muestra la pila de protocolos H.323:



Figura 2-3 Pila de protocolos H.323.

- **H.225** Es un protocolo de comunicación de la familia de protocolos H.323, utilizados comúnmente para Voz sobre IP y para videoconferencia basada en IP.

El principal objetivo de H.225 es la definición de los siguientes mensajes:

- *Señalización de llamada*: establecimiento, control y finalización de una llamada H.323. La señalización H.225.0 está basada en los procedimientos de establecimiento de llamada de RDSI, Recomendación Q.931/Q932. Las señalizaciones de llamadas son enviadas sobre TCP por el puerto 1720.

Los mensajes más comunes de Q.931/Q.932 son:

- **Setup**. Es enviado para iniciar una llamada y establecer una conexión con una entidad H.323. Entre la información que contiene el mensaje se encuentra el puerto dirección IP y alias del llamante o la dirección IP y puerto del llamado.
 - **Call Proceeding**. Enviado por el "Gatekeeper" a un terminal advirtiéndolo del intento de establecer una llamada una vez analizado el número llamado.
 - **Alerting**. Indica el inicio de la fase de generación de tono.
 - **Connect**. Indica el comienzo de la conexión.
 - **Release Complete**. Enviado por el terminal para iniciar la desconexión.
 - **Facility**. Es un mensaje de la norma Q.932 usado como petición o reconocimiento de un servicio suplementario.
-
- *Señalización RAS*: lleva a cabo los procedimientos de registro, admisión y estado. La función de señalización RAS usa un canal separado (canal RAS).

Este conjunto de mensajes recibe el nombre de Registro, Admisión y Estado (del inglés Registration, Admission and Status - RAS).

Los mensajes son codificados de acuerdo a las Normas de Codificación de Paquetes (del inglés Packed Encoding Rules - PER) de la norma ASN.1.

La estructura de H.225.0 sigue el estándar Q.931 tal y como se ve en la siguiente figura:

8	7	6	5	4	3	2	1	Octeto
Discriminador de Protocolo								1
0	0	0	0	Longitud de bits de referencia de llamada				2
Valor de referencia de llamada								3 (-4)
0	Tipo de mensaje							
Elementos de información								
Estructura H.225								

Figura 2-4 Estructura de mensajes del estándar Q.931.

- **H.245** Es un protocolo de control de canal usado dentro de sesiones de comunicación H.323. También ofrece la posibilidad de ser encapsulado dentro de los mensajes de señalización de llamada de H.225.0. Esto facilita su paso a través de los “firewalls”

H.245 tiene la capacidad de transmitir y proporcionar la información necesaria para la comunicación multimedia, tal como la codificación, el control de flujo, la gestión de jitter y las peticiones de prioridad, así como el procedimiento de apertura y cierre de los canales lógicos usados para transmitir los flujos de medios.

Un problema grave que tenía la versión inicial de H.323 era el prolongado mecanismo de establecimiento de comunicación del protocolo H.245, necesario durante la apertura de los canales lógicos de una sesión telefónica, que era de cuatro vías. Versiones posteriores de H.323 introdujeron el procedimiento de Conexión Rápida, usando el elemento fastStart de un mensaje H.225.0. La Conexión Rápida redujo la negociación a tan solo dos vías. Existe otra recomendación, la H.460.6, que define el Procedimiento Extendido de Conexión Rápida, el cual fija un mecanismo de establecimiento de comunicación de una sola vía.

- **H.450** - Describe los Servicios Suplementarios.
- **H.235** - Describe la seguridad de H.323.
- **H.239** - Describe el uso de la doble trama en videoconferencia, normalmente uno para video en tiempo real y la otra para presentación.
- **H.281** - Describe el control de cámara lejana para movimientos PTZ (Pan-Tilt-Zoom)

En la figura 2.5 se muestran los mensajes de señalización de una llamada típica:

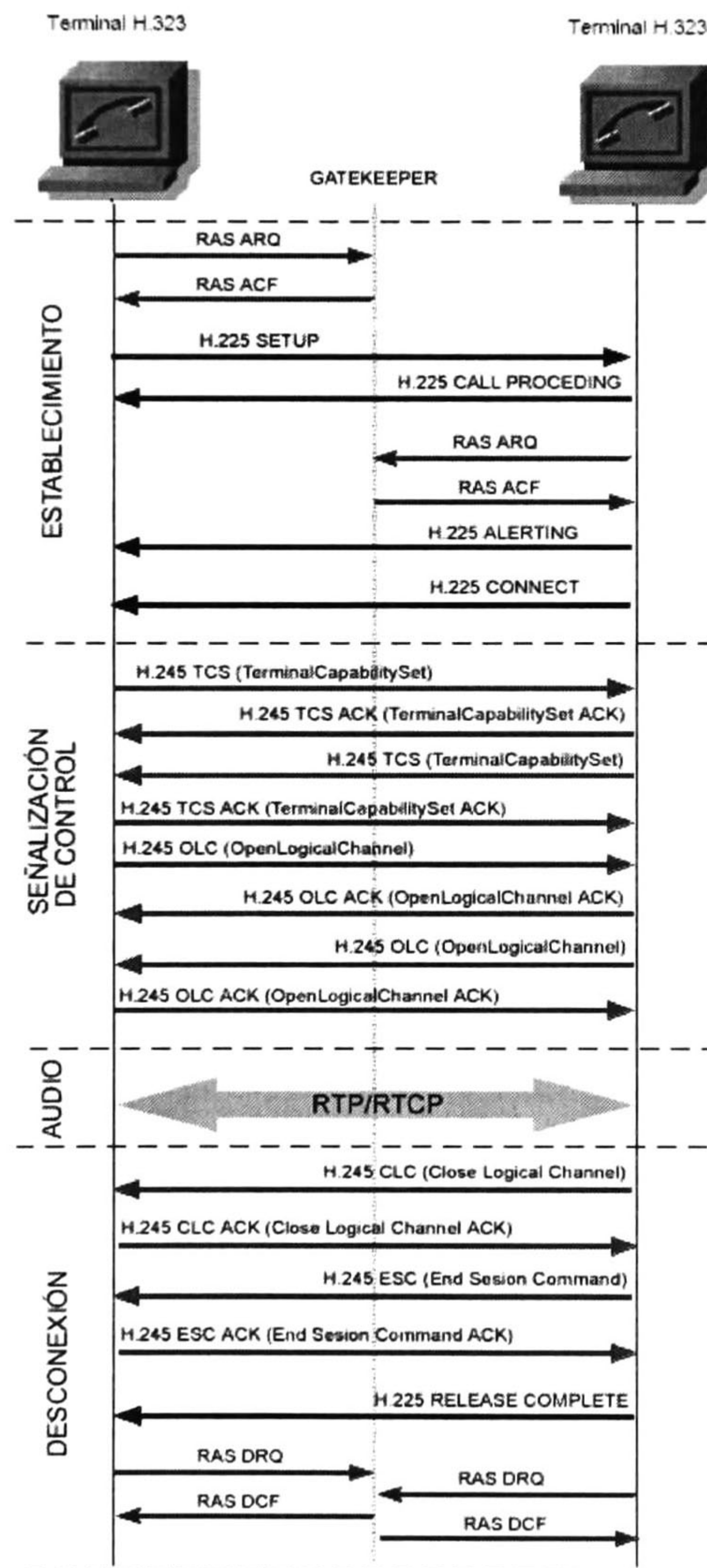


Figura 2-5 Señalización H.323

Una llamada H.323 se caracteriza por las siguientes fases:

- **Establecimiento**
 - En esta fase lo primero que se observa es que **uno de los terminales se registra en el "gatekeeper"** utilizando el protocolo RAS (Registro, admisión y estado) con los mensajes ARQ y ACF.
 - Posteriormente utilizando el protocolo H.225 (que se utiliza para establecimiento y liberación de la llamada) **se manda un mensaje de SETUP para iniciar una llamada H.323.**
 - **El terminal llamado contesta con un CALL PROCEEDING advirtiendo del intento de establecer una llamada.**
 - En este momento el segundo terminal tiene que registrarse con el "gatekeeper" utilizando el protocolo RAS de manera similar al primer terminal
 - El mensaje **ALERTING** indica el inicio de la fase de generación de tono.
 - Y por último **CONNECT** indica el comienzo de la conexión.

- **Señalización de Control**
 - En esta fase **se abre una negociación de capacidades entre terminales mediante el protocolo H.245**, establecen quién será maestro y quién esclavo, los CODECs de audio y video a utilizar. **Como punto final de esta negociación se abre el canal de comunicación** (direcciones IP, puerto).
 - Los principales mensajes H.245 que se utilizan en esta fase son:
 - **TerminalCapabilitySet (TCS).** Mensaje de intercambio de capacidades soportadas por los terminales que intervienen en una llamada.
 - **OpenLogicalChannel (OLC).** Mensaje para abrir el canal lógico de información que contiene información para permitir la recepción y codificación de los datos. Contiene la información del tipo de datos que será transportados.

- **Flujo RTP**

Los terminales inician la comunicación y el intercambio de audio (o video) mediante el protocolo RTP/RTCP.

- **Desconexión**
 - En esta fase cualquiera de los participantes activos en la comunicación puede iniciar el proceso de finalización de llamada **mediante mensajes CloseLogicalChannel y EndSessionComand de H.245.**
 - Posteriormente **utilizando H.225 se cierra la conexión con el mensaje RELEASE COMPLETE**

Por último **se liberan los registros con el "gatekeeper"** utilizando mensajes del protocolo RAS.

2.3.2 SIP

Definido en el RFC 3261 [17], es un protocolo de señalización (situado en el nivel ISO / OSI de aplicación) para el establecimiento, mantenimiento y terminación de sesiones interactivas de usuarios donde intervienen elementos multimedia como el video, voz, mensajería instantánea, juegos online y realidad virtual. SIP fue creado en 1996 por Mark Handley y Henning Schulzrinne, ha sido estandarizado por la IETF (Internet Engineering Task Force). La especificación más reciente de SIP se puede encontrar en el RFC 3261. SIP no define por si mismo un sistema de comunicaciones ni ofrece servicio alguno; es un protocolo flexible que se limita a ofrecer una serie de primitivas que las aplicaciones pueden utilizar para implementar servicios.

El protocolo SIP es solo un protocolo de señalización. Una vez la sesión establecida, los participantes de la sesión intercambian directamente su tráfico de audio / video a través del protocolo "Real-Time Transport Protocol" o RTP. Por otra parte, SIP no es un protocolo de reservación de recursos, y en consecuencia, no puede asegurar la calidad de servicio. Se trata de un protocolo de control de llamada y no de control del medio.

SIP ofrece funciones tales como:

- Redirección de llamadas
- Resolución de direcciones
- Determinar la disponibilidad de un punto final
- Establecer llamadas punto a punto o multipunto

2.3.2.1 COMPONENTES SIP

SIP define cuatro componentes lógicos. Estos componentes se pueden implementar en dispositivos físicos, tal como teléfonos IP, o bien como aplicaciones software; en cualquier caso un mismo dispositivo físico puede incluir uno o más componentes lógicos. Todos los elementos SIP deben implementar obligatoriamente TCP y UDP. En ambos protocolos escucha en el puerto 5060.

- El **Agente de Usuario** es una aplicación con arquitectura cliente / servidor que se utiliza para iniciar y terminar las sesiones. El cliente usuario-agente (UAC) se encarga de realizar peticiones SIP, mientras que el servidor usuario-agente (UAS) notifica al usuario cuando se recibe una petición y responde a dicha petición dependiendo de la acción tomada por el usuario.
- El **Servidor de Redirecciones** acepta una petición SIP y envía una respuesta al cliente que contiene las direcciones de los servidores con los que debe contactar el cliente.
- El **Servidor Proxy**, que contiene funciones de servidor y cliente, actúa como un intermediario que realiza peticiones en nombre de otros clientes: para ello

interpreta la cabecera del mensaje y la reescribe identificando al proxy como el que inicia la solicitud, recibe la respuesta del destinatario y se la reenvía al cliente.

- Un **Servidor de Registro** almacena (o actualiza) en una base de datos la información de contacto del usuario que realiza la petición.

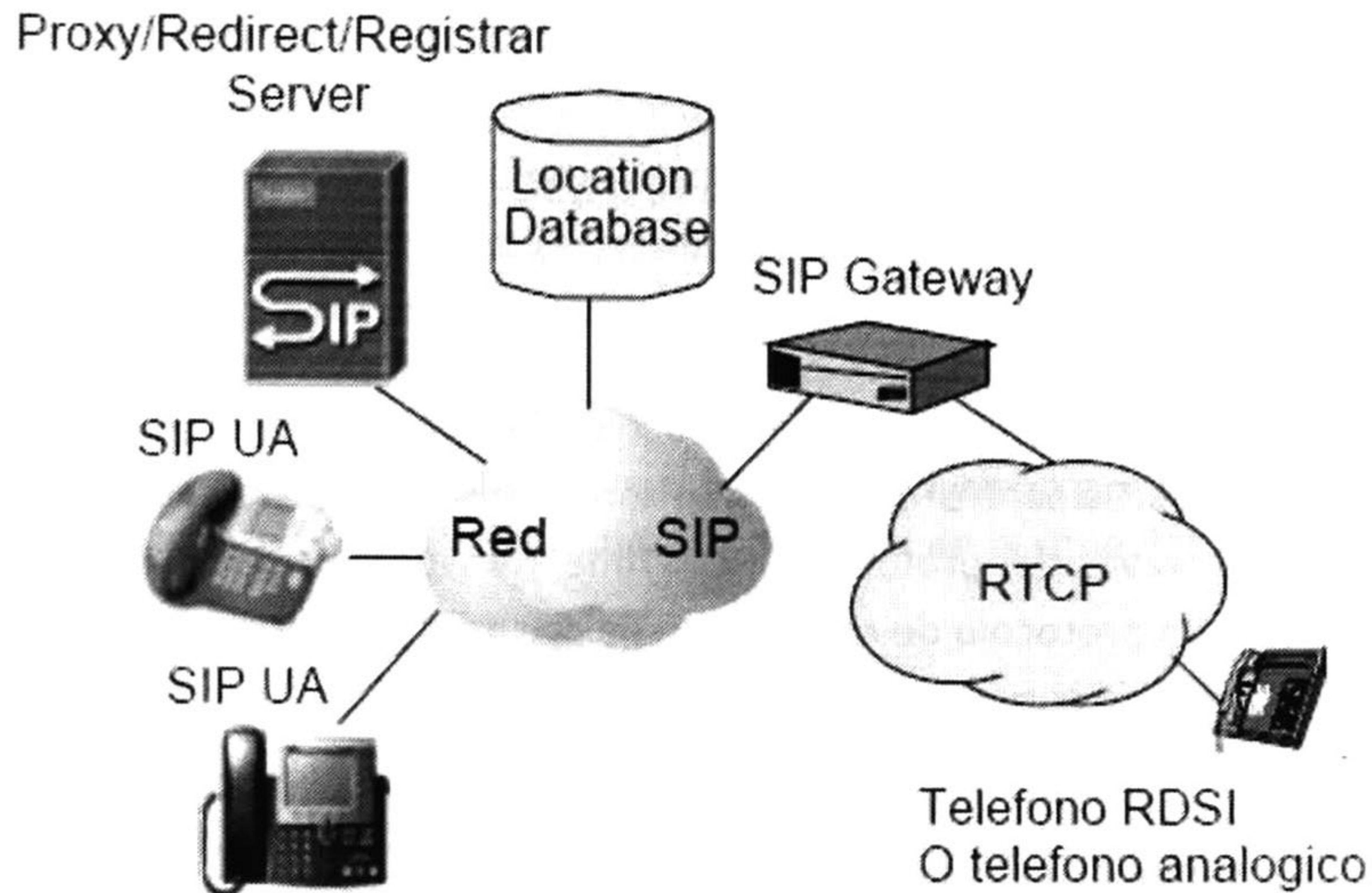


Figura 2-6 Componentes de una red SIP.

2.3.2.2 PROTOCOLOS ESPECIFICADOS POR SIP

SIP emplea SDP para descubrir las capacidades multimedia del punto final destino y suele utilizar RTP/RTCP para el transporte de voz.

- **SDP (Session Description Protocol)** SDP es el protocolo empleado para describir una sesión multimedia, que consiste en un conjunto de flujos de medios (audio, vídeo o datos) que existen durante un determinado tiempo. Los paquetes SDP contienen (entre otros campos) información acerca del ancho de banda, los protocolos de transporte empleados, los CODECs utilizados en la sesión, y la dirección de contacto del iniciador de la sesión.

2.3.2.3 SEÑALIZACIÓN SIP

SIP es un protocolo textual que usa una semántica semejante a la del protocolo HTTP. Los UAC realizan las peticiones y los UAS retornan respuestas a las peticiones de los clientes. **SIP define la comunicación a través de dos tipos de mensajes. Las solicitudes (métodos) y las respuestas (códigos de estado)** emplean el formato de mensaje genérico establecido en el RFC 2822 , que consiste en una línea inicial seguida de un o más campos de cabecera (headers), una línea vacía que indica el final de las cabeceras, y por último, el cuerpo del mensaje que es opcional.

- **Métodos SIP**

Las peticiones SIP son caracterizadas por la línea inicial del mensaje, llamada Request-Line, que contiene el nombre del método, el identificador del destinatario de la petición (Request-URI) y la versión del protocolo SIP. Existen seis métodos básicos SIP (definidos en RFC 254) que describen las peticiones de los clientes:

- **INVITE:** Permite invitar un usuario o servicio para participar en una sesión o para modificar parámetros en una sesión ya existente.
- **ACK:** Confirma el establecimiento de una sesión.
- **OPTION:** Solicita información sobre las capacidades de un servidor.
- **BYE:** Indica la terminación de una sesión.
- **CANCEL:** Cancela una petición pendiente.
- **REGISTER:** Registrar al User Agent.

- **Respuestas (Códigos de estado) SIP.**

Después de la recepción e interpretación del mensaje de solicitud SIP, el receptor del mismo responde con un mensaje. Este mensaje, es similar al anterior, difiriendo en la línea inicial, llamada Status-Line, que contiene la versión de SIP, el código de la respuesta (Status-Code) y una pequeña descripción (Reason-Phrase). El código de la respuesta está compuesto por tres dígitos que permiten clasificar los diferentes tipos existentes. El primer dígito define la clase de la respuesta.

Código	Clases
1xx	Mensajes provisionales.
2xx	Respuestas de éxito
3xx	Respuestas de redirección.
4xx	Respuestas de fallo de método.
5xx	Respuestas de fallos de servidor.
6xx	Respuestas de fallos globales.

Tabla 2-5. Clase de respuestas de estado del protocolo SIP

A continuación se analizará detalladamente una llamada. En una llamada SIP hay varias transacciones SIP. Una transacción SIP se realiza mediante un intercambio de mensajes entre un cliente y un servidor.

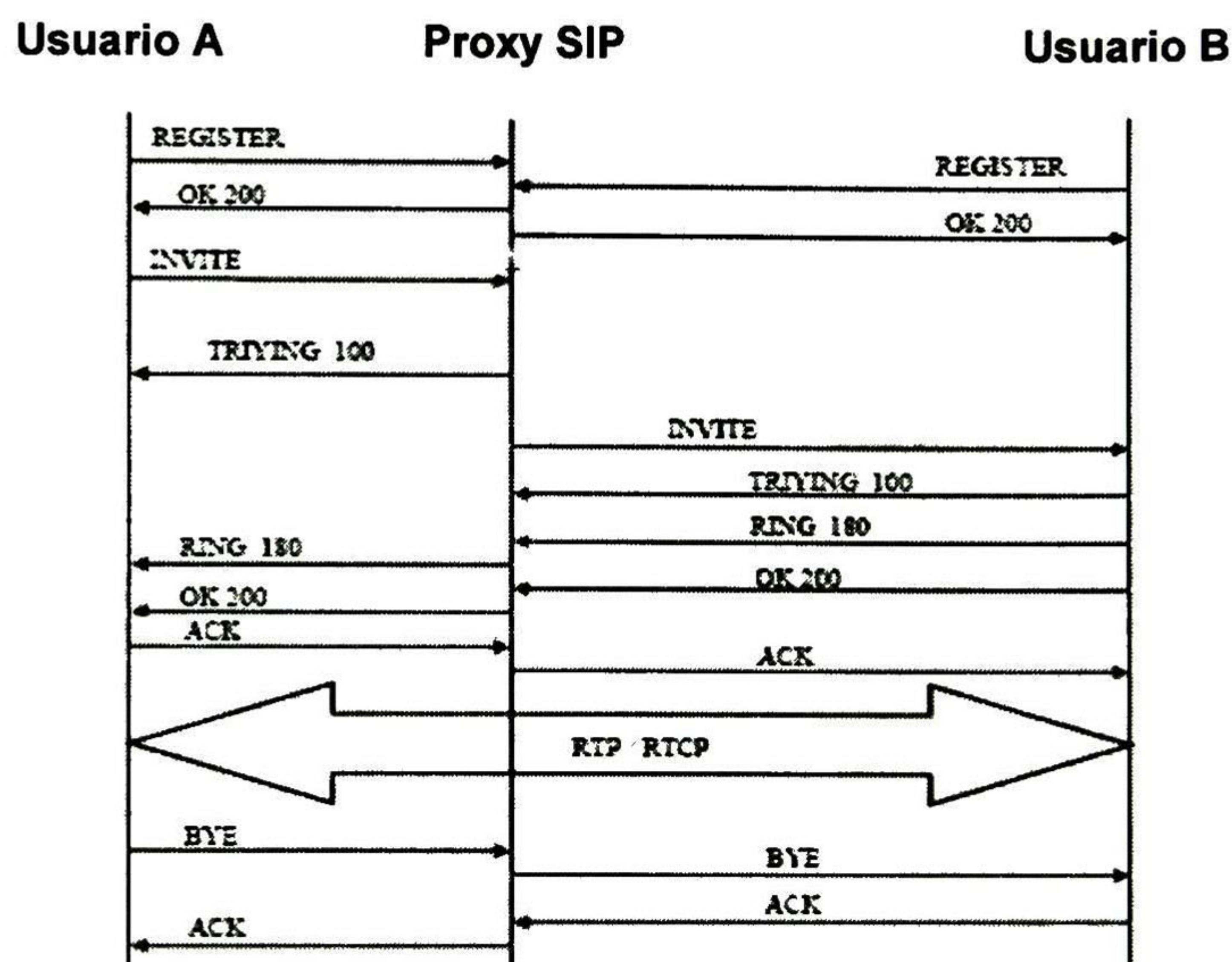


Figura 2-7 Señalización del protocolo SIP.

- **Las dos primeras transacciones corresponden al registro de los usuarios.** Los usuarios deben registrarse para poder ser encontrados por otros usuarios. En este caso, los terminales envían una petición REGISTER, donde los campos "from" y "to" corresponden al usuario registrado. El servidor Proxy, que actúa como Register, consulta si el usuario puede ser autenticado y envía un mensaje de OK en caso positivo.
- **La siguiente transacción corresponde a un establecimiento de sesión.** Esta sesión consiste en una petición INVITE del usuario al proxy. Inmediatamente, el proxy envía un TRYING 100 para parar las retransmisiones y reenvía la petición al usuario B. El usuario B envía un Ringing 180 cuando el teléfono empieza a sonar y también es reenviado por el proxy hacia el usuario A. Por último, el OK 200 corresponde a aceptar la llamada (el usuario B descuelga).
- **En este momento la llamada está establecida, pasa a funcionar el protocolo de transporte RTP** con los parámetros (puertos, direcciones, CODECs, etc.) establecidos en la negociación mediante el protocolo SDP.
- **La última transacción corresponde a una finalización de sesión.** Esta finalización se lleva a cabo con una única petición BYE enviada al Proxy, y posteriormente reenviada al usuario B. Este usuario contesta con un OK 200 para confirmar que se ha recibido el mensaje final correctamente.

En este capítulo se definen los diferentes tipos de codificación de voz (forma de onda, paramétrica e híbrida), los protocolos IP (Protocolo de Internet-IP, Protocolo de Control de Transmisión-TCP, Protocolo de Datos de Usuario-UDP, Protocolo de Transporte en Tiempo Real-RTP, Protocolo de Control de Transporte en Tiempo Real-RTCP) que permiten empaquetar la voz para que pueda ser transmitida como los datos y se describen las principales características de las tecnologías de conmutación de circuitos y paquetes. La voz sobre el protocolo de Internet es una tecnología que permite transmitir voz sobre redes de conmutación de paquetes soportadas bajo la pila de protocolos TCP/IP.

Se definen y comparan, los estándares de VoIP (H.323, SIP) que en conjunto con los protocolos IP, hacen posible la transmisión de voz sobre Redes IP.

3 QOS EN SISTEMAS VOIP

El auge de la telefonía IP es evidente debido al reaprovechamiento de los recursos y disminución en el costo de llamadas a través de Internet.

La VoIP adolece de la calidad de los sistemas telefónicos tradicionales, la cual es transmitida a través de una red de conmutación de circuitos que ofrece una cierta cantidad fija de ancho de banda lo que no hace el internet; aunque hoy en día la infraestructura y la tecnología han ido aumentando el ancho de banda disponible en internet. En la calidad de una llamada telefónica influyen diversos factores como son: el codec, la paquetización, pérdida de paquetes, retardos, jitter, y la arquitectura de la red.

La calidad del servicio en VoIP deriva de dos factores principalmente:

a) **Internet es un sistema basado en conmutación de paquetes no orientado a la conexión** y por tanto la información no viaja siempre por el mismo camino. Esto produce efectos como la pérdida de paquetes o el jitter.

b) **Las comunicaciones VoIP son en tiempo real** lo que produce que efectos como el eco, la pérdida de paquetes y el retardo o latencia sean muy molestos y perjudiciales y deben ser evitados, controlados, o minimizados.

3.1 QoS

La satisfacción de un producto o servicio es directamente proporcional a la calidad que este mantiene u ofrece. La calidad y el costo son factores importantes que afectan el atractivo de cualquier servicio, es por todo esto que es importante determinar la calidad de servicio (QoS- Quality of Service).

La VoIP ha hecho del monitoreo de la QoS un elemento importante para ofrecer una mejor calidad.

La comunicación de VoIP es montada a través de una red de conmutación de paquetes, de recursos compartidos y no garantiza una cierta calidad de servicio, solo nos garantiza el mejor esfuerzo (best effort), lo cual presenta nuevos retos a enfrentar.

En las comunicaciones existen diferentes tipos de aplicaciones, las cuales tienen sus características y sus propias demandas para garantizar una cierta calidad de servicio. La fig. 3.1 muestra los tipos de aplicaciones y sus diferentes demandas.

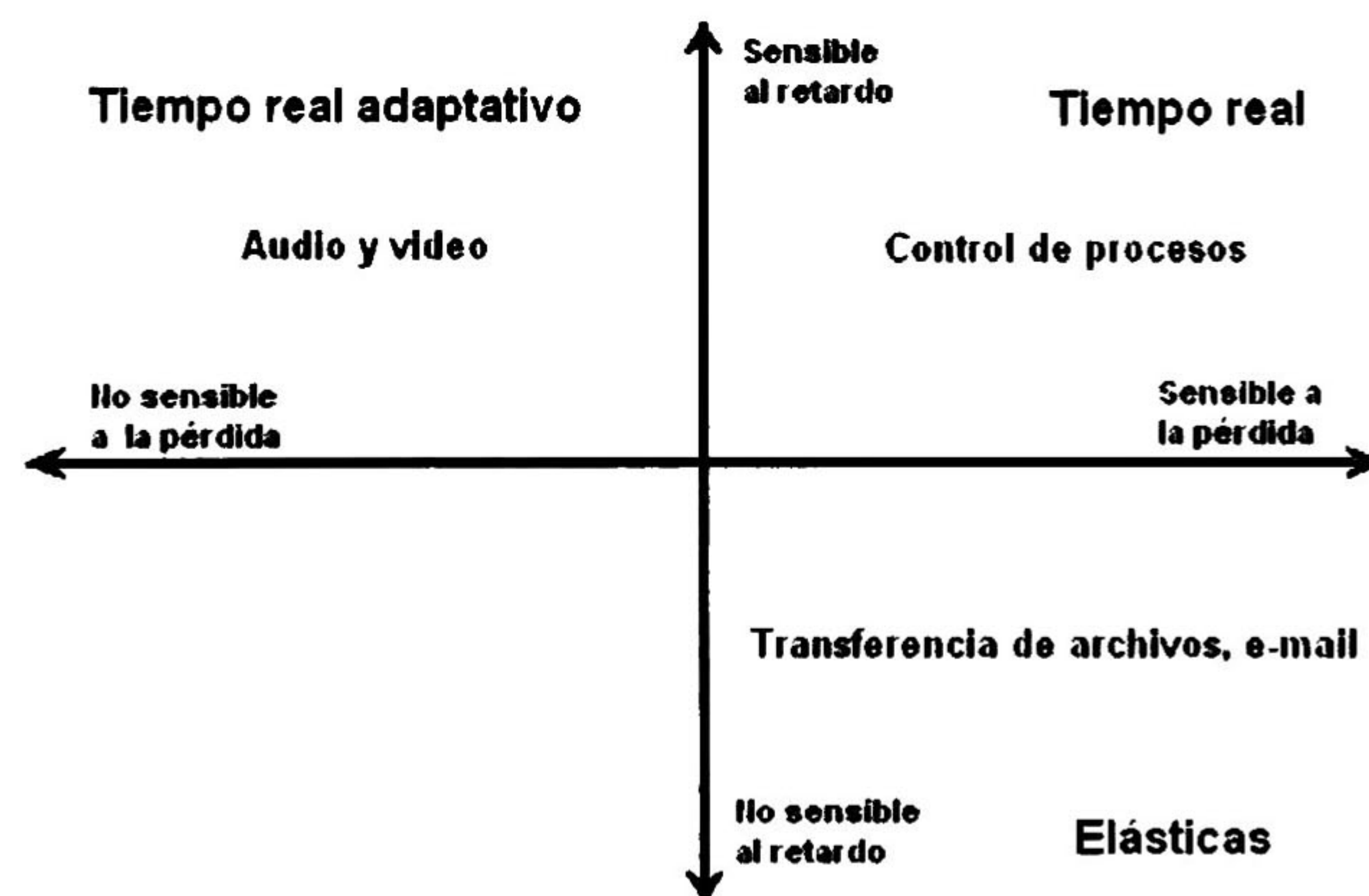


Figura 3-1 Demandas de los diferentes tipos de aplicaciones.

La comunicación de VoIP es una aplicación de tiempo real, y como se muestra en la figura 3.1, es sensible al retardo y a la pérdida de paquetes

En la Tabla 3.1 se muestra una comparativa entre las diferentes aplicaciones que existen y compiten en internet.

Aplicación	Dimensiones de Desempeño			
	Ancho de Banda	Sensibilidad a		
		Delay	Jitter	Pérdida
Telefonía IP	Bajo	Alto	Alto	Alto
Video Conferencias	Alto	Alto	Alto	Medio
Flujo media	Bajo- Alto	Alto	Bajo	Medio
Transacciones Cliente/Servidor	Bajo	Alto	Bajo	Alto
Email	Bajo	Bajo	Bajo	Alto
Tráfico BestEffort	Bajo - Medio	Bajo	Bajo	Bajo

Tabla 3-1 Demandas de Aplicaciones en Internet.

3.2 METRICAS DE DESEMPEÑO

Para proporcionar un buen servicio se debe proporcionar una buena calidad, y dicha calidad satisfacer ciertos criterios o garantizar ciertos parámetros para indicar si la calidad es buena o mala.

En VoIP se estudian principalmente 4 métricas que determinan la calidad y el desempeño de la comunicación:

- Retardo o Delay
- Jitter
- Pérdida de paquetes
- Ancho de banda

3.2.1 DELAY

El retardo en el Internet consta de 4 sub-métricas:

- *Retardo de Propagación:* es el tiempo que tarda un bit en transmitirse de un punto a otro. Es una cantidad casi constante y depende principalmente de la velocidad de propagación de la señal en el medio de transmisión.
- *Retardo de procesamiento:* es el intervalo de tiempo desde que el paquete arriva al router o switch hasta que es puesto en una cola para su transmisión.
- *Retardo de encolamiento:* es el tiempo que el paquete espera en la cola hasta que es puesto en un puerto para su transmisión.
- *Retardo de transmisión:* es el tiempo que tarda en enviarse un paquete desde el primer bit hasta el último.

Usualmente se toman solamente en cuenta los últimos dos componentes, aunque el retardo de procesamiento ya no es despreciable. En general se estudian dos tipos de Retardos el OWD (Retardo en un sentido) y el RTT (Retardo ida y vuelta).

3.2.1.1 OWD

El retardo en un sentido (OWD) es el tiempo que transcurre entre la ocurrencia de dos eventos de referencia de paquetes IP correspondientes, el evento de ingreso IPRE1 en el momento t_1 y el evento de egreso IPRE2 en el momento t_2 [18]. El IPTD extremo a extremo es un evento unidireccional entre el punto de medición (MP) de la fuente y el MP del destino.

$$OWD = t_2 - t_1 \geq T_{max} \quad t_2 > t_1 \quad \text{Ecuación 3-1}$$

donde T_{max} es el retardo máximo del paquete IP a partir del cual se considera que el paquete se ha perdido.

La figura 3.2 muestra un diagrama del OWD.

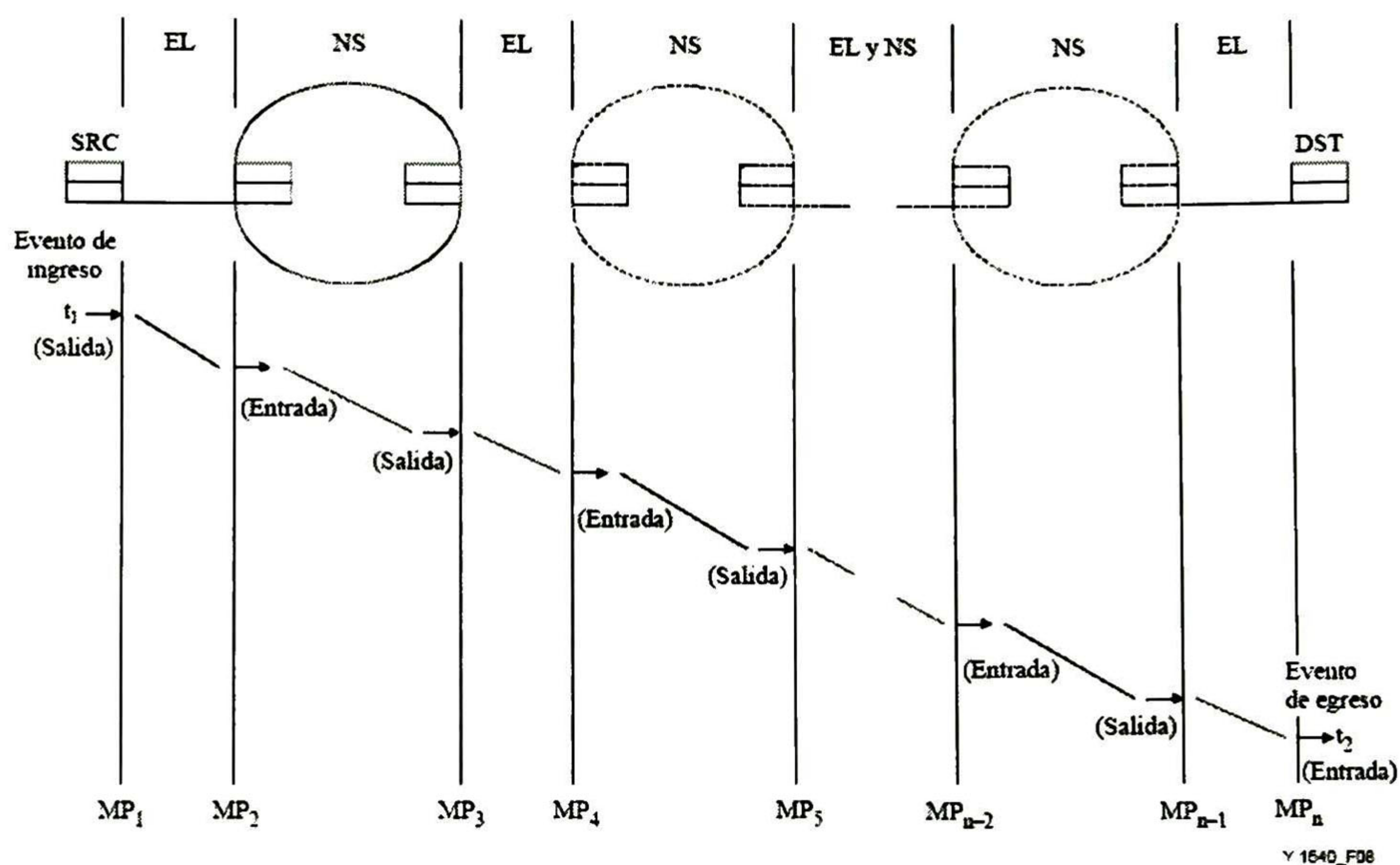


Figura 3-2 OWD

El ITU-T G.114 recomienda los siguientes límites para el tiempo de transmisión en un solo sentido [19], en conexiones que cuenten con un control de eco adecuado conforme a la Recomendación G.131:

- De 0 a 150 ms: aceptable para la mayoría de las aplicaciones de usuario.
- De 150 a 400 ms: aceptable siempre y cuando las administraciones conozcan la influencia del tiempo de transmisión en la calidad de transmisión de las aplicaciones de usuario (véase la Nota 3).
- Por encima de 400 ms: inaceptable a efectos de planificación general de la red; se acepta, sin embargo, que este límite pueda ser rebasado en ciertos casos excepcionales.

Pueden utilizarse los valores de planificación indicados en la Tabla 3.2 para evaluar el tiempo de transmisión total de conjuntos específicos de elementos que pueden constituir circuitos o conexiones.

Tabla 3-2 Valores de Planificación del Tiempo de Transmisión [19].

Sistema de transmisión o de procesamiento	Contribución al tiempo de transmisión en un sentido	Observaciones
Sistema de cable coaxial terrenal o radioenlace; transmisión FDM y digital	4 μ s/km	
Sistema de cable de fibra óptica; transmisión digital	5 μ s/km	Se tiene en cuenta el retardo en repetidores y regeneradores
Sistema de cable coaxial submarino	6 μ s/km	
Sistema de cable coaxial terrenal o radioenlace; transmisión FDM y digital	4 μ s/km	
Sistema de cable de fibra óptica; transmisión digital	5 μ s/km	Se tiene en cuenta el retardo en repetidores y regeneradores
Sistema de cable coaxial submarino	6 μ s/km	
Sistema de fibra óptica submarino: – terminal transmisor; – terminal receptor.	13 ms 10 ms	Caso más desfavorable
Sistema por satélite: – a 1400 km de altitud; – a 14 000 km de altitud; – a 36 000 km de altitud.	12 ms 110 ms 260 ms	Propagación en el espacio exclusivamente (entre estaciones terrenas)
Equipo de modulación o demodulación de canal FDM	0,75 ms	Semisuma de los tiempos de transmisión en ambos sentidos de transmisión
Equipo de modulación o demodulación de canal FDM con compansor	0,5 ms	
Codificador y decodificador MIC (Rec. G.712)	0,75 ms	
Codificador y decodificador MICDA (Recs. G.721, G.726, G.727)	0,25 ms	
Transcodificación MIC/MICDA/MIC	0,5 ms	
Codificador y decodificador LD-CELP (Rec. G.728)	2,0 ms	
Codificador y decodificador CS-ACELP a 8 kbit/s; (Rec. G.729)	15 ms	
Sistema móvil terrestre público (PLMS) – objetivo 40 ms,	80-110 ms	
Codificadores y decodificadores de vídeo de la serie de Recomendaciones H.260	En estudio	
DCME (Rec. G.763), por par: – con señales vocales y datos en la banda vocal sin remodulación; – con datos en la banda vocal con remodulación.	30 ms 350 ms	
Equipo MIC (Rec. G.764), por par: – con señales vocales y datos en la banda vocal sin remodulación; – con datos en la banda vocal con remodulación.	35 ms 70 ms	
Transmultiplexor	1,5 ms	
Central de tránsito digital, digital-digital	0,45 ms	
Central local digital, analógico-analógico	1,5 ms	
Central local digital, línea de abonado analógica-enlace digital	0,975 ms	
Central local digital, línea de abonado digital-enlace digital	0,825 ms	
Compensadores de eco	0,5 ms	
Modo de transferencia asíncrono ATM (CBR utilizando AAL1)	6,0 ms	

3.2.1.2 RTT

El retardo ida y vuelta (RTT) es el tiempo que transcurre entre la ocurrencia de dos eventos de referencia de paquetes IP correspondientes, es decir cuando un paquete i viaja del transmisor al receptor y luego de vuelta al propio transmisor.

$$RTT = (t_2 - t_1) + (t_2' - t_1') \quad \text{Ecuación 3-2}$$

Como se observa en 3.2 el RTT se puede representar como una suma de OWD, siendo $OWD=t_2-t_1$ el retardo en el sentido de ida y $OWD'=t_2'-t_1'$ el retardo en el sentido de vuelta con lo cual el RTT quedaría expresado en términos del OWD.

$$RTT = (t_2 - t_1) + (t_2' - t_1') = OWD + OWD' \quad \text{Ecuación 3-3}$$

En la figura 3.3 se puede observar de forma gráfica el RTT.

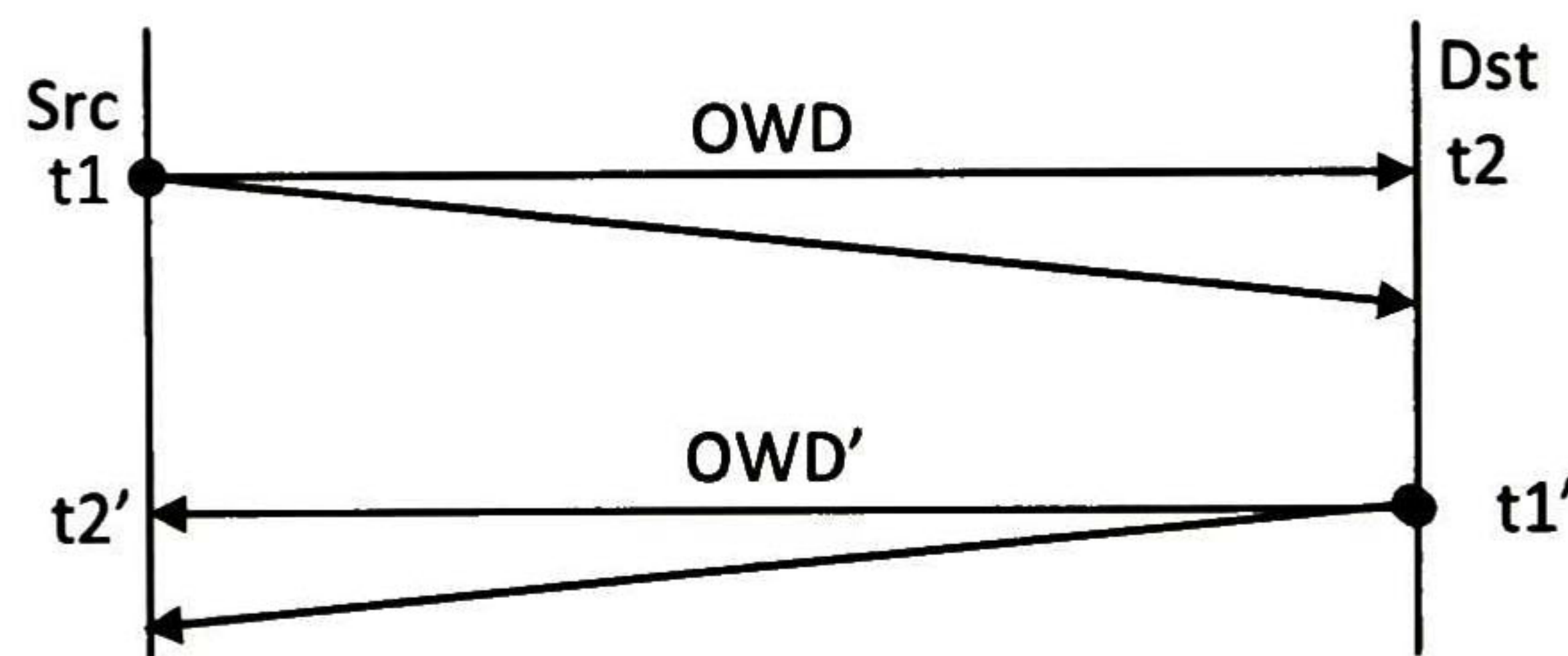


Figura 3-3 RTT

3.2.2 JITTER

La variación del retardo de paquetes IP entre dos puntos extremo a extremo, o Jitter, se define en base a las observaciones de los arribos de paquetes IP correspondientes a los MP de ingreso $a_{1,k}$ y egreso $a_{2,k}$. El Jitter J_k para el paquete IP k entre la fuente y el destino es la diferencia entre el OWD absoluto x_k del paquete y un OWD de referencia definido, $d_{1,2}$, entre esos MP.

$$x_k = a_{2,k} - a_{1,k} \quad \text{Ecuación 3-4}$$

$$J(k) = x_k - d_{1,2} \quad \text{Ecuación 3-5}$$

Donde OWD de referencia $d_{1,2}$ puede ser:

- El OWD absoluto del primer paquete entre esos dos MP,
- el IP OWD TD promedio o
- el OWD con el valor mínimo de una población de paquetes.

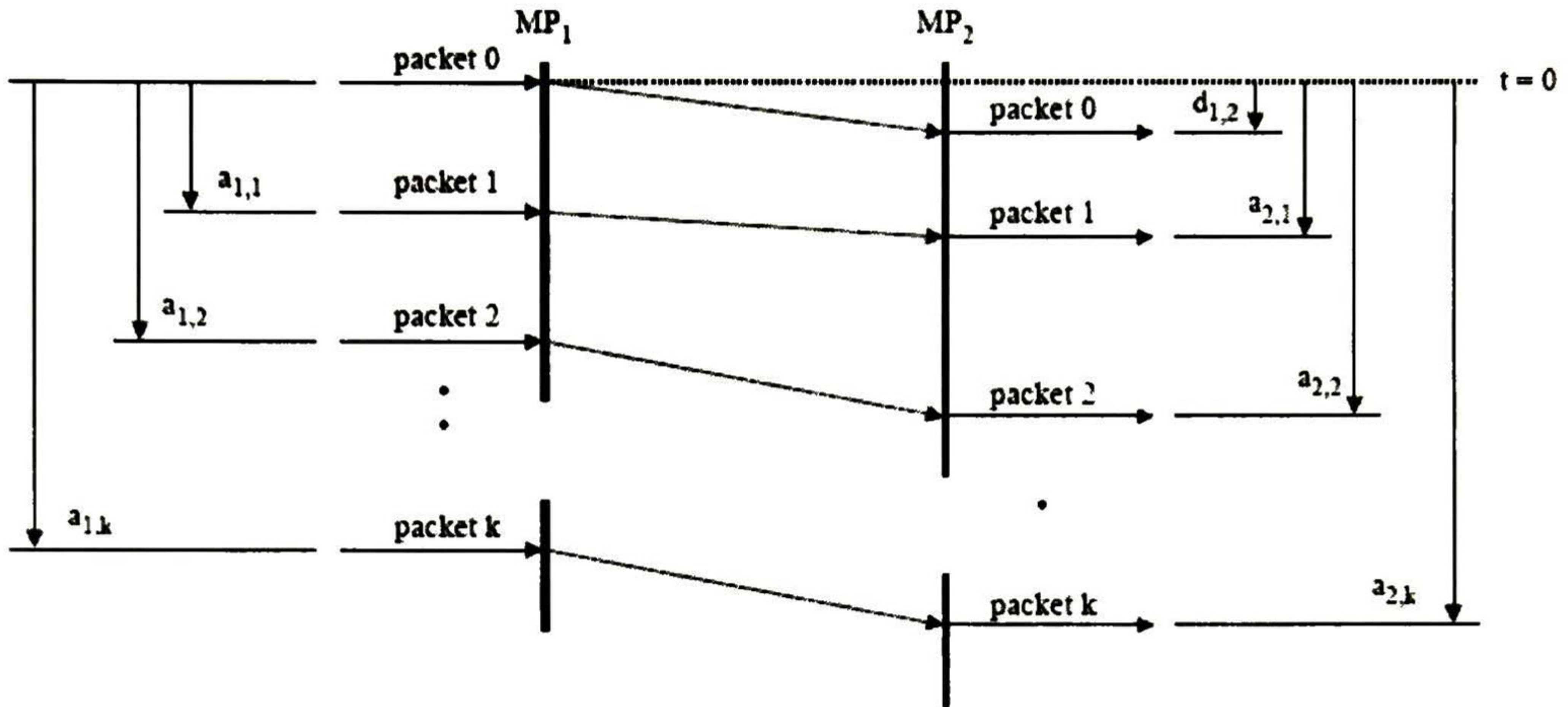


Figura 3-4 Jitter

El Jitter también puede ser expresado en términos del OWD y RTT como se muestra en las siguientes ecuaciones:

$$J(k)_{OWD} = OWD_{k+1} - OWD_k \quad \text{Ecuación 3-6}$$

$$J(k)_{RTT} = RTT_{k+1} - RTT_k \quad \text{Ecuación 3-7}$$

Otra definición del Jitter se establece en el RFC 1889 [11]. En donde es definido como la diferencia en el "relative transit time" para dos paquetes; el "relative transit time" es la diferencia entre la estampa de tiempo RTP del paquete y el reloj del receptor en el tiempo de arribo, medida en las mismas unidades.

La diferencia en el "relative transit time" entre dos paquetes adyacentes se calcula siguiendo la fórmula 3.8.

$$D(i, j) = (R_j - S_j) - (R_i - S_i) \quad \text{Ecuación 3-8}$$

Donde:

$D(i, j)$: es el "relative transit time" entre dos paquetes adyacentes i y j .

R_j : es la hora en la que se recibió el paquete j .

S_j : es la hora en que se envió el paquete j (determinada por la fecha y hora RTP).

R_i : es la hora en la que se recibió el paquete i .

S_i : es la hora en que se envió el paquete i (determinada por la fecha y hora RTP).

Cada paquete de medios RTP entrante provoca un nuevo cálculo de la diferencia del "relative transit time" entre el actual paquete i recibido y el anterior paquete $i-1$ recibido. En lugar de escribir esta diferencia $D(i-1, i)$, consideramos la sintaxis más simple D_i .

De acuerdo con el RFC 1889 [11], el "Jitter" J es calculado de acuerdo a la ecuación 3-9.

$$J = J' + \frac{(|D_i| - J')}{16} \quad \text{Ecuación 3-9}$$

Donde:

J : es el valor calculado actual.

J' : es el valor calculado anterior.

Ethereal utiliza esta última definición para el cálculo del jitter, durante la captura de paquetes, por tal motivo en nuestro estudio optaremos por utilizar esta definición.

3.2.3 PERDIDA DE PAQUETES

La pérdida de paquetes es importante en numerosas aplicaciones, principalmente en aquellas de tiempo real. La pérdida de paquetes es causada por la congestión de la red, donde los buffer de los routers sufren de desborde y se ven forzados a descartar paquetes. En un sistema de tiempo real como VoIP no son deseables las pérdidas elevadas de paquetes, para la calidad del servicio y funcionamiento de este último.

La pérdida de paquetes en una red puede ocurrir:

- Si el paquete tiene un retardo mayor que el del timer de retransmisión.
- Si el paquete es corrompido y se recibe un NAK.
- Si se recibe un paquete ICMP TTL Exceeded.
- Si el paquete no obtiene recursos en el hop dado y es descartado.

Usualmente en la pérdida de paquetes hablamos de que la probabilidad de pérdida, P_k , para un paquete k , no debe exceder un límite, es decir:

$$P(k = \text{perdido}) \leq W_{\min} \quad \text{Ecuación 3-10}$$

Una ventaja del tráfico multimedia es su tolerancia a pérdidas; tasas < 2% suelen pasar inadvertidas; típicamente los requerimientos para la pérdida de paquetes por la codificación de voz son < 1% [20]. Estas pérdidas podrían eliminarse con TCP, pero la estrategia de retransmisión implica retardos que son habitualmente inaceptables; por ello, el protocolo usado es UDP.

Se puede mejorar este parámetro aumentando la prioridad de los paquetes de voz respecto a los de otras aplicaciones. Para atenuar los efectos en caso de pérdidas elevadas se usan distintas técnicas no excluyentes que permiten elevar el umbral permisible de pérdidas incluso hasta el 20%, dependiendo de la codificación:

- **FEC (Forward Error Correction):** Técnica utilizada en sistemas de transmisión digital de datos para proveer robustez contra canales ruidosos, al proporcionar un servicio de corrección de errores. El codificador de FEC, agrega información redundante a los bits de información que serán transmitidos. En la Fig. 4.5 se presenta una trama FEC. El Decodificador de FEC utiliza esta información redundante, para detectar y corregir errores [21]. Una desventaja de esta técnica de corrección de errores es el aumento de la utilización de ancho de banda debido a la redundancia añadida a los paquetes.

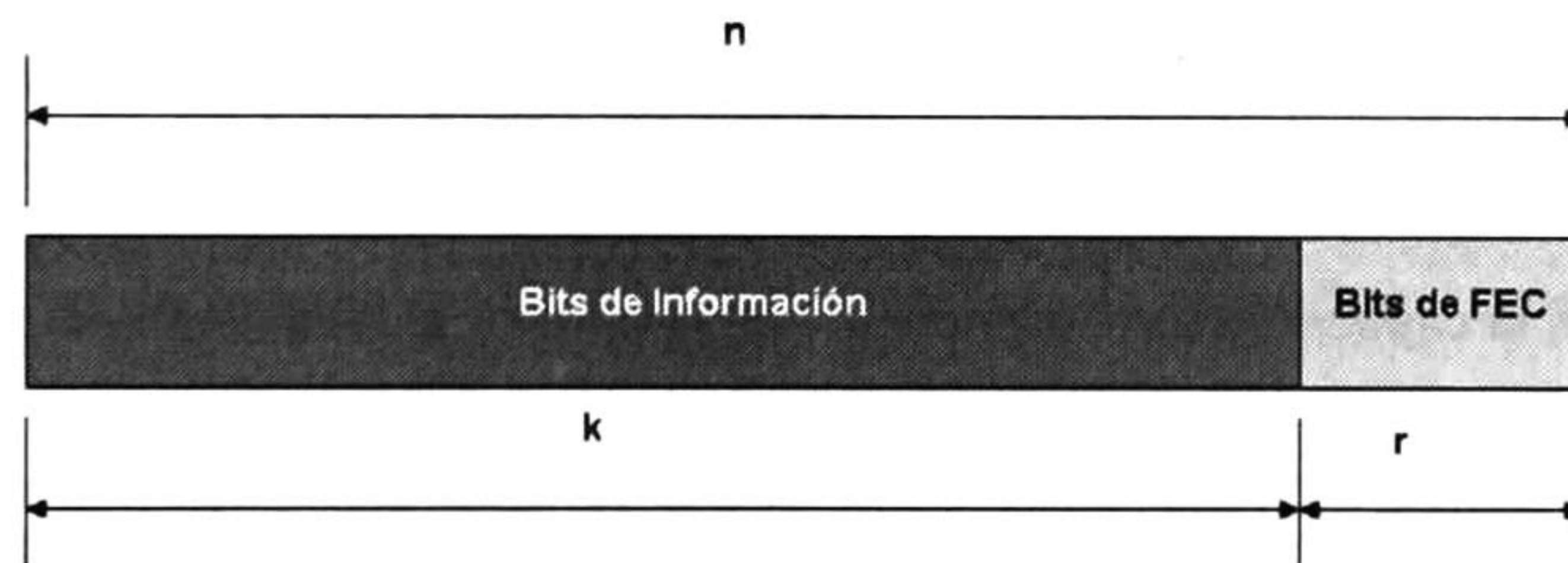


Figura 3-5 Trama FEC

- **Interleaving para atenuar los efectos de pérdidas de ráfagas:** El procedimiento implica el reacomodo (orden temporal) de las tramas originales, de acuerdo a una distancia específica dada por el algoritmo "interleaving" para asegurar que las tramas previamente consecutivas están separadas en la transmisión y cambiadas de vuelta en su secuencia original en el receptor. Dentro de la técnica de "interleaving", un solo paquete perdido dará lugar solamente a múltiples boquetes cortos en diversas corrientes de los datos recibidos, que el receptor puede tolerar [22]. Teniendo como desventaja el aumento en el retardo extremo a extremo, debido al tiempo de procesamiento de esta técnica. La Fig. 4.6 ilustra esta técnica.

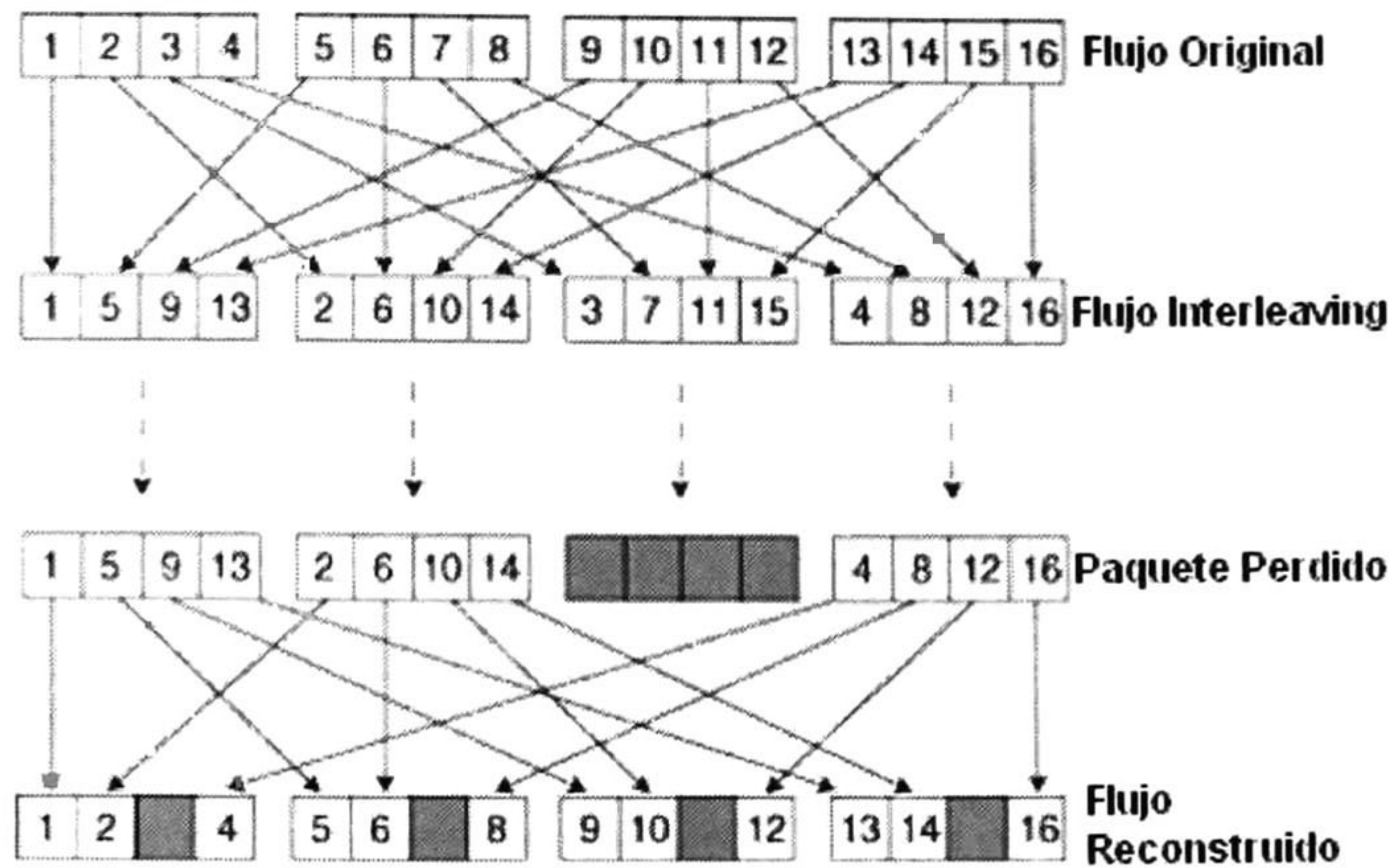


Figura 3-6 Técnica de Interleaving.

3.2.4 ANCHO DE BANDA

El ancho de banda es una medida de la cantidad de información por unidad de tiempo que se puede transportar entre dos puntos A y B [16].

Cuatro métricas importantes para el ancho de banda son [23]:

- **Capacidad:** Es el máximo ancho de banda que un enlace o camino de red puede transportar. Cuando tenemos un camino, dado por la secuencia de saltos $\lambda_1, \lambda_2, \dots, \lambda_n$ con correspondientes capacidades C_1, C_2, \dots, C_n , entonces su capacidad está dada por:

$$C = \min(C_i)_{i=1}^n \quad \text{Ecuación 3-11}$$

Es decir, la capacidad del camino, está limitada o es igual a la del enlace con la mínima capacidad.

- **Ancho de banda disponible:** Es el máximo ancho de banda sin usar en un enlace o trayectoria de red. A diferencia de la capacidad, el ancho de banda disponible es una métrica dinámica en el tiempo, por lo cual su estimación es más compleja. Analíticamente el ancho de banda disponible se obtiene de la siguiente manera; supongamos que C_i representa la capacidad del enlace i , ahora definamos como u_i el coeficiente de utilización del enlace i en el tiempo t , de tal manera que $0 \leq u_i \leq 1$, entonces el ancho de banda disponible para el enlace i , A_i , está dado por:

$$A_i(t) \equiv C_i(1 - u_i(t)) \quad \text{Ecuación 3-12}$$

Debido a que u_i es un parámetro dependiente del tiempo, entonces la ecuación 3.11 en el intervalo $t + \tau$ puede expresarse como:

$$A_i(t + \tau) \equiv C_i(1 - u_i(t + \tau)) \quad \text{Ecuación 3-13}$$

Para un camino de n saltos lo anterior viene dado por la siguiente ecuación:

$$A(t + \tau) \equiv \min\{A_i(t + \tau)\}_{i=1}^n \quad \text{Ecuación 3-14}$$

Como en el caso de la capacidad el ancho de banda disponible para un camino es igual al enlace con el mínimo ancho de banda disponible.

- *Throughput*: Es el "bit rate" medido en la capa de red (IP) en cierto periodo de tiempo.
- *BTC (Bulk Transfer Capacity)*: Es el máximo "bit rate" obtenido en un enlace, visto desde la capa de transporte (conexión TCP\UDP).

Actualmente la voz que recibe un "gateway" es digitalizada y comprimida según distintos algoritmos, que se caracterizan por conseguir mayores proporciones de compresión, aunado con la implementación de codificadores a tasas variables que reciben información sobre el estado de congestión de la red y del porcentaje de pérdidas en una transmisión para adecuar el régimen binario de compresión de la voz a cada situación. La Tabla 3.3 resume las descripciones de los codecs especificados por la recomendación H.323, incluyendo ancho de banda del audio, tiempo de muestreo, y ancho de banda IP total [20].

Hay que considerar así mismo el parámetro denominado supresión de silencio; mediante la implementación de un detector de actividad de voz (VAD) que consigue suprimir en la información que se transmite, los silencios de una conversación. Los intervalos de silencio constituyen más del 50% del tiempo total por lo que su supresión constituye un gran ahorro de ancho de banda.

Con este parámetro activado, se consigue que la transmisión de paquetes se reduzca, liberando ancho de banda.

CODEC	Algoritmo de Codificación	Ancho de Banda del Audio	Tiempo de Muestreo	Ancho de Banda IP Total
G.711	PCM	64kbps	0.125ms	80kbps
G.723.1	ACELP	5.6kbps 6.4kbps	30ms	16.27kbps 17.07kbps
G.726	ADPCM	32kbps	0.125ms	48kbps
G.728	LD-CELP	16kbps	0.625ms	32kbps
G.729 (A)	CS-ACELP	8kbps	10ms	24kbps

Tabla 3-3 Especificaciones de CODECs de Audio [20].

3.3 CALIDAD DE VOZ

Aplicaciones de voz y video son particularmente sensibles a la calidad de los servicios de red. En las aplicaciones VoIP una señal de voz es primero empaquetada y después transmitida sobre la red IP. Sin embargo, en el extremo receptor, los paquetes pueden llegar distorsionados o se pierden debido al retardo, jitter y errores presentes en la red. Esta pérdida y retraso de los paquetes degrada la calidad de la voz en el receptor.

Para poder estimar la calidad del flujo de los paquetes de voz, es fundamental producir, de manera general, medidas cuantitativas que reflejen una valoración objetiva de la calidad de este servicio. El MOS, es una prueba subjetiva aceptada como estándar de verificación de la calidad de voz. Sin embargo, la calificación subjetiva MOS consume mucho tiempo y no es precisa.

El modelo-E, es una herramienta para predecir como un "usuario promedio" calificaría la calidad de la voz de una llamada telefónica, proporciona una fórmula para calcular la calidad mediante parámetros conocidos.

3.3.1 MOS

La métrica de percepción de voz más utilizada es el MOS, cuyo valor se deriva de:

- Asignar valores numéricos a un conjunto de descriptores subjetivos.
- Calcular la media de los valores numéricos correspondientes.

Los valores particulares asignados a cada categoría pueden ser tomados de cualquier escala. No obstante, se utiliza como referencia la recomendación P.800 [25] de la ITU, la cual indica el uso de una escala de cinco valores como se muestra en la tabla 3-3. Cualquiera que fuese la escala utilizada, la suposición que se realiza al describir MOS es que el valor que representa es el resultado de:

- Que un gran conjunto de usuarios es cuestionado acerca de su opinión subjetiva de la calidad de un número de conexiones de voz dentro de la escala especificada.

- Los valores numéricos en los reportes generados por los usuarios es promediado.

Calidad de voz	Calificación
Excelente	5
Buena	4
Regular	3
Mediocre	2
Mala	1

Tabla 3-4 Escala MOS para la calidad de voz.

Debido a la definición formal y al extenso uso de la prueba MOS, existe una tendencia a asumir que la calificación media de opinión mantiene un valor constante. Sin embargo, en el caso de que el valor MOS sea de 3.75, no se puede realizar una interpretación significativa sin tener al menos la descripción del contexto en el que se calculó el valor.

3.3.2 MODELO E

El modelo E es un modelo computacional, estandarizado por la ITU en la recomendación G.107 [24], que utiliza parámetros de transmisión para predecir de manera subjetiva la calidad de la comunicación. El modelo combina diferentes limitantes basándose en el principio que el efecto percibido a causa de éstas es aditivo cuando es trasladado a la escala psico-acústica R , cuyo valor oscila entre 0 (i.e. calidad pobre) y 100 (i.e. calidad excelente).

$$R = R0 - Is - Id - Ie + A \quad \text{Ecuación 3-15}$$

Donde:

- $R0$ es una función de los niveles de ruido en el circuito y en las habitaciones en ambos extremos de la comunicación.
- Is es una función del error de cuantificación ocasionado por codecs con una tasa de codificación baja.
- Id es principalmente una función del nivel y retardo del eco con respecto a la señal original y los retardos boca a oído en ambas direcciones (i.e. OWD).

- I_e cubre las eventualidades causadas por lo que G.107 [24] define como el uso de equipo especial, es una función del tipo de codificador utilizado (i.e. CodecTyp) y de la fracción de paquetes perdidos (i.e. PLR).
- A es el factor de expectación, el cual captura el hecho que los usuarios pueden aceptar algo de degradación en calidad en cambio de una comunicación más rápida.

Además G.107 estipula que cuando la comunicación es establecida por un medio fijo, i.e. conexión por red Ethernet, se pueden utilizar valores por defecto para cada parámetro de la escala R . Lo cual permite reducir (3.15) a:

$$R \sim 93.4 - I_d(\text{OWD}_{\text{ms}}) - I_e(\text{CodecTyp}, \text{PLR}) \quad \text{Ecuación 3-16}$$

Donde

$$I_d \sim 0.024 * \text{OWD}_{\text{ms}} + 0.11(\text{OWD}_{\text{ms}} - 177.3) * H(\text{OWD}_{\text{ms}} - 177.3) \quad \text{Ecuación 3-17}$$

$$H(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} \quad \text{Ecuación 3-18}$$

$$I_e \sim \text{CodecTyp} + 40 * \ln(1 + 0.1 * \text{PLR}) \quad \text{Ecuación 3-19}$$

$$\text{CodecTyp} = \begin{cases} 0, & \text{G.711} \\ 10, & \text{G.729} \end{cases} \quad \text{Ecuación 3-20}$$

Basándose en el valor de R , se puede predecir las reacciones subjetivas de los usuarios, como las obtenidas en una prueba MOS.

$$\text{MOS} = \begin{cases} 1 & R < 0 \\ 1 + 0.035R + R(R - 60)(100 - R) * 7 * 10^{-6} & 0 < R < 100 \\ 4.5 & R > 100 \end{cases} \quad \text{Ecuación 3-21}$$

En términos prácticos, cuando una comunicación muestra un valor de $R < 60$, se considera que proporciona al usuario una calidad de servicio pobre. La tabla 3.4 muestra la relación entre el valor de la escala R y el MOS, así como las distintas categorías de calidad de servicio.

R	MOS	Grado de satisfacción del usuario
90-100	4.34-4.50	Muy Satisfecho
80-89	4.02-4.31	Satisfecho
70-79	3.60-3.99	Algunos usuarios insatisfechos
60-69	3.10-3.55	Muchos usuarios insatisfechos
50-59	2.58-3.05	La mayoría de usuarios insatisfechos
1-49	1-2.52	No recomendado

Tabla 3-5 Relación entre R y MOS.

Este capítulo hace mención de los factores que determinan la calidad de servicio (métricas de desempeño) en los sistemas voz sobre IP. El aspecto esencial de la evaluación global de un servicio es la opinión de los usuarios. El resultado de esta evaluación expresa los grados de satisfacción de los usuarios.

La calidad de servicio es el efecto global de la calidad de funcionamiento de un servicio que determina el grado de satisfacción de un usuario.

En el tráfico de voz una buena QoS se traduce en una alta calidad de voz, la cual está determinada por muchos parámetros que se negocian.

4 ESPECIFICACIÓN DE REQUERIMIENTOS

En los capítulos anteriores se explicó que es un sistema de VoIP, así como su arquitectura y sus elementos que la integran, al igual que su importancia debido a su bajo costo y la gran infraestructura del internet.

Esta comunicación es afectada por diversos factores que son inherentes en el tráfico a través de una red de conmutación de paquetes, la cual no nos garantiza una cierta calidad de servicio, solo ofrece su mejor esfuerzo (best effort) como es el caso del internet.

Es por esto que se ha desarrollado un sistema de comunicación de VoIP el cual, además de poder establecer una llamada, podrá estimar y calcular los valores que afectan dicha calidad de servicio para su posterior estudio, procesamiento y análisis.

Para entender el funcionamiento y la arquitectura de este sistema se describen a continuación su:

- Especificación de requerimientos del Sistema VoIP.
 - Requerimientos Funcionales.
 - Requerimientos No Funcionales.
- Casos de uso
- Arquitectura del Sistema VoIP
- Doxygen y la Documentación del Sistema.

4.1 ESPECIFICACIÓN DE REQUERIMIENTOS DEL SISTEMA VOIP

La especificación de requerimientos es una herramienta muy importante en el desarrollo de software, esta nos dice las condiciones, capacidades y limitantes del sistema nos explica toda la funcionalidad de este, es decir, nos da una información detallada de sus funciones. Estos requerimientos se dividen en dos grupos:

- **Funcionales:** Expresan la naturaleza del funcionamiento del sistema (como interacciona el sistema con su entorno y cuáles van a ser su estado y funcionamiento)
- **No funcionales:** Restricciones en el espacio de posibles soluciones:
 - Rendimiento del sistema: fiabilidad, tiempo de respuesta, disponibilidad...
 - Interfaces: dispositivos de E/S, usabilidad, interoperabilidad...
 - Proceso de desarrollo: estándares, herramientas, plazo de entrega...

4.1.1 REQUERIMIENTOS FUNCIONALES

ID	NOMBRE	PRORIDAD
RF-VoIPAS-01	Realización de Llamadas	Alta
DESCRIPCIÓN		
El sistema permitirá que dos usuarios los cuales ya estén registrados en el servidor, puedan establecer una llamada de VoIP.		
CRITERIO DE EXITO		
Los usuarios establecerán y realizaran una comunicación si ningún problema.		

ID	NOMBRE	PRORIDAD
RF-VoIPAS-02	Codec's	Alta
DESCRIPCIÓN		
<p>El sistema permitirá al usuario elegir el códec con el cual quiere que se realice la llamada.</p> <ul style="list-style-type: none"> • PCM • G711 A Law • G711 Mu Law 		
CRITERIO DE EXITO		
<p>Los usuarios establecerán y realizaran una comunicación con el códec elegido si ningún problema.</p>		

ID	NOMBRE	PRORIDAD
RF-VoIPAS-03	Modo de Captura	Media
DESCRIPCIÓN		
<p>El sistema permitirá al usuario elegir el modo de captura que desee y pueden ser por:</p> <ul style="list-style-type: none"> • Manual: Se realizara una llamada y cálculo de parámetros hasta que el usuario lo desee. • Muestras: El sistema realizara una captura de N muestras, siendo N un número entero indicado por el usuario. • Tiempo: El sistema realizara una captura de M minutos, siendo N un número entero indicado por el usuario. 		
CRITERIO DE EXITO		
<p>Los usuarios establecerán y realizaran una comunicación con el códec elegido si ningún problema.</p>		

ID	NOMBRE	PRORIDAD
RF-VoIPAS-04	Parámetros Medidos	Alta
DESCRIPCIÓN		
<p>El sistema durante la llamada de VoIP realizara el cálculo de las siguientes métricas:</p> <ol style="list-style-type: none"> 1. Paquetes Transmitidos. 2. Paquetes Recibidos. 3. Paquetes Perdidos. 4. Porcentaje de Perdida 5. Secuencias Recibidas. 6. OWD. 7. Jitter de OWD. 8. Absoluto del Jitter de OWD. 9. Promedio del Absoluto del Jitter de OWD. 10. InterArrival. 11. Jitter de InterArrival. 12. Absoluto del Jitter de InterArrival. 13. Promedio del Absoluto del Jitter de InterArrival. 14. RTT 15. Jitter de RTT 16. Absoluto del Jitter de RTT 17. Promedio del Absoluto del Jitter de RTT 		
CRITERIO DE EXITO		
El sistema realiza correctamente los cálculos de estos parámetros.		

ID	NOMBRE	PRORIDAD
RF-VoIPAS-05	Estimación de Hurst	Alta
DESCRIPCIÓN		
<p>El sistema además de la medida de los parámetros antes mencionados el sistema también estimara su respectivo parámetro de autosimilitud H (Parámetro de Hurst) de las trazas de los parámetros mencionados en el requerimiento RF-VoIPAS-06.</p> <p>Los métodos que se implementaran para la estimación del parámetro de Hurst serán:</p> <ul style="list-style-type: none"> • R/S Statistic • Absolute Moment • Variance Method • Variance of Residuals • Modified Allan Variance • Periodogram • LWhittle • Abry-Veitch Method 		
CRITERIO DE EXITO		
El sistema realiza correctamente los cálculos de estos parámetros.		

ID	NOMBRE	PRORIDAD
RF-VoIPAS-06	Calculo del MOS	Alta
DESCRIPCIÓN		
<ul style="list-style-type: none"> • El sistema podrá calcular el valor del MOS, lo cual ayudara a determinar la calidad de servicio desarrollada durante la llamada de VoIP. Las variantes calculadas del MOS serán: <ul style="list-style-type: none"> ○ MOS OWD. ○ MOS RTT. 		
CRITERIO DE EXITO		
El sistema realiza correctamente los cálculos de estos parámetros.		

ID	NOMBRE	PRORIDAD
RF-VoIPAS-07	Guardado Manual de Trazas de Métricas Capturadas	Media
DESCRIPCIÓN		
El sistema le dará la opción al usuario de poder guardar las trazas de las métricas capturadas en archivos independientes, cuando el usuario elija la moda de captura manual.		
CRITERIO DE EXITO		
El correcto guardado de las trazas de las métricas elegidas.		

ID	NOMBRE	PRORIDAD
RF-VoIPAS-08	Guardado Automático de Trazas de Métricas Capturadas	Media
DESCRIPCIÓN		
El sistema le guardara las trazas capturadas de forma automática, cuando se elija los modos de captura por paquetes o por minutos.		
CRITERIO DE EXITO		
El correcto guardado de las trazas de las métricas elegidas.		

ID	NOMBRE	PRORIDAD
RF-VoIPAS-09	Proposición de Nombre de Guardado	Baja
DESCRIPCIÓN		
Cuando el usuario guarde las trazas de las métricas el sistema le pospondrá un nombre para el archivo a guardar tomado en cuenta los nombres de usuario, fecha y tipo de parámetro medido.		
CRITERIO DE EXITO		
La propuesta del nombre de guardado sea guardado.		

ID	NOMBRE	PRORIDAD
RF-VoIPAS-10	Grafica de Métricas	Media
DESCRIPCIÓN		
El sistema tendrá la opción de poder graficarnos las trazas de las métricas calculadas.		
CRITERIO DE EXITO		
Las graficas se muestren sin ningún error.		

ID	NOMBRE	PRORIDAD
RF-VoIPAS-11	Impresión de Graficas	Baja
DESCRIPCIÓN		
Las gráficas generadas se podrán imprimir.		
CRITERIO DE EXITO		
La correcta impresión de las gráficas.		

ID	NOMBRE	PRORIDAD
RF-VoIPAS-12	Guardado de Gráficas	Baja
DESCRIPCIÓN		
Las graficas generadas por el sistema tendrán la opción de ser guardadas.		
CRITERIO DE EXITO		
El correcto guardado de las graficas.		

ID	NOMBRE	PRORIDAD
RF-VoIPAS-13	Frecuencia de Actualización	Baja
DESCRIPCIÓN		
El usuario podrá elegir la frecuencia con la cual quiere que se actualicen los datos del estado de la llamada: paquetes transmitidos, recibidos, perdidos, porcentaje de pérdidas, H y MOS.		
CRITERIO DE EXITO		
El correcto guardado de las graficas.		

ID	NOMBRE	PRORIDAD
RF-VoIPAS-14	Contestación Automática	Baja
DESCRIPCIÓN		
La aplicación contara con la opción de contestador automático, esto significa que si esta opción está habilitada, el sistema cuando reciba un petición de comunicación esta responderá de manera automática sin necesidad de intervención de algún usuario.		
CRITERIO DE EXITO		
Cuando queramos establecer comunicación con otra instancia que tenga esta opción habilitada esta responderá de manera automática sin necesidad de intervención de algún usuario.		

4.1.2 REQUERIMIENTOS NO FUNCIONALES

En la siguiente tabla se enumeran los requerimientos no funcionales.

<u>ID</u>	<u>Descripción</u>
RNFunc-01	El sistema deberá funcionar en la plataforma del sistema operativo de Microsoft Windows XP.
RNFunc-02	El programa se desarrollara con el lenguaje de C++.
RNFunc-03	El Compilador que se utilizara será el Borland Builder C++ 6.0.
RNFunc-04	El sistema deberá presentarte en el idioma ingles.
RNFunc-05	El sistema deberá manejar una convención de nombres consistente para identificar los resultados de las operaciones realizadas y utilizarlos como nombres de archivo sugeridos.
RNFunc-06	El sistema deberá almacenar las configuraciones que se realicen.
RNFunc-07	El sistema contará con un manual de mantenimiento técnico, proporcionado a través de la documentación del código fuente que se realizará con una herramienta adicional (Doxygen).
RNFunc-08	La interfaz de usuario del sistema deberá proveer teclas y botones de acceso rápido a las operaciones más importantes del sistema.

Tabla 4-1 Requerimientos No Funcionales

5 DISEÑO DEL SISTEMA

Una vez realizado el análisis de requerimiento, procederemos a realizar un diseño que cubra cada uno de los requerimientos especificados en el capítulo anterior. Para ello trataremos los siguientes puntos:

1. Casos de uso.
2. Arquitectura del sistema de VoIP.
3. Diseño orientado a objetos.
4. Cobertura de requerimientos.
5. Interfaz del sistema.
6. Doxygen y la documentación del sistema.

5.1 CASOS DE USO

Los casos de uso son técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software. Cada caso de uso proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico.

La tabla 5-1 nos muestra los actores participantes en los casos de uso de nuestro sistema.

Actores	Descripción	Acciones
Usuario	Es la persona que realiza y/o contesta la llamada.	<ul style="list-style-type: none">• Llamar• Responder• Cancelar Llamada• Terminar Llamada
Aplicación Cliente	Es la aplicación que permite realizar las llamadas y mediciones.	<ul style="list-style-type: none">• Registrarse al servidor.• Hacer .llamadas• Cancelar llamadas.• Hacer mediciones de parámetros• Guardar Archivos.• Graficas mediciones.
Aplicación Servidor	Aplicación que se encarga de registrar los usuarios que se encuentran conectados y sus respectivas direcciones IP.	<ul style="list-style-type: none">• Registrar Clientes

Tabla 5-1 Actores de VoIPAS

La Figura 5-1 nos muestra el diagrama de Casos de Uso de la aplicación.

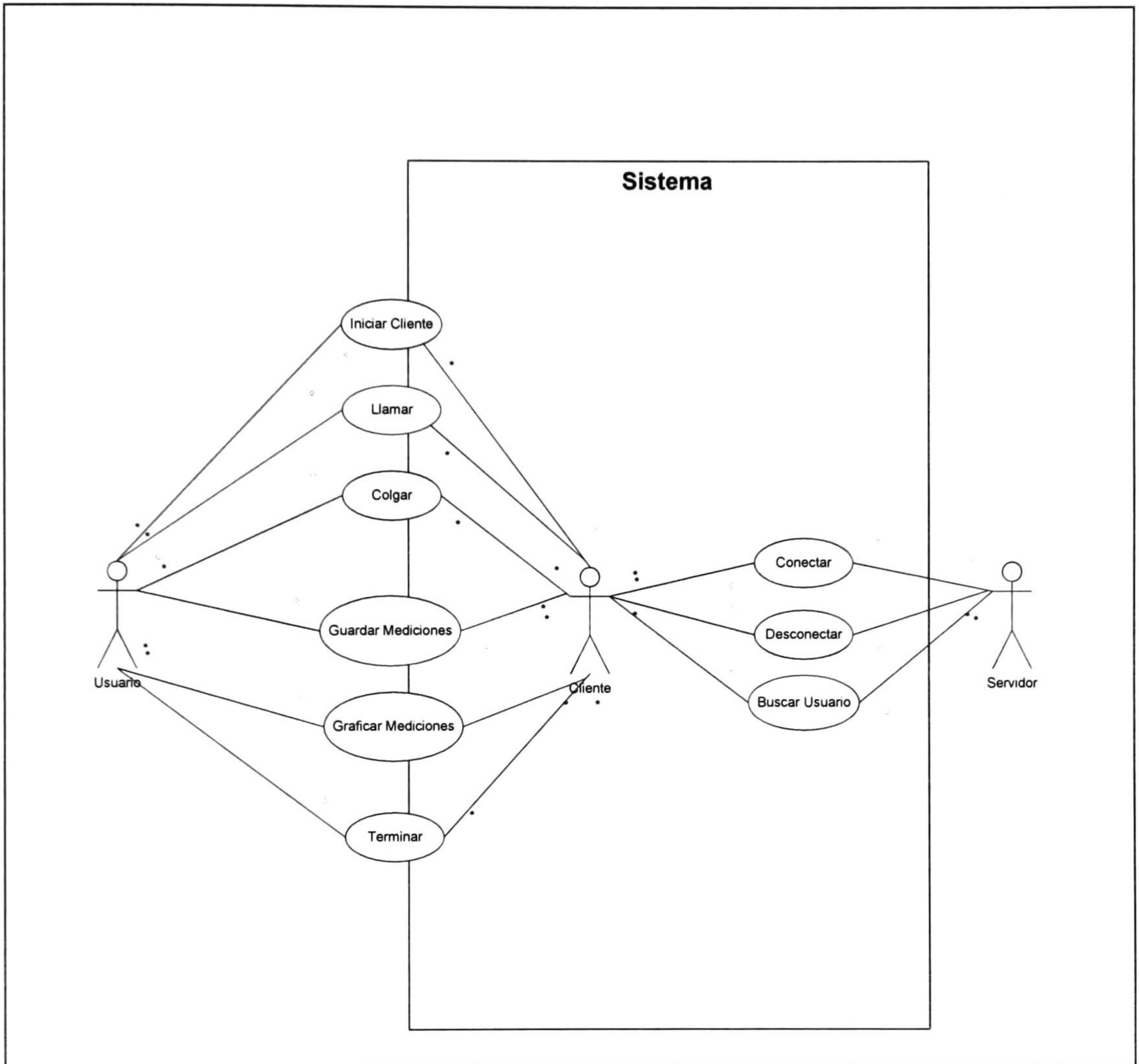


Figura 5-1 Diagrama de Casos de Uso

A continuación se describen los casos de usos que se desarrollan en nuestro sistema y se ilustraron en la figura 5-1:

Código:	CS-VoIPAS-01.
Nombre:	Iniciar
Autor:	Jesús Antonio Argáez Xool
Fecha:	09/09/2008
Descripción:	Iniciar la Aplicación cliente y cargar la configuración guardada o predeterminada.
Actores:	<ol style="list-style-type: none"> 1. Aplicación Cliente. 2. Usuario.
Precondiciones:	<ol style="list-style-type: none"> 1. Se deberá contar con todos los archivos necesarios para el correcto funcionamiento de la aplicación cliente.
Flujo Normal:	<ol style="list-style-type: none"> 1. El usuario ejecuta la aplicación cliente. 2. La aplicación cliente carga la configuración guardada. 3. El cliente se conecta al servidor.
Flujo Alternativo:	<ol style="list-style-type: none"> 2. La aplicación cliente si no carga una configuración guardada, carga una configuración predeterminada. 3. El cliente no se puede conectar al servidor.
Pos condiciones:	

Código:	CS-VoIPAS-02.
Nombre:	Conectar
Autor:	Jesús Antonio Argáez Xool
Fecha:	09/09/2008
Descripción:	
Permite registrar la aplicación cliente al servidor guardando su nombre de usuario e IP.	
Actores:	
<ol style="list-style-type: none"> 1. Aplicación Cliente. 2. Aplicación Servidor. 3. Usuario(Origen /Destino) 	
Precondiciones:	
<ol style="list-style-type: none"> 1. El servidor ya se debe encontrar en ejecución. 2. La aplicación cliente debe tener registrado la IP del servidor en su archivo ini. 	
Flujo Normal:	
<ol style="list-style-type: none"> 1. El usuario inicia la aplicación cliente. 2. El cliente envía una solicitud de conexión al servidor. 3. El servidor le responde su aceptación. 4. El cliente le informa al usuario de su conexión. 	
Flujo Alternativo:	
<ol style="list-style-type: none"> 3. El servidor le informa de la negación de conexión por existencia de usuario. 4. El cliente le informa al usuario de su negación de conexión por existencia de usuario. 5. Si el servidor no responde en s segundos la aplicación cliente le informa al usuario que el servidor no responde. 	
Pos condiciones:	

Código:	CS-VoIPAS-03.
Nombre:	Desconectar
Autor:	Jesús Antonio Argáez Xool
Fecha:	09/09/2008
Descripción:	<p>Permite eliminar la dirección IP y nombre de usuario de la aplicación cliente del servidor.</p>
Actores:	<ol style="list-style-type: none"> 4. Aplicación Cliente. 5. Aplicación Servidor.
Precondiciones:	<ol style="list-style-type: none"> 1. El servidor ya se debe encontrar en ejecución. 2. El servidor debe de tener registrado al cliente. 3. La aplicación cliente debe tener registrado la IP del servidor en su archivo ini.
Flujo Normal:	<ol style="list-style-type: none"> 1. El cliente envía una solicitud de desconexión al servidor. 2. El servidor busca si el nombre de usuario se encuentra registrado. 3. Si esta registrado el servidor lo borra de la tabla de direcciones.
Flujo Alternativo:	<ol style="list-style-type: none"> 3. Si el nombre del usuario no está registrado la solicitud simplemente es ignorada.
Pos condiciones:	

Código:	CS-VoIPAS-04.
Nombre:	Buscar Usuario
Autor:	Jesús Antonio Argáez Xool
Fecha:	09/09/2008
Descripción:	
<p>Quando un usuario quiere establecer una comunicación con otro usuario solo necesitara el nombre del otro usuario, ya que la aplicación cliente del usuario se encargará de preguntarle al servidor la dirección IP del usuario, el servidor será encargado de buscar la dirección del usuario en la tabla de direcciones y la devolverá si esta existe.</p>	
Actores:	
<ol style="list-style-type: none"> 1. Aplicación Cliente. 2. Aplicación Servidor. 	
Precondiciones:	
<ol style="list-style-type: none"> 1. El servidor ya se debe encontrar en ejecución. 2. El servidor debe de tener registrado al cliente destino con el que se desea obtener comunicación. 	
Flujo Normal:	
<ol style="list-style-type: none"> 1. El cliente le envía una solicitud de búsqueda de usuario al servidor y el nombre del usuario con el que desea establecer una llamada. 2. El servidor busca la dirección IP del usuario. 3. Si la encuentra le envía una respuesta indicándole que lo hallo y su dirección IP. 	
Flujo Alternativo:	
<ol style="list-style-type: none"> 3. Si no encuentra al usuario le envía una respuesta al cliente del primer usuario indicándole que el usuario no existe. 	
Pos condiciones:	

Código:	CS-VoIPAS-05.
Nombre:	Llamada
Autor:	Jesús Antonio Argáez Xool
Fecha:	09/09/2008
Descripción:	
Permite la comunicación entre dos usuarios.	
Actores:	
<ol style="list-style-type: none"> 1. Usuario. 2. Aplicación Cliente. 3. Aplicación Servidor. 	
Precondiciones:	
<ol style="list-style-type: none"> 1. El servidor ya se debe encontrar en ejecución. 2. El servidor debe de tener registrado al cliente destino con el que se desea obtener comunicación. 3. El cliente origen debe tener registrada la dirección IP del servidor. 	
Flujo Normal:	
<ol style="list-style-type: none"> 1. El usuario proporciona el nombre del usuario con el que desea establecer comunicación. 2. El cliente le solicita al servidor una búsqueda de usuario para obtener su IP. 3. El servidor le proporciona la dirección IP del cliente destino. 4. El cliente origen le envía una solicitud de llamada al cliente destino. 5. El cliente destino le responde al cliente origen la aceptación de la llamada. <p>Se inicia la llamada</p>	
Flujo Alternativo:	
<ol style="list-style-type: none"> 3. El servidor le indica al cliente origen que el nombre de usuario no existe. 5. El cliente destino le responde al cliente origen el rechazo de la llamada. 	
Pos condiciones:	

Código:	CS-VoIPAS-06.
Nombre:	Colgar
Autor:	Jesús Antonio Argáez Xool
Fecha:	09/09/2008
Descripción:	Permite la terminación de la llamada entre dos usuarios.
Actores:	<ol style="list-style-type: none"> 1. Usuario. 2. Aplicación Cliente.
Precondiciones:	<ol style="list-style-type: none"> 1. Los usuarios deben estar realizando una llamada
Flujo Normal:	<ol style="list-style-type: none"> 1. El cliente envía una solicitud de colgado al otro cliente y termina la transmisión de datos. 2. El cliente que recibe la solicitud termina la transmisión de datos.
Flujo Alternativo:	
Pos condiciones:	

Código:	CS-VoIPAS-07.
Nombre:	Guardar Mediciones
Autor:	Jesús Antonio Argáez Xool
Fecha:	09/09/2008
Descripción:	<p>Permite guardar todos los parámetros deseados, que fueron medidos durante una llamada.</p>
Actores:	<ol style="list-style-type: none"> 1. Usuario. 2. Aplicación Cliente.
Precondiciones:	<ol style="list-style-type: none"> 1. Una llamada tuvo que haber sido realizada.
Flujo Normal:	<ol style="list-style-type: none"> 1. El usuario selecciona el guardado de parámetros medidos. 2. El usuario selecciona los parámetros que desea guardar. 3. El usuario elige la ruta donde se desea guardar. 4. El cliente guarda todos los parámetros elegidos en archivos diferentes.
Flujo Alternativo:	<ol style="list-style-type: none"> 4. Si el nombre ya existe lo indica al usuario.
Pos condiciones:	

Código:	CS-VoIPAS-08.
Nombre:	Graficar Mediciones
Autor:	Jesús Antonio Argáez Xool
Fecha:	09/09/2008
Descripción:	<p>Permite graficar todos los parámetros deseados, que fueron medidos durante una llamada.</p>
Actores:	<ol style="list-style-type: none"> 1. Usuario. 2. Aplicación Cliente.
Precondiciones:	<ol style="list-style-type: none"> 1. Una llamada tuvo que haber sido realizada.
Flujo Normal:	<ol style="list-style-type: none"> 1. El usuario selecciona el graficar los parámetros medidos. 2. El usuario selecciona los parámetros que desea graficar. 3. El cliente grafica todos los parámetros elegidos en ventanas diferentes.
Flujo Alternativo:	
Pos condiciones:	

Código:	CS-VoIPAS-09.
Nombre:	Terminar
Autor:	Jesús Antonio Argáez Xool
Fecha:	09/09/2008
Descripción:	Termina la ejecución de aplicación cliente.
Actores:	<ol style="list-style-type: none"> 1. Usuario. 2. Aplicación Cliente.
Precondiciones:	<ol style="list-style-type: none"> 1. La aplicación cliente se debe encontrar en ejecución.
Flujo Normal:	<ol style="list-style-type: none"> 1. El usuario selecciona el cierre de la aplicación cliente. 2. La aplicación cliente envía una solicitud de colgado si se encuentra en ese momento participando en una llamada. 3. La aplicación cliente envía una solicitud de desconexión al servidor para su borrado de la tabla de direcciones.
Flujo Alternativo:	
Pos condiciones:	

5.2 ARQUITECTURA DEL SISTEMA DE VOIP

En esta sección se presentaran distintos gráficos, que mostraran la arquitectura del sistema con el fin de ver y entender los componentes que integran este sistema.

Los diagramas siguientes mostraran la arquitectura del sistema.

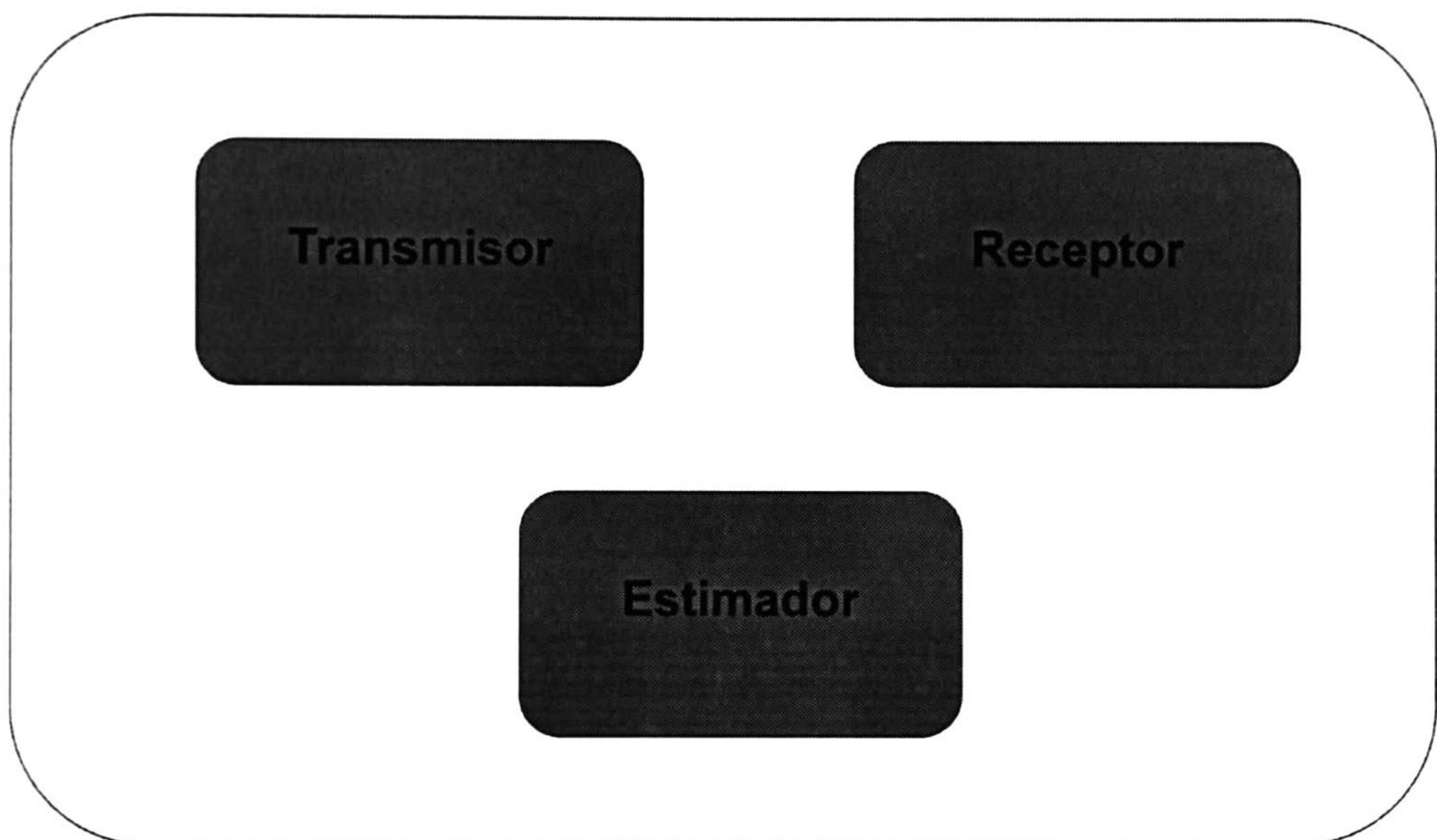


Figura 5-2 Sistema

En la figura 5-2 se observa el bloque principal del sistema, aunque el sistema desde el punto de vista del usuario sea una aplicación, se puede dividir en diferentes módulos los cuales cumplen cierta tarea específica, dichos módulos son cuatro y se pueden observar en la figura 5-2 y se describen a continuación:

1. **Transmisor:** Es el encargado de capturar el audio, codificarlo y transmitirlo. Se muestra en la figura 5-3.

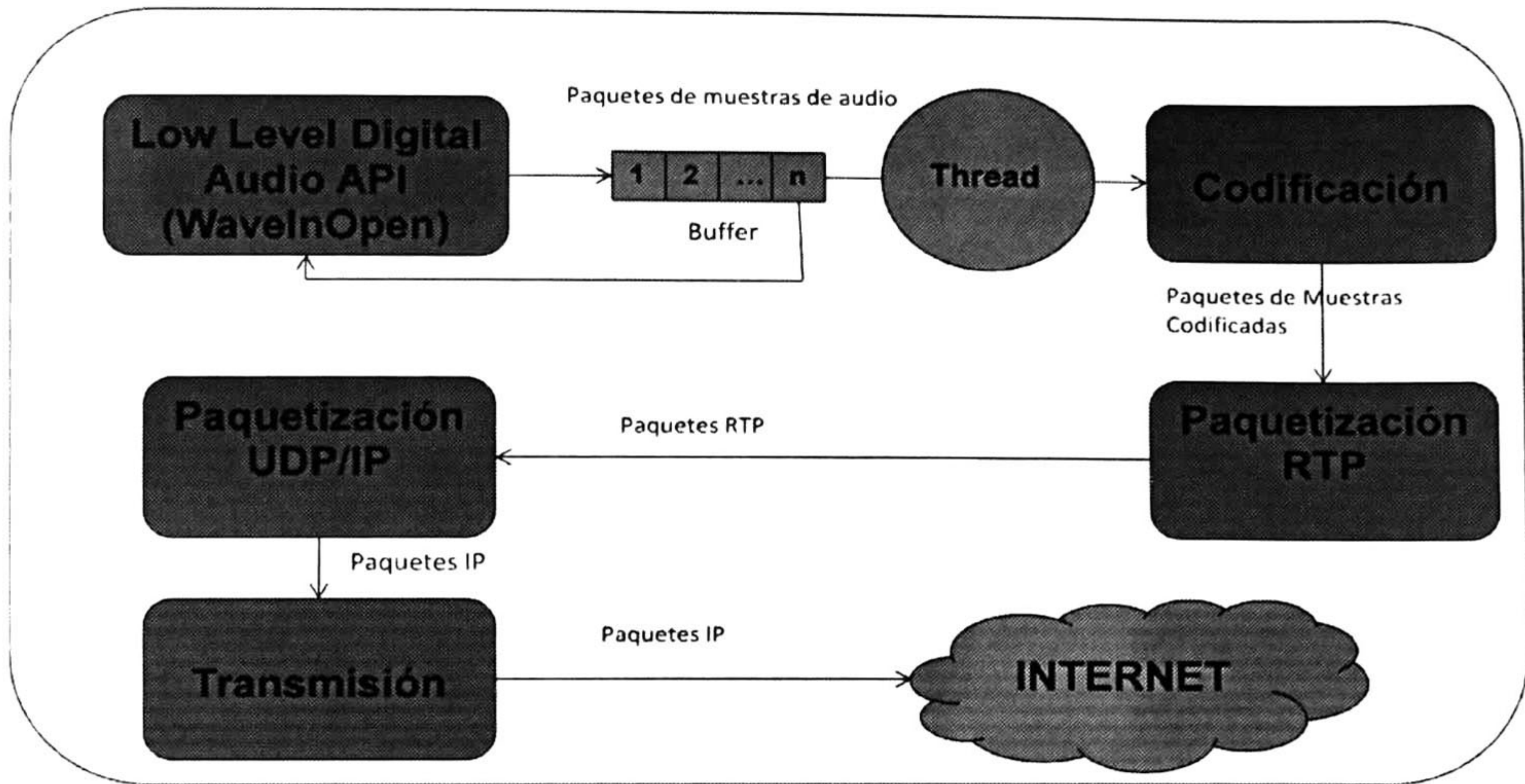


Figura 5-3 Bloque Transmisor.

Como se observa en la figura anterior el transmisor realiza la captura de audio haciendo uno de la programación de bajo nivel, con la API de Windows, usando la función WaveInOpen.

Posteriormente la almacena en un buffer para luego ser codificada con el codec que se desee y forma los paquetes de cierta longitud que el usuario defina. Este a su vez se monta en un paquete RTP, y este a su vez se monta en un paquete UDP y se transmite a través de un socket a la red.

2. **Receptor:** Se encarga de recibir los paquetes, reproducir el audio y extraer los datos de los paquetes para ser proporcionados al Estimador. Figura 5-4.

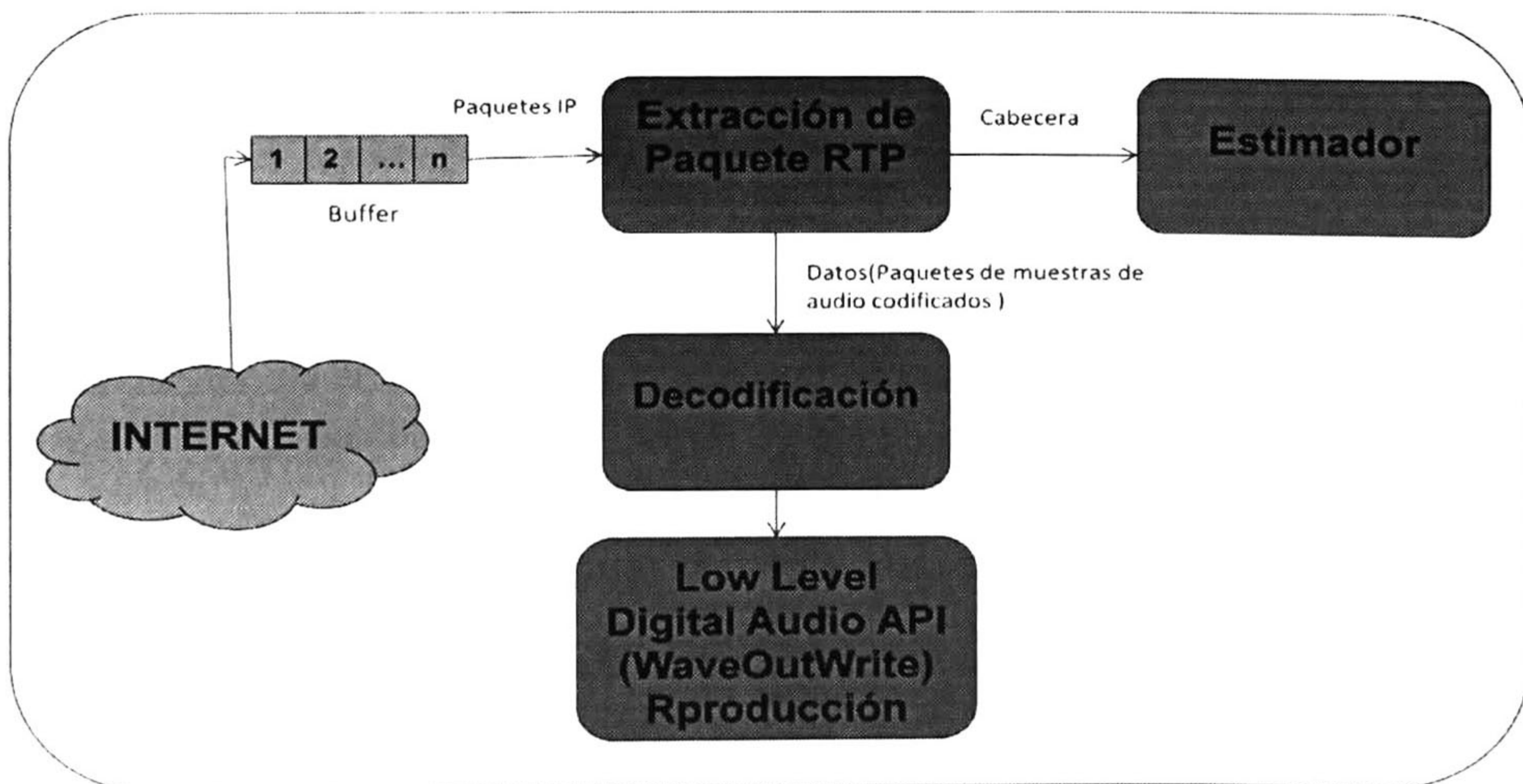


Figura 5-4 Bloque Receptor.

Este bloque es casi función inversa del bloque Transmisor. Este recibe los paquetes de la red extrayendo el paquete RTP, ya que aquí se encuentran las dos partes importantes:

- a. **La cabecera:** Contiene todos los datos necesarios para las estadísticas.
- b. **Los Datos:** Contiene la voz que se transmitió.

Una vez extraídos los datos antes mencionados, son enviados al estimador para realizar los cálculos de los parámetros y la reproducción del audio.

3. **Estimador:** Se encarga de calcular y estimar ciertos parámetros de calidad de servicio ya definidos. Recibe la cabecera de un paquete RTP y ya calcula dichos parámetros.

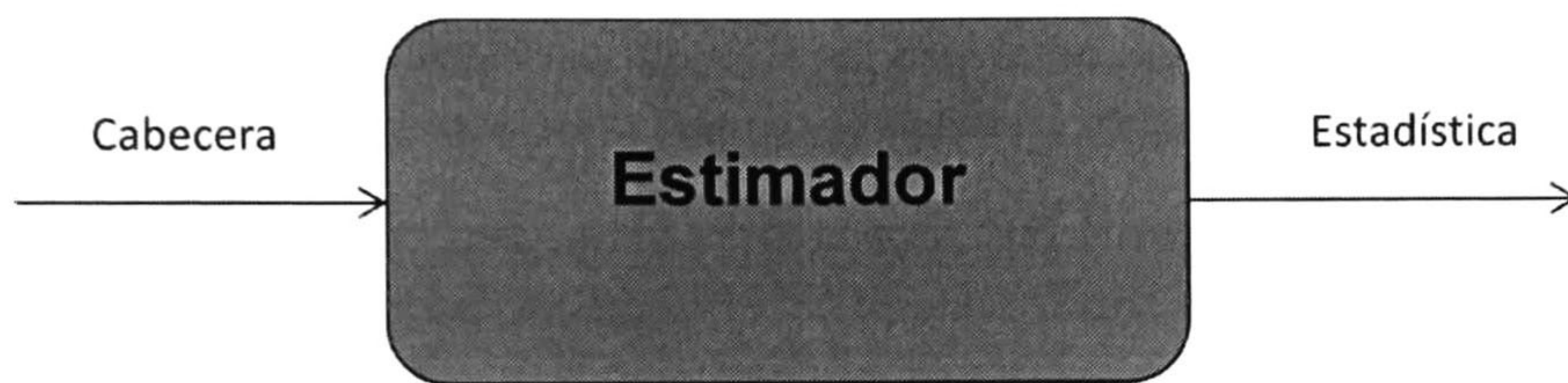


Figura 5-5 Estimador.

5.3 DISEÑO ORIENTADO A OBJETOS

El sistema fue desarrollado en c++, un lenguaje orientado a objetos, por ello se utilizo un paradigma orientado objetos. También se reutilizo código que ya se había desarrollado en otro trabajo [1].

A continuación listaremos las clases que se utilizan en esta aplicación, primeramente las clases que se desarrollaron específicamente para esta aplicación y continuamente las que se reutilizaron.

Clases Desarrolladas

Nombre de la Clase	Objetivo Principal
TfMain	Es la ventana principal del sistema.
TfConfiguration	Configurar la aplicación como los puertos de control, frecuencia de actualización, repuesta automática, directorio de salida y nombre de usuario.
TfnSession	Permite configurar una llamada indicando el códec, tamaño de paquete, dirección ip destino, método del cálculo del parámetro de Hurst, tamaño de la ventana para el cálculo del parámetro de Hurst y el tipo de captura que se desea realizar.
Tfsaveparam	Permite guardar de forma manual las trazas de los parámetros calculados.
Tfgraphparam	Permite gráficas los parámetros que se han calculado.
RTPPacket	Crear Paquetes RTP y manipular su contenido y estructura.
RTPSession	Permite establecer una sesión de comunicación de

paquetes RTP.	
Estadísticas	Se encarga de calcular los parámetros durante la llamada.
Utilities	Un Conjunto de funciones de uso general.

Tabla 5-2 Clases Desarrolladas

A continuación listaremos las clases que reutilizaron del trabajo antes mencionado, solo nos limitaremos a enlistarlas, para más información de estas ir a la referencia indicada.

Clases Reutilizadas

Nombre de la Clase	Objetivo Principal
TRSCalculus	Estimar el valor de H para una serie de tiempo, por el método del estadístico R/S.
TABCalculus	Estimar el valor de H para una serie de tiempo, por el método de Momento Absoluto.
TVarianceMethod	Estimar el valor de H para una serie de tiempo, por el método de la varianza agregada.
TVarResiduals	Estimar el valor de H para una serie de tiempo, por el método de la varianza de residuales.
TMavar	Estimar el valor de H para una serie de tiempo, por el método de la varianza modificada de Allan.
TPeriodogram	Estimar el valor de H para una serie de tiempo, por el método del periodograma.

TWhittle	Estimar el valor de H para una serie de tiempo, por el método de Whittle Local.
Hurst_AV	Estimar el valor de H para una serie de tiempo, por el método de Abry Veitch.
TSelqosChartForm	Gestionar la visualización de un gráfico bidimensional y las opciones de configuración del mismo. Este es el módulo principal para la presentación de resultados visuales en la aplicación.
TTitlesForm	Configurar las opciones generales para desplegar un gráfico.
TBorderForm	Configurar los estilos de borde de las diferentes secciones de un gráfico.
TPatternForm	Configurar el patrón utilizado para mostrar el título en un gráfico.
TScalesForm	Configurar las escalas de los ejes cartesianos en un gráfico.
TAbout	Mostrar la información de los desarrolladores del componente de graficación.

Tabla 5-3 Clases Reutilizadas

5.4 COBERTURA DE REQUERIMIENTOS

A continuación listaremos los requerimientos que se establecieron para el desarrollo de este sistema y cuáles son las clases que se desarrollaron para cumplir con estos requerimientos.

Requerimiento	Clase(s) que cumple(n) con este requerimiento
RF-VoIPAS-01	RTPSession
RF-VoIPAS-02	TfnSession
RF-VoIPAS-03	TfnSession
RF-VoIPAS-04	Estadísticas
RF-VoIPAS-05	Estadísticas, TRSCalculus, TABCalculus, TVarianceMethod, TVarResiduals, TMavar, TPeriodogram, TWhittle, Hurst_AV
RF-VoIPAS-06	Estadísticas
RF-VoIPAS-07	Tfsaveparam
RF-VoIPAS-08	TfMain
RF-VoIPAS-09	Tfsaveparam
RF-VoIPAS-10	Tfgraphparam, TselqosChartForm, TTitlesForm, TBorderForm
RF-VoIPAS-11	TselqosChartForm, TTitlesForm, TBorderForm
RF-VoIPAS-12	TselqosChartForm, TTitlesForm, TBorderForm
RF-VoIPAS-13	TfConfiguration
RF-VoIPAS-14	TfConfiguration

Tabla 5-4 Coberturas de Requerimientos

5.5 INTERFAZ DEL SISTEMA

Existen infinidad de formas para desarrollar una interfaz del sistema, sin embargo, se optó por desarrollar una interfaz muy sencilla con accesos rápidos y sin tantas complicaciones y listaremos a continuación:

1. Ventana Principal

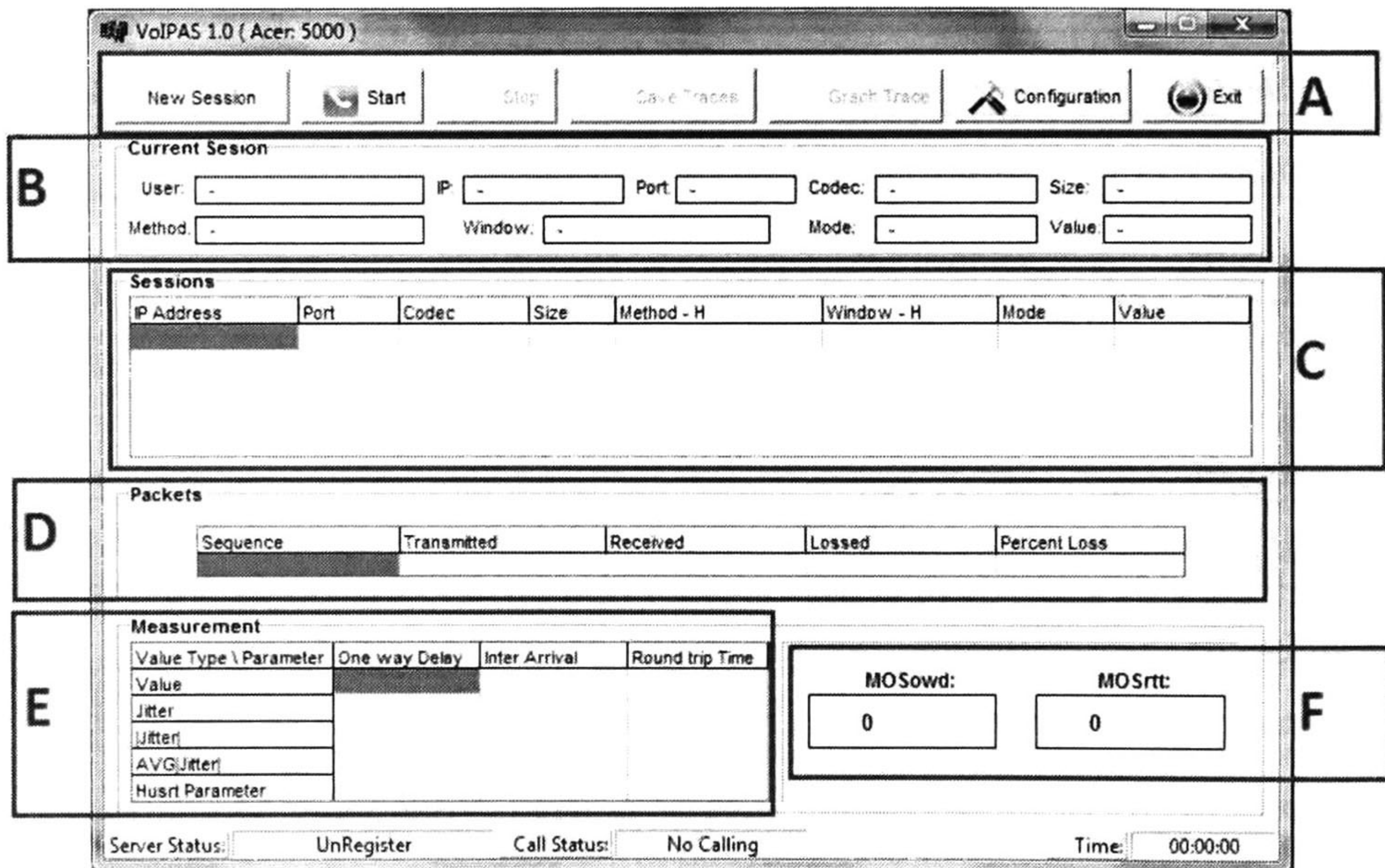


Figura 5-6 Ventana Principal

Esta es la ventana inicial del sistema, aquí se pueden observar seis secciones principales:

- A. Panel de botones de acceso directo: En este panel se encuentran los botones que nos permiten realizar las funciones principales del sistema.
 - i. New Session.- Abre una ventana que nos permite configurar una nueva sesión de medición.
 - ii. Start.- Inicia una sesión previamente creada y configurada.
 - iii. Stop.- Finaliza la sesión que se encuentre en desarrollo.
 - iv. Save Traces.- Permite elegir que parámetros o parámetros deseamos guardar en archivos con extensión txt. Este es un

guardado manual si así se requiere, ya que cuando se configura un grupo de sesiones están se guardarán automáticamente al terminar y antes de empezar la siguiente sesión.

- v. Graph Traces.- G los diferentes parámetros que se hayan medido.
 - vi. Configuration.- C las diferentes características del sistema.
 - vii. Exit.- Finaliza la aplicación.
- B. Panel de sesión actual: Muestra los datos de la sesión que se encuentre o desarrollo o la última que se haya realizado.
- C. Panel de sesiones configuradas: Aquí se listan las diferentes sesiones que se hayan configurado y que faltan por ejecutar.
- D. Panel de información de paquetes: Muestra toda la información de los paquetes:
- i. El número de secuencia de paquete que se está transmitiendo.
 - ii. Número de paquetes transmitidos.
 - iii. Número de paquetes recibidos.
 - iv. Número de paquetes perdidos.
 - v. Porcentaje de pérdidas de paquetes.
- E. Panel de mediciones: Muestra los resultados de los diferentes parámetros medidos, actualizando los resultados cada determinado tiempo que se haya configurado.
- F. Panel de MOS calculado: Muestra los valores del MOS RTT y MOS OWD calculados.

2. Ventana de Configuración.

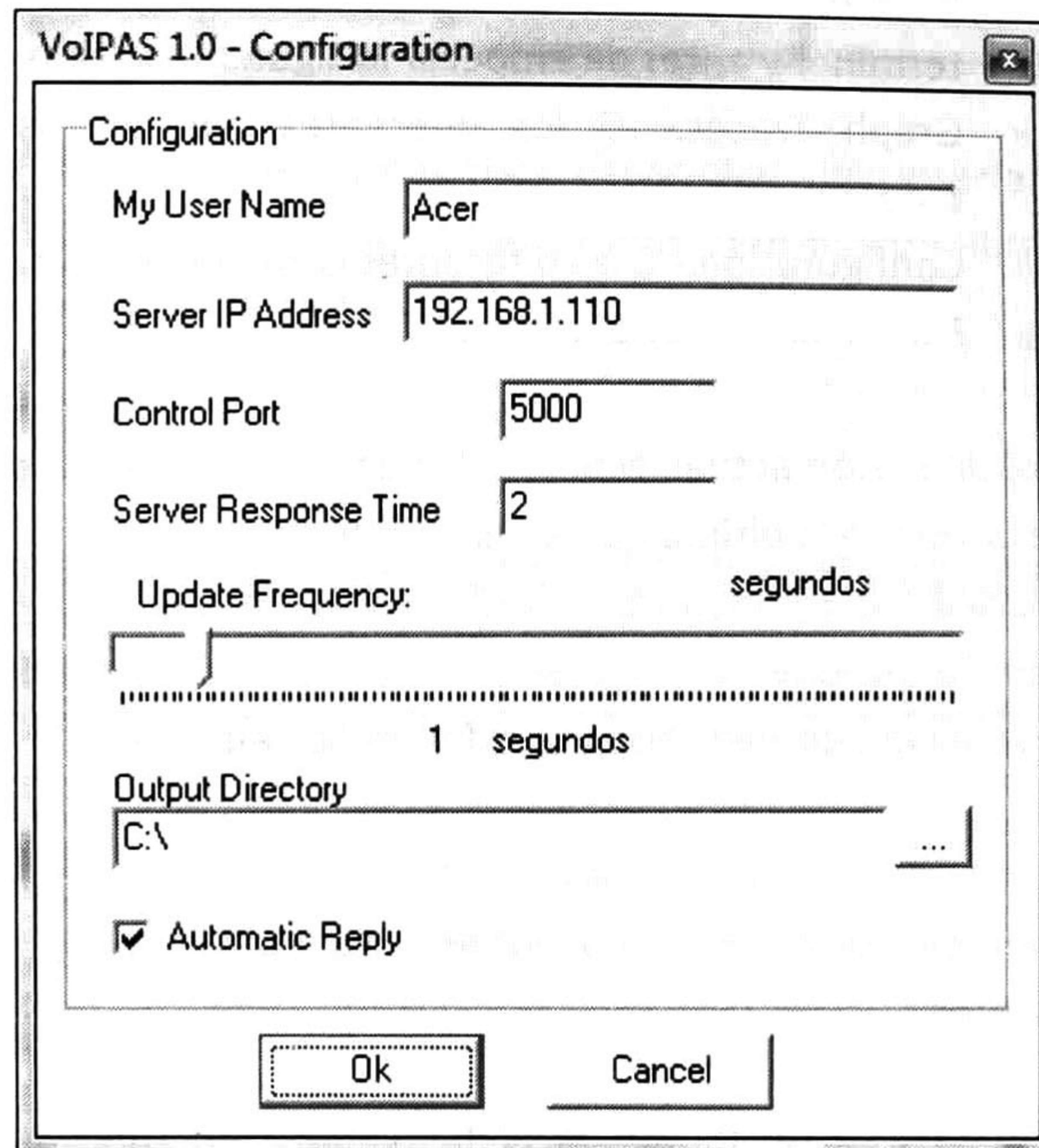


Figura 5-7 Ventana de Configuración

La ventana de configuración nos permite, darle los valores a ciertas características del sistema:

- a. My User Name.- Es el nombre con el cual nos identificaremos con otras instancias de la aplicación.
- b. Server IP Address.- Es la dirección IP del servidor, este es muy importante si no sabemos la ip destino, sino solamente el nombre de usuario de este.
- c. Control Port.- Es el puerto por donde se mandaran las señales de control entre ambas aplicaciones.
- d. Server Response Time.- Es el tiempo, en segundos, que se esperara para que el servidor responda; una vez pasado este tiempo y no haya respuesta del servidor se le indicara al usuario que el servidor no respondió a la petición.
- e. Update Frequency.- Este es el tiempo, en segundos, en que los parámetros medidos y presentados en la ventana principal serán actualizados.
- f. Output Directory.- Esta es la ruta hacia un directorio, el cual será utilizado para guardar automáticamente las mediciones que se realicen por paquetes o por minutos.
- g. Automatic Reply.- Si esta activada esta opción cuando nos realicen una petición la contestación será automáticamente, si esta no estuviera activada al recibir

una petición de comunicación se nos mostrara un mensaje si deseamos contestarla o no.

3. Ventana de Nueva Sesión

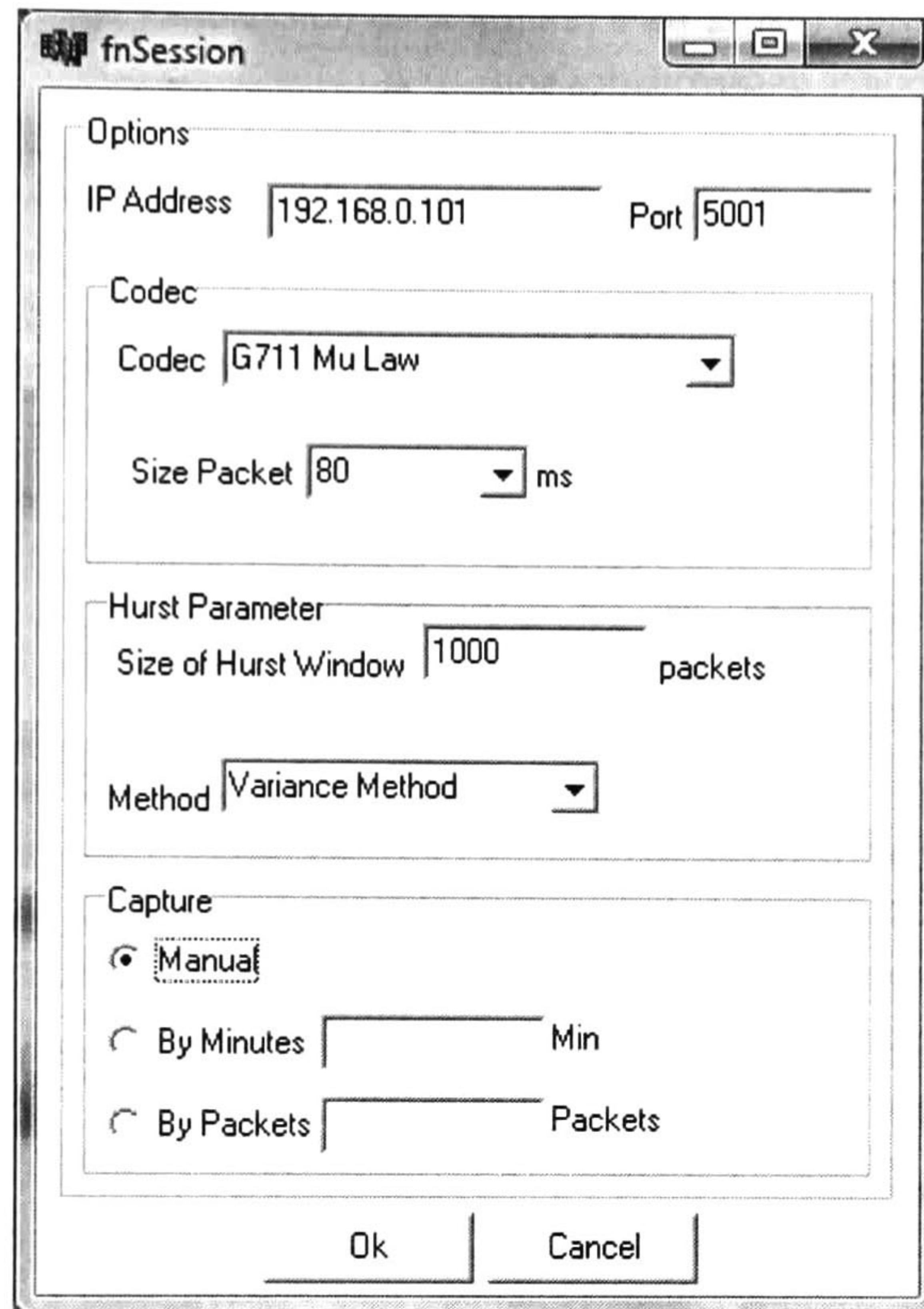


Figura 5-8 Ventana de Nueva Sesión

En esta ventana podremos configurar las diferentes sesiones que deseemos realizar, las opciones que podemos configurar se divide en 4 grupos:

A. Destino.

En este grupo es para indicar con quien nos queremos comunicar, indicándole la dirección ip del destino y el numero de puerto rtp, ya que puede haber varias instancias del sistema en la misma pc pero comunicándose por diferentes puertos.

B. Codec.

Aquí tenemos que indicar el tipo de códec y el tamaño de paquete que queremos utilizar para realizar la comunicación.

Los codecs desarrollados son:

- PCM
- G711 A Law
- G711 Mu Law

Y los posibles tamaños de paquetes que podemos seleccionar son:

- 10 ms
- 20 ms
- 40 ms
- 60 ms
- 80 ms

C. Parametro de Hurst

En esta opción podemos configurar con que método vamos a calcular el parámetro de Hurst y de que tamaño será la ventana de paquete sobre la cual vamos a estar haciendo este cálculo.

Entre los métodos disponibles están:

- Variance Method
- Variance of residuals
- Modified Allan Variance
- Periodogram
- LWhittle
- Abry-Veitch Method

D. Modo de Captura

Por último habría que indicar el modo de captura que deseamos, habiendo 3 métodos distintos:

- Manual.
Este modo significa que nosotros una vez iniciado la comunicación, estaremos obligados nosotros a terminarla.
- By Minutes.
En este modo tenemos que proporcionar un número deseado de minutos después de los cuales, la aplicación terminara la comunicación, guardara los resultados de forma automática y continuara con las siguientes sesiones configuradas, si es el caso de que haya más sesiones configuradas y encoladas.

- **By Packet.**
Mus similar al modo anterior solo que aquí la condición para terminar la comunicación será el número de paquetes que hayamos seleccionado en la configuración de la sesión.

4. La Ventana de Guardado de Estadísticas

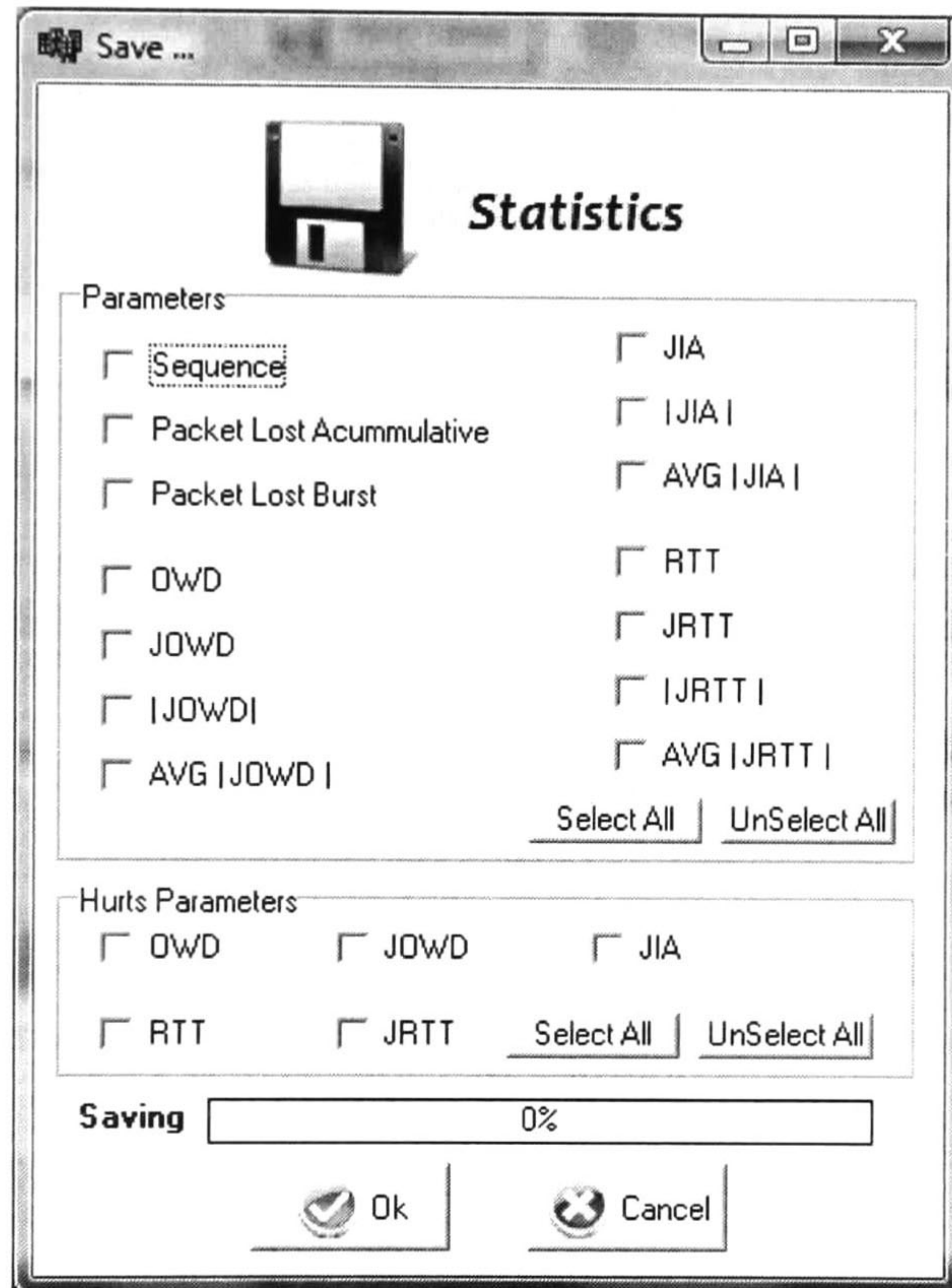


Figura 5-9 Ventana de Guardado de Estadísticas

Esta ventana nos permite guardar las estadísticas de forma manual de los parámetros calculados durante la comunicación, ya sea porque queremos seleccionar un directorio diferente para guardar o porque seleccionamos el modo de captura manual.

5. La Ventana de Graficación de Parámetros

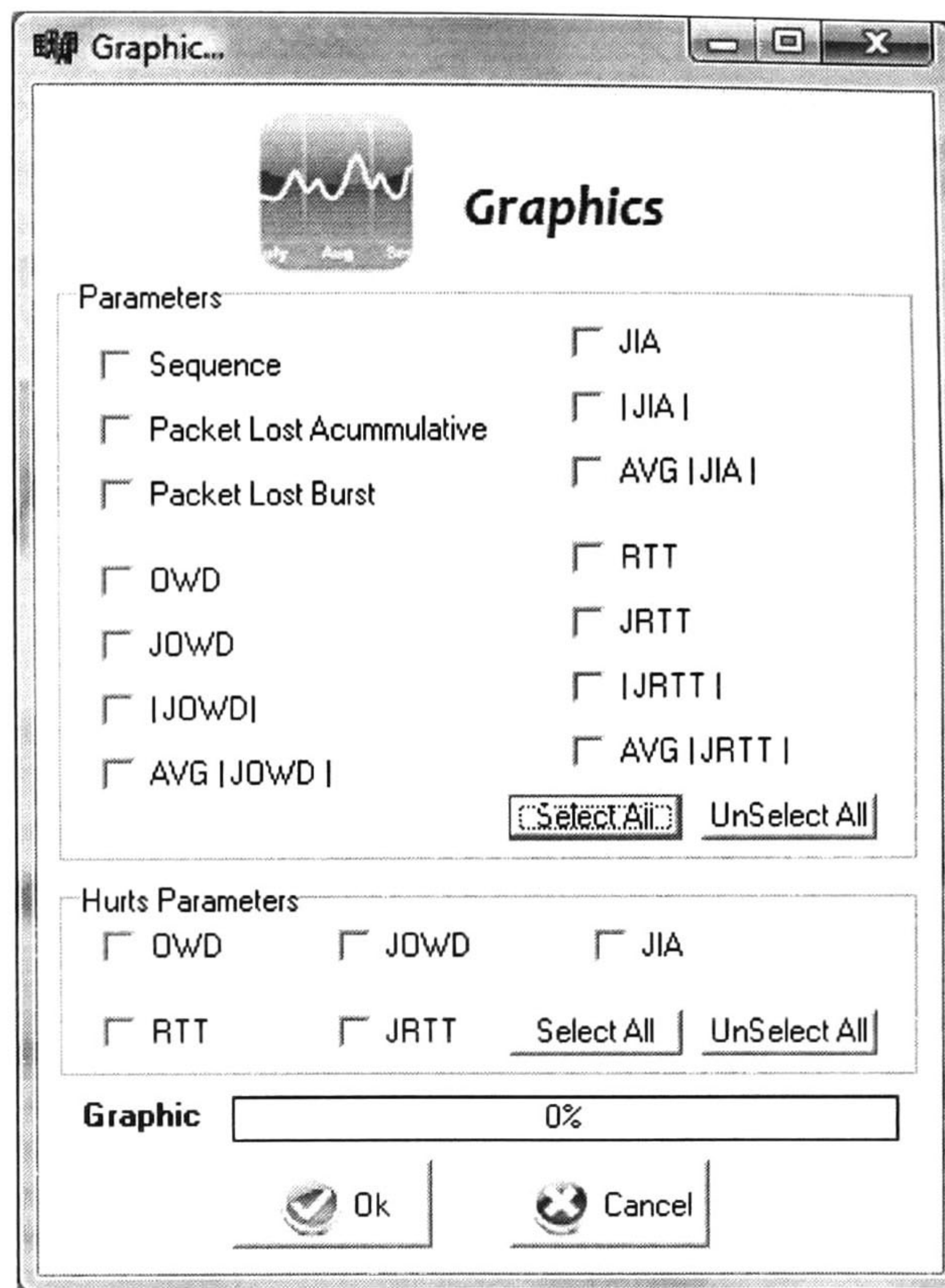


Figura 5-10 Ventana de Graficación de Parámetros

En esta ventana podemos seleccionar los distintos parámetros que deseamos graficar

6. La Ventana de la Grafica de Parámetros

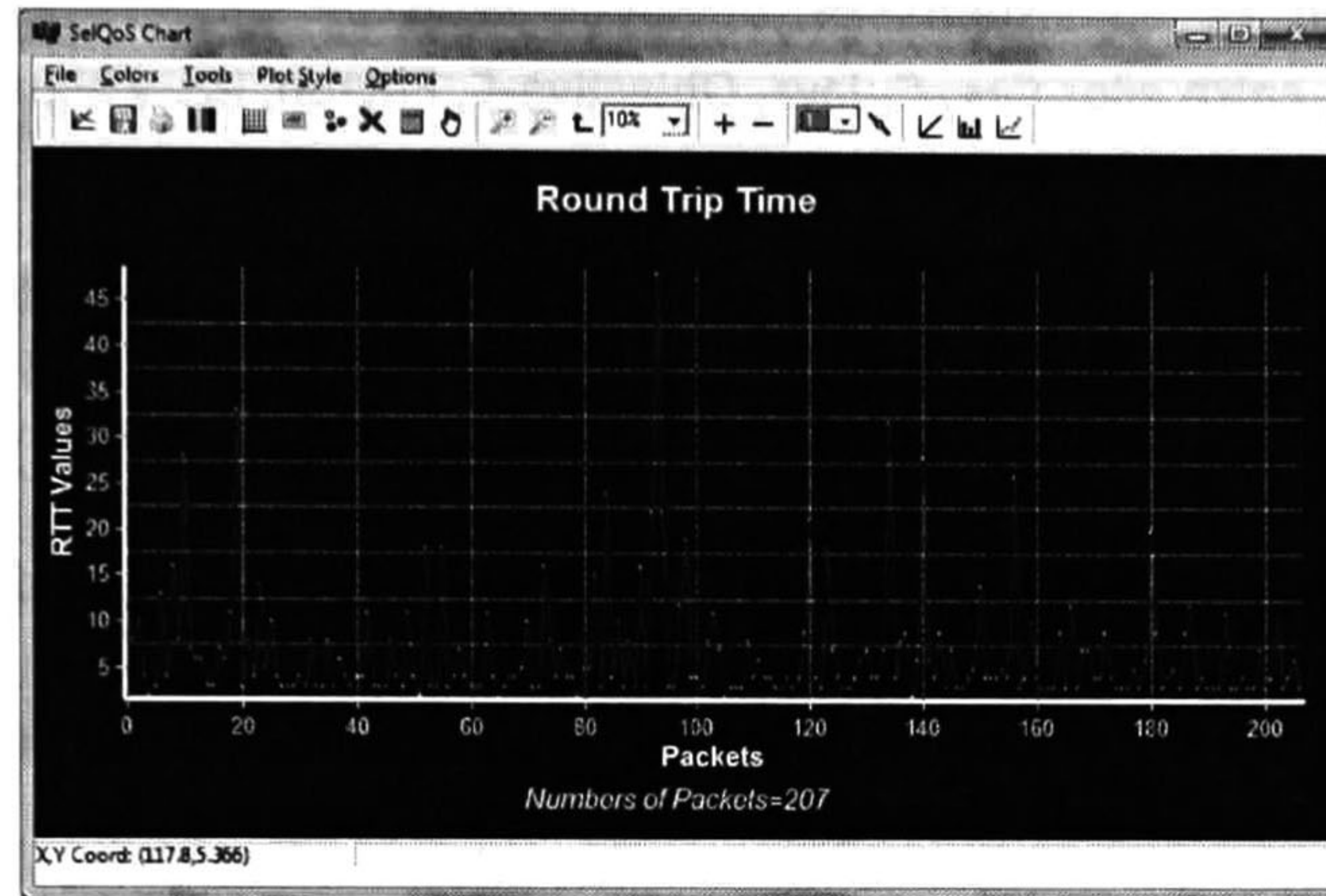


Figura 5-11 Ventana de la Gráfica de un Parámetro

Esta ventana de graficación fue reutilizada de la aplicación SelQoS [35], agregándole unas pequeñas modificaciones y variantes para poder adaptarla a este sistema.

Como se puede observar en el diseño de la interfaz se optó por una interfaz sencilla y cómoda de utilizar.

5.6 DOXYGEN Y LA DOCUMENTACIÓN DEL SISTEMA

Doxygen es un sistema de documentación para diferentes lenguajes de programación, entre ellos C++, C, Java, Objective-C, Python, IDL y algunas extensiones de PHP, C# y D.

Esta herramienta tiene como principal característica la posibilidad de generar navegadores de documentación en línea (HTML) y/o manuales de referencia (RTF, PS), a partir de un conjunto de archivos de código fuente.

Tales archivos habrán de ser convenientemente documentados mediante el estándar que define el software para enriquecer los manuales de referencia, resaltando secciones claves del sistema en desarrollo. Este esquema permite mantener fácilmente la consistencia entre la documentación y el código fuente.

El manual de referencia con la documentación detallada de VoIPAS, generada a través de Doxygen, puede consultarse en el Anexo A –Documentación del código.

Además de documentar el código fuente como tal, Doxygen tiene la capacidad de extraer la estructura del sistema de archivos aún sin documentar; esto es sumamente útil para ubicar un camino en proyectos muy voluminosos. Además, permite visualizar las relaciones entre los elementos del sistema a través de gráficos de dependencia, diagramas de herencia y colaboración, entre otros, los cuales son generados automáticamente.

Doxygen está desarrollado bajo Linux y Mac OS X, pero ha sido configurado para ser altamente portable. Como resultado, a la fecha es posible ejecutarlo en la mayoría de los sistemas Linux e inclusive en Windows.

5.7 CONCEPTOS MATEMÁTICOS

En esta sección se describe el fundamento matemático utilizado para el conjunto de operaciones implementadas en el sistema.

5.7.1 OWD

El OWD o retardo extremo a extremo en un sentido del paquete k , se define como:

$$OWD_k = R_k - S_k \quad R_k > S_k \quad \text{Ecuación 5-1}$$

donde:

S_k : la estampa de tiempo de transmisión del paquete k .

R_k : la estampa de tiempo de recepción del paquete k .

5.7.2 RTT

El RTT o retardo de ida y vuelta del paquete k se define como:

$$RTT_k = S'_k - S_k \quad S'_k > S_k \quad \text{Ecuación 5-2}$$

donde:

S_k : la estampa de tiempo de transmisión del paquete k .

S'_k : la estampa de tiempo de recepción del paquete k , cuando este regresa al transmisor.

5.7.3 JITTER

Existen diferentes definiciones de Jitter o variación de retardo, en este trabajo implementamos las siguientes:

1. OWD

El Jitter de OWD entre dos paquetes consecutivos k y $k+1$ se define como:

$$J_{OWD} = OWD_{k+1} - OWD_k \quad \text{Ecuación 5-3}$$

2. RTT

El Jitter de RTT entre dos paquetes consecutivos k y $k+1$ se define como:

$$J_{RTT} = RTT_{k+1} - RTT_k \quad \text{Ecuación 5-4}$$

5.7.4 ESTIMADORES DEL INDICE DE AUTO-SIMILITUD PARA SERIES DISCRETAS DE TIEMPO

Mientras el parámetro de Hurst como un valor único está perfectamente definido de forma matemática, calcularlo en la realidad es una tarea difícil. Las mediciones deben realizarse a grandes separaciones de tiempo o bajas frecuencias, para lo cual, generalmente, se consiguen menos lecturas [28].

Existen múltiples métodos para estimar el índice de auto-similitud; ninguno de ellos es capaz de determinar el valor exacto pues todos presentan vulnerabilidad ante la periodicidad o tendencias en los datos, no-estacionariedades y otras fuentes de corrupción como secuencias de valores atípicos o cambios abruptos en la media y varianza.

Los estimadores de H pueden clasificarse en tres grupos:

- Los que operan en el dominio del tiempo
- Los que operan en el dominio de la frecuencia
- Los que operan en el dominio de ondeletas (wavelets)

A continuación se describe brevemente cada uno de los estimadores que han sido implementados en VoIPAS.

5.7.4.1 MÉTODO R/S

El estadístico R/S [29] fue uno de los primeros métodos para inferir el grado de auto-similitud en series de tiempo. Fue desarrollado por Edwin Hurst cuando estudiaba la fluctuación de los niveles de agua en el río Nilo.

Dada $Y(t) = \sum_{i=1}^t X(i)$ la suma parcial de la serie $X(t)$ y $S^2(n)$ la varianza,

entonces la estadística R/S se define como:

$$\frac{R}{S}(n) = \frac{1}{S(n)} \left[\max_{0 \leq t \leq n} \left(Y(t) - \frac{t}{n} Y(n) \right) - \min_{0 \leq t \leq n} \left(Y(t) - \frac{t}{n} Y(n) \right) \right] \quad \text{Ecuación 5-5}$$

donde n representa un corrimiento en el tiempo sobre la serie. Hurst descubrió que para ciertos registros históricos, la estadística R/S tiende a ser bien representada por la

relación $E\left\{\frac{R}{S}(n)\right\} \sim cn^H, n \rightarrow \infty$, donde H , es el índice de auto-similitud.

El método de estimación primero divide la serie de tiempo $X(t)$ en K bloques no traslapados de tamaño N/K . Posteriormente para cada corrimiento de tamaño n , se calcula el estadístico R/S comenzando en el inicio de cada bloque, con lo cual se obtiene un conjunto de valores (uno por bloque incluido) para cada n .

Calculando la esperanza matemática de cada conjunto de valores los resultados se reducen a una serie única, con lo cual se puede aprovechar la relación descrita en la ecuación 5.5 para realizar un gráfico log – log de $E\left\{\frac{R}{S}(n)\right\}$ contra n , del cual se espera obtener una línea recta de pendiente H .

5.7.4.2 MÉTODO DEL MOMENTO ABSOLUTO

El método del momento absoluto o del valor absoluto [29][30] funciona calculando la serie agregada de nivel m , posteriormente, se obtiene el momento absoluto de esta serie, definido como:

$$AM^{(m)} = \frac{1}{N/M} \sum_{k=1}^{N/M} |X^{(m)}(k) - \bar{X}| \quad \text{Ecuación 5-6}$$

donde \bar{X} es la media de la serie original $X(t)$.

Dado que

$$AM^{(m)} \sim m^{H-1}, m \rightarrow \infty \quad \text{Ecuación 5-7}$$

entonces un gráfico log – log de los valores de AM contra m debe dar una línea recta con pendiente $H - 1$.

5.7.4.3 MÉTODO DE LA VARIANZA

Al igual que el método del valor absoluto, para el método de la varianza [29][30] se obtiene la serie agregada para un nivel de agregación m dado y entonces se aplica la siguiente fórmula a la nueva serie:

$$Var X^{(m)} = \frac{1}{N/M} \sum_{k=1}^{N/M} (X^{(m)}(k) - \bar{X})^2 \quad \text{Ecuación 5-8}$$

donde \bar{X} es la media de la serie original $X(t)$.

Para este caso

$$Var X^{(m)} \sim m^{2H-2}, m \rightarrow \infty \quad \text{Ecuación 5-9}$$

por lo tanto un gráfico de $\log(Var X^{(m)})$ contra $\log(m)$ debe dar una línea recta con pendiente $2H - 2$.

5.7.4.4 MÉTODO DE LA VARIANZA DE RESIDUALES

En el método de Varianza de Residuales [29][30], la serie $X(t)$ es dividida en bloques de tamaño m y a cada bloque se le calcula la suma parcial

$$Y(t) = \sum_{i=1}^t X(i) \quad \text{Ecuación 5-10}$$

Habiendo hecho esto, se ajusta una línea recta $a + bt$ con los valores de las sumas parciales para cada bloque y entonces se calcula la varianza de residuales de acuerdo a la siguiente fórmula:

$$RES^{(m)} = \frac{1}{m} \sum_{i=1}^m (Y(t) - a - bt)^2 \quad \text{Ecuación 5-11}$$

De lo anterior se obtiene una serie de varianzas de residuos para cada m ; a cada serie se le calcula la *media* o *mediana* para reducir a una sola serie y aplicar la siguiente propiedad:

$$RES^{(m)} \sim m^{2H}, m \rightarrow \infty \quad \text{Ecuación 5-12}$$

Dado esta fórmula se infiera que un gráfico log – log de $RES^{(m)}$ contra m debe dar una línea recta con pendiente $2H$.

5.7.4.5 MÉTODO DE LA VARIANZA MODIFICADA DE ALLAN

Este método se apoya en el uso de la Varianza Modificada de Allan [31], estandarizada por la ITU-T a través del siguiente estimador:

$$Mod \sigma_x^2(\tau) = \frac{1}{2n^4 \tau_0^2 (N - 3n + 1)} \sum_{j=1}^{N-3n+1} \left[\sum_{i=j}^{n+j-1} X(i+2n) - 2X(i+n) + X(i) \right]^2 \quad \text{Ecuación 5-13}$$

donde τ_0 representa el período de muestreo por el que se separan los valores de la serie $X(t)$; $n=1, 2, \dots, \lfloor N/3 \rfloor$ y $\tau=n\tau_0$.

Considerando un número relativamente alto de muestras se ha demostrado que la estimación de la fórmula anterior obedece la ley de potencia

$$\text{Mod } \sigma_x^2(\tau) \sim k\tau^\mu \quad \text{Ecuación 5-14}$$

idealmente cuando $n \rightarrow \infty$ pero en la práctica para $n > 4$.

Por definición

$$\mu = 2H - 4 \quad \text{Ecuación 5-15}$$

de forma que un gráfico logarítmico en ambos ejes arroja una recta con pendiente μ , la cual nos permite calcular la estimación $H = \frac{\mu}{2} + 2$.

5.7.4.6 MÉTODO DEL PERIODOGRAMA

Este método [4][5] se basa en el cálculo del periodograma el cual se define de la siguiente forma:

$$I(\nu) = \frac{1}{2\pi N} \left| \sum_{j=1}^N X(j)e^{j\nu} \right|^2 \quad \text{Ecuación 5-16}$$

donde ν representa la frecuencia.

Si la varianza de $X(t)$ es finita, entonces $I(\nu)$ es un estimador de la densidad espectral de dicha serie; suponiendo que además esta serie presenta dependencia de largo rango (LRD) entonces tendrá una densidad espectral proporcional a $|\nu|^{1-2H}$ cerca del origen.

Por tal razón, un gráfico log – log del periodograma contra la frecuencia debe generar una línea recta con pendiente $1-2H$

En la práctica, sólo se utiliza el 10% de las frecuencias inferiores para calcular el parámetro de Hurst dado que se asume este comportamiento alrededor de cero.

5.7.4.7 MÉTODO DE WHITTLE LOCAL

El método de Whittle estándar [28][29][30][32] es un estimador de máxima probabilidad estrechamente relacionado con el periodograma a través de la siguiente ecuación:

$$Q(\eta) = \int_{-\pi}^{\pi} \frac{I(\nu)}{f(\nu; \eta)} d\nu + \int_{-\pi}^{\pi} \log f(\nu; \eta) d\nu \quad \text{Ecuación 5-17}$$

donde η es un vector de parámetros ajustables respecto a la forma de la densidad espectral de la serie estudiada, $I(\nu)$ es el periodograma y $f(\nu; \eta)$ es la densidad espectral en la frecuencia ν .

Este método tiene la desventaja de que requiere el conocimiento de la forma paramétrica de la densidad espectral; si este requisito no se cumple completamente el estimador puede polarizarse, y ante las características no-gráficas del mismo, dicho efecto es muy difícil de detectar.

La versión semi-paramétrica de este método, conocida como Whittle Local [29][32][33][34], tiene la característica de que asume sólo la forma funcional del espectro cuando las frecuencias se encuentran cerca de cero:

$$f(\nu) \sim G(H)|\nu|^{1-2H}; \nu \rightarrow 0 \quad \text{Ecuación 5-18}$$

Para este espectro de potencia, el análogo discreto de la función Q (ver ecuación 5-19) es:

$$Q(G, H) = \frac{1}{M} \sum_{j=1}^M \left(\frac{I(\nu_j)}{G \nu_j^{1-2H}} + \log G \nu_j^{1-2H} \right) \quad \text{Ecuación 5-19}$$

Este cálculo involucra el parámetro M , $M < N/2$, el cual debe satisfacer además que $\frac{1}{M} + \frac{M}{N} \rightarrow 0$, $N \rightarrow \infty$.

Reemplazando la constante G por su estimado $\hat{G} = M^{-1} \sum_{j=1}^M \frac{I(v_j)}{v_j^{1-2H}}$, la tarea se reduce a minimizar la función $R(H)$, definida como $R(H) = Q(\hat{G}, H) - 1$, es decir:

$$R(H) = \log\left(\frac{1}{M} \sum_{j=1}^M \frac{I(v_j)}{v_j^{1-2H}}\right) - (2H - 1) \frac{1}{M} \sum_{j=1}^M \log v_j \quad \text{Ecuación 5-20}$$

En la literatura algunos autores han sugerido que $M = N/32$ es una porción razonable de frecuencias para estimar el parámetro de Hurst.

5.7.4.8 MÉTODO ABRY-VEITCH

La siguiente descripción y el algoritmo de la sección 0 son cortesía de Leopoldo Estrada Vargas, quien estuvo a cargo del análisis e implementación en SelQoS [35] de este método.

Para estimar el valor de H , el método de Abry-Veitch se apoya en el cálculo de la transformada Wavelet Discreta (DWT), la cual consiste en un análisis multiresolución (MRA) de series de tiempo, es decir, que contiene información entiendo y en frecuencia simultáneamente. El MRA utilizado en la DWT analiza las componentes de frecuencia de la señal con diferentes resoluciones.

El MRA implementado en la DWT tiene buena resolución en tiempo y baja resolución en frecuencia para las altas frecuencias, así como baja resolución en tiempo y alta resolución en frecuencia para las bajas frecuencias. Esto se muestra en la Figura 5-6.

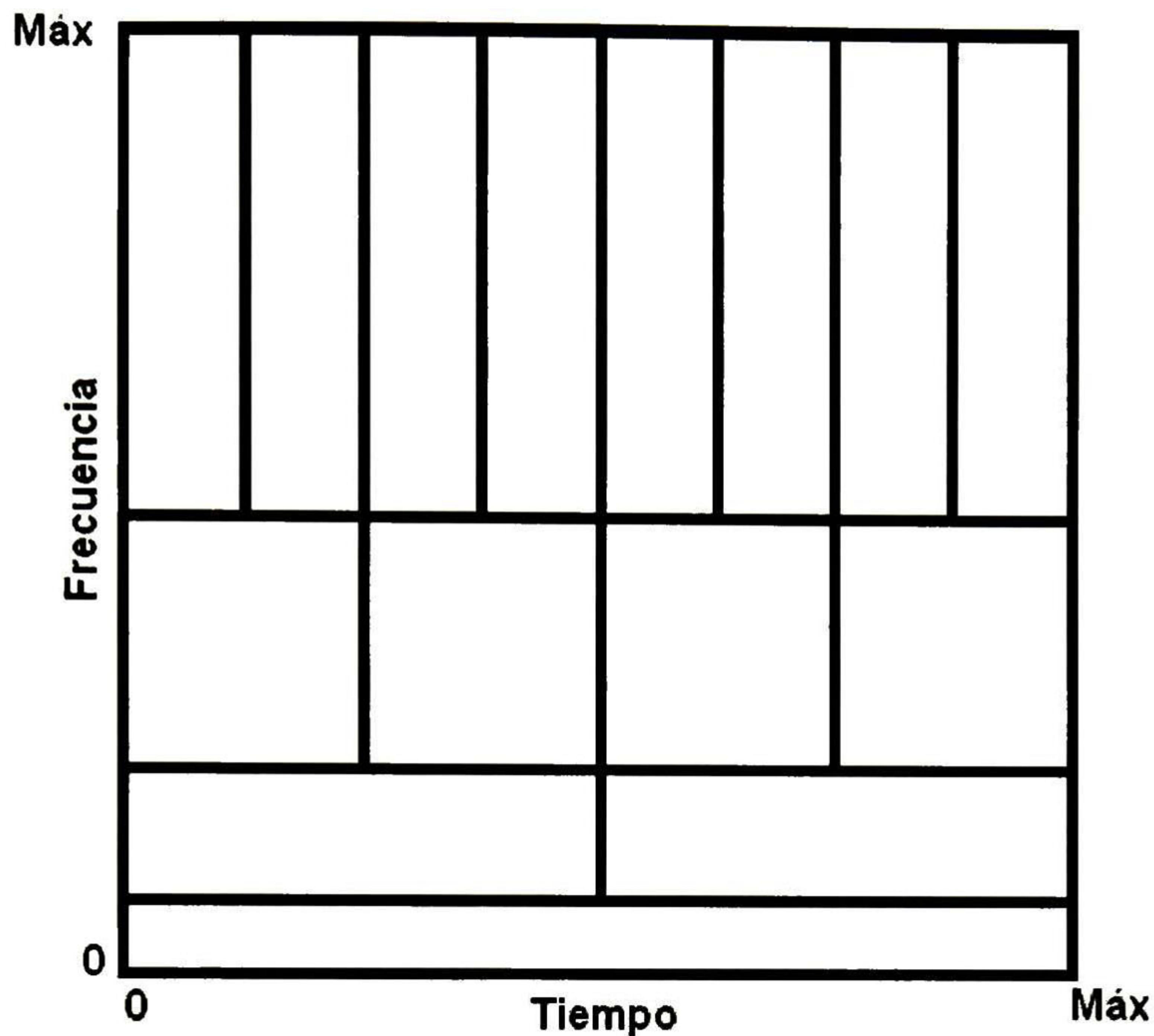


Figura 5-12 Resolución en tiempo y frecuencia del MRA

La implementación seleccionada para la DWT en VoIPAS se presenta en la Figura 5-7. La señal de entrada x se pasa por dos filtros. Un filtro pasa-bajas y un filtro pasa-altas. Esto genera dos señales, cada una contiene la información de la mitad del espectro: La parte alta (Hx) y la parte baja (Lx). Para representar la mitad del espectro es suficiente con la mitad de las muestras, por lo tanto, se hace un diezmado de cada una de las dos trazas obtenidas.

La parte superior del espectro es una parte de la "salida" de la DWT. Para obtener los demás coeficientes se toma la parte baja y se pasa por los dos filtros. Así se obtienen las trazas respectivas LLX y HLX. HLX es otra parte de la salida y LLx se vuelve a filtrar y así sucesivamente.

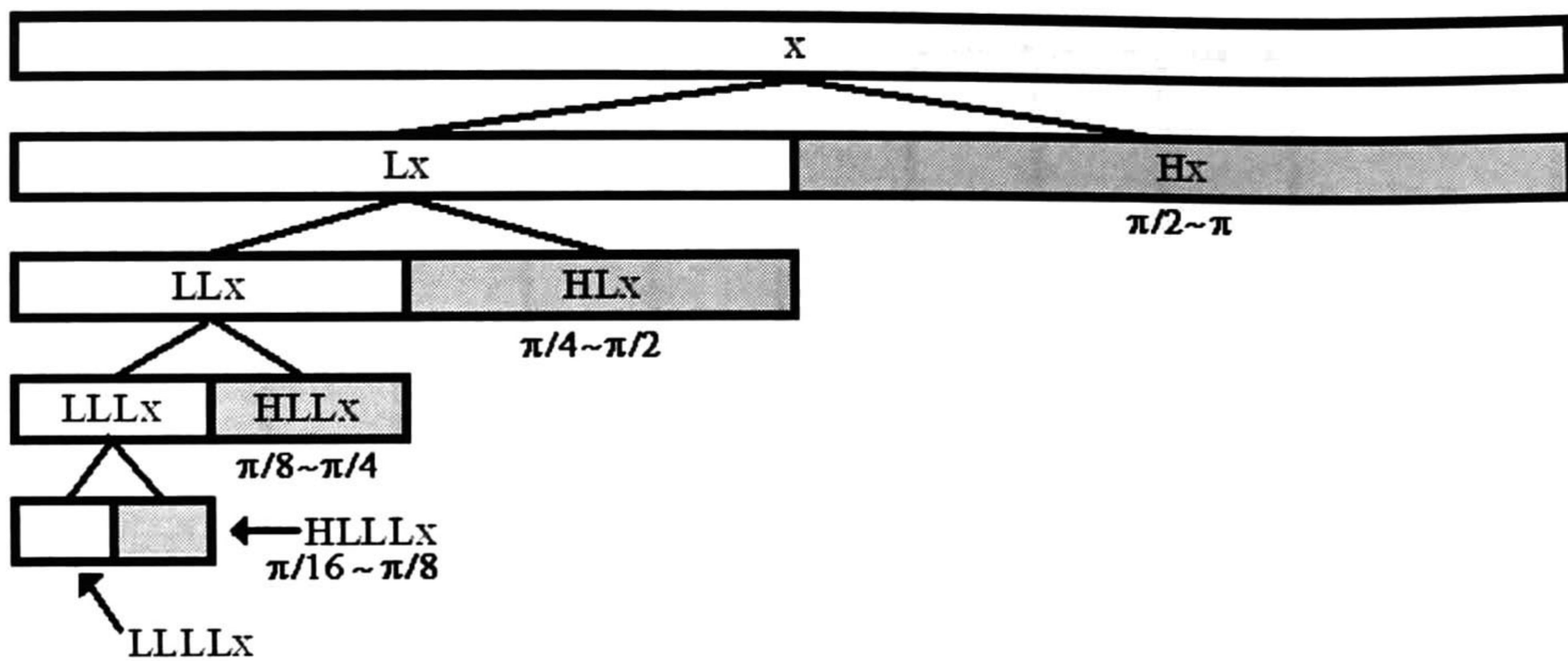


Figura 5-13 Implementación de la transformada de Wavelet Discreta

Las trazas Hx , HLx , etc., se concatenan para formar la DWT como se muestra en la Figura 5-8.

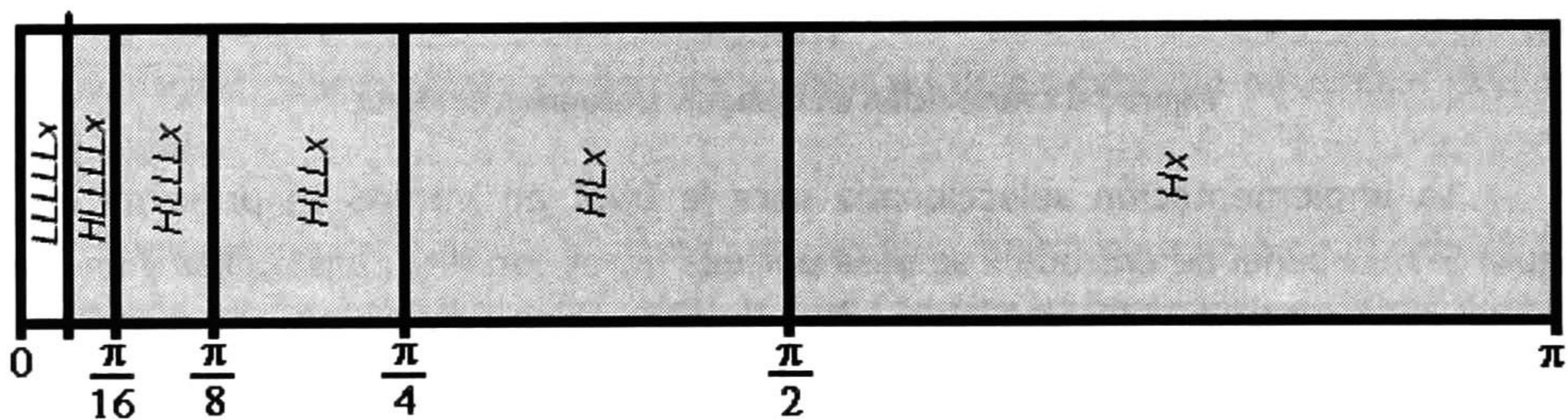


Figura 5-14 Salida de la transformada de Wavelet Discreta

El cálculo del parámetro de Hurst basado en la DWT considera que la densidad espectral cumple con una ley de potencia, como se muestra en la siguiente ecuación.

$$f_x(v) \sim c_f |v|^{-\alpha}, |v| \rightarrow 0 \quad \text{Ecuación 5-21}$$

Cada bloque de la Figura 5-8 representa una parte del espectro. Las componentes medias respectivas de cada uno de ellos es la mitad de la del bloque superior, es decir, forman una serie geométrica con razón de 2.

Entonces, el método para estimar el parámetro α de la ecuación anterior consiste en obtener un gráfico log-log de frecuencia media contra energía por bloque y calcular la pendiente mediante un algoritmo de regresión. Generalmente se suele reemplazar el eje de la frecuencia media por el de número de octava, que representa

la cantidad de filtros que se aplicaron para llegar a dicho bloque. Esto afecta la ordenada al origen pero no la pendiente de la regresión lineal.

Es bien sabido que la relación de H y α es la siguiente ecuación.

$$\alpha = 2H - 1 \leftrightarrow H = \frac{1+\alpha}{2} = \frac{1-m}{2} \quad \text{Ecuación 5-22}$$

donde m es la pendiente de la ecuación de la recta obtenida mediante regresión.

5.7.5 MOS

El cálculo del MOS implementado en el sistema está basado en el Modelo E [24] que presenta un método para estimar la QoS.

El Modelo E es un modelo computacional que combina un conjunto de parámetros en un valor total, llamado factor R que tiene un rango de 0 a 100, este valor es convertido a escalas de MOS para predecir niveles de calidad de servicio.

El factor R es expresado como la suma de 4 términos:

$$R = 100 - I_s - I_d - I_{ef} + A \quad \text{Ecuación 5-23}$$

donde:

- I_s es una función del error de cuantificación ocasionado por codecs con una tasa de codificación baja.
- I_d es principalmente una función del nivel y retardo del eco con respecto a la señal original y los retardos boca a oído en ambas direcciones (i.e. OWD).
- I_{ef} cubre las eventualidades causadas por lo que G.107 [24] define como el uso de equipo especial, es una función del tipo de codificador utilizado (i.e. CodecTyp) y de la fracción de paquetes perdidos (i.e. PLR).
- A es el factor de expectación, el cual captura el hecho que los usuarios pueden aceptar algo de degradación en calidad en cambio de una comunicación más rápida.

Tomando los valores por defecto que de los parámetros I_s y A , especificados en [24], podemos reducir la ecuación 5-23 a:

$$R = 93.2 - I_d - I_{ef} \quad \text{Ecuación 5-24}$$

Apoyándonos de [27] obtenemos la siguiente expresión para los términos I_d e I_{ef} :

$$I_d = 0.024d + 0.11(d - 177.3)H(d - 177.3) \quad \text{Ecuación 5-25}$$

donde:

$$H(x) \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases} \quad \text{Ecuación 5-26}$$

$$d = OWD \quad \text{ó} \quad d = \frac{RTT}{2}$$

$$I_{ef} \sim \gamma_1 + \gamma_2 \ln(1 + \gamma_3 e) \quad \text{Ecuación 5-27}$$

donde :

$e =$ La probabilidad total de paquetes peridos (toma valores de 0 a 1)

$\gamma =$ Valores fijos, que dependen del tipo de codec que se utilice

Sustituyendo las ecuaciones 5-25 y 5-27 en 5-24 obtenemos la expresión final del factor R:

$$R \sim a - \beta_1 d - \beta_2 (d - \beta_3) H(d - \beta_3) - \gamma_1 - \gamma_2 \ln(1 + \gamma_3 e) \quad \text{Ecuación 5-28}$$

donde:

$$a = 93.2 \quad \beta_1 = 0.024 \quad \beta_2 = 0.11 \quad \beta_3 = 177.3$$

para G.711 tenemos

$$\gamma_1 = 0 \quad \gamma_2 = 30 \quad \gamma_3 = 15$$

Y para G.729

$$\gamma_1 = 11 \quad \gamma_2 = 40 \quad \gamma_3 = 10$$

La relación entre el factor R y el MOS está dada por la siguiente expresión [27]:

$$MOS = \begin{cases} 1 & R < 0 \\ 4.5 & R > 100 \\ 1 + 0.035R + 7 \times 10^{-6} R(R - 60)(100 - R) & 0 < R \leq 100 \end{cases}$$

A continuación se muestra la tabla con los valores del factor R, la opinión de los usuarios y el valor del MOS asociado.

Factor R	Opinión de los Usuarios	MOS
90 < R < 100	Muy satisfecho	4.34 - 4.5
80 < R < 90	Satisfecho	4.03 - 4.34
70 < R < 80	Algunos usuarios inconformes	3.60 - 4.03
60 < R < 70	Muchos usuarios inconformes	3.10 - 3.60
0 < R < 60	Casi todos los usuarios inconformes	2.58 - 3.10

Tabla 5-5 Factor R, la opinión de los usuarios y MOS asociado

6 ANÁLISIS Y EVALUACIÓN DEL SISTEMA

En este capítulo estudiaremos los problemas que comprende el desarrollo de este sistema, además, analizaremos su desempeño y su correcto funcionamiento del sistema. Se quiso también realizar una comparativa con otro sistema como punto de referencia para demostrar la importancia y mejora de este sistema, pero no fue posible, debido a que no existe otro sistema similar a este.

Por ello que nos limitaremos a analizar el uso de la memoria y el procesador, utilizados por el sistema, durante su funcionamiento y los resultados proporcionados por este con ciertas mediciones realizadas.

También es importante señalar que el diseño y desarrollo del sistema representó todo un reto debido a que se requiere tiempo de gran resolución en los procesos periódicos como la transmisión constante de los paquetes a una cierta tasa siendo el valor de esta tasa igual al tamaño del paquete. Otro problema a enfrentar era la implementación de las estampas de tiempo ya que el sistema de Windows no nos provee estampas de tiempo precisas.

Se ha mencionado que la resolución del tiempo es un factor muy importante en esta herramienta, y el reto es debido a que los sistemas operativos de hoy en día son multitarea, es decir, los recursos del sistema son compartidos entre las diversas tareas que se ejecuten en el sistema, por lo tanto no tenemos a nuestra entera disposición todos los recursos.

Otro detalle es que la resolución mínima en la ejecución de procesos periódicos, como thread y timers, en el sistema es de 15 ms. Resolución que no nos serviría ya que nosotros empleamos tamaños de paquetes de 10ms, es por esto que requeriríamos de una resolución menor de 10 ms.

En los siguientes apartados se discutirá como se enfrentaron estos problemas y se probara el cumplimiento de estos requerimientos.

6.1 ANÁLISIS DE DESEMPEÑO

Las pruebas de memoria y procesador se realizaron en un equipo con las siguientes características:

- Computadora Portátil Acer Aspire 5100.
- Memoria (RAM) de 1278 MB.
- Procesador AMD Turion(tm) 64 X2 Mobile Technology TL-50 1.60 GHz.
- Memoria de Video Dedicada ATI Radeon de 256 MB.
- Sistema Operativo de 32 bits Windows Vista Ultimate.

La siguiente figura muestra la aplicación al iniciarla.

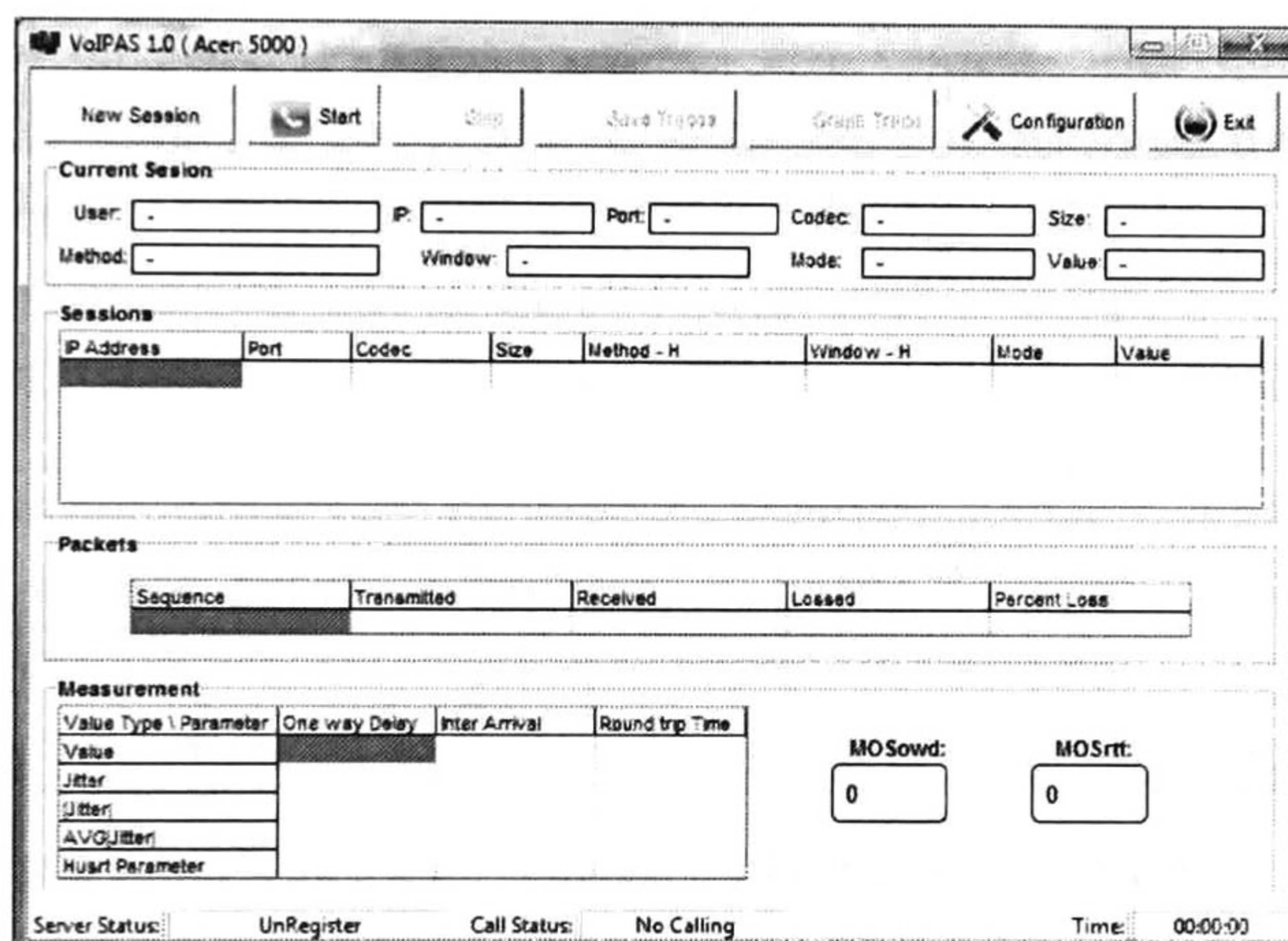


Figura 6-1 Pantalla Principal de VoIPAS

Al iniciar la aplicación tiene un consumo de memoria de 2.044 KB y 0% del procesador.

Cuando se pone en marcha el sistema para la transmisión y medición de los parámetros de calidad de servicio la memoria aumenta en base a la cantidad de tiempo y/o paquetes transmitidos y capturados.

El uso de procesador varía solamente del 0% al 4%. La cantidad de memoria requerida varía dependiendo del tamaño de paquete seleccionado, no tanto del códec, se realizaron 5 conjunto de mediciones con el códec G711 A Law y los diversos tamaño de paquetes: 10ms, 20ms, 40ms, 60ms y 80ms, con duración de una hora. Y los resultados del uso de procesador y memoria RAM requerida se listan en la tabla 4.1.

Códec	Tamaño de Paquete (ms)	Uso de Procesador (%)	Memoria RAM (KB)
G711 A Law	10	0 – 3	135,368
	20	0 – 2	70,240
	40	0 – 1	35,928
	60	0 – 1	24,300
	80	0 – 1	18,692
G711 Mu Law	10	0 – 2	135,724
	20	0 – 2	70,144
	40	0 – 1	35,048
	60	0 – 1	24,572
	80	0 – 1	18,876

Tabla 6-1 Comparativa de uso de CPU y memoria RAM para el códec G729 A Law

El uso de memoria RAM es inversamente proporcional al tamaño de los paquetes transmitidos, es decir, mientras disminuimos el tamaño de paquete pequeño el uso de la memoria aumentará; obviamente ya que cuando se recibe un paquete se generan los cálculos de la estadísticas, mencionadas en los requerimientos funcionales RF-VoIPAS-05, RF-VoIPAS-06 y RF-VoIPAS-07 el apartado de requerimientos funcionales descritos en el capítulo 3, y se guardan en la memoria principal para su uso futuro, como el cálculo de los parámetros de HURST.

Se eligió ser guardada en la memoria principal ya que de hacer lo en algún tipo memoria secundaria, como disco duro u otro dispositivo de almacenamiento, provocaría un gran retardo en la aplicación, ya que los accesos de escritura y lectura a los dispositivos de almacenamiento secundario toman un tiempo considerable en los tipos de aplicaciones de tiempo real que requieren una gran resolución de tiempo en sus procesos como se explica en el siguiente apartado.

La resolución en el tiempo de ejecución y planificación periódica de los procesos es un factor importante en la funcionalidad de este sistema. Los métodos más habituales utilizados para ejecutar procesos periódicos con alta resolución de tiempo son los Thread y los Timers y Multimedia Timers.

6.2.1 TEMPORIZADORES

Threads

Un thread, en sistemas operativos, es una característica que permite a una aplicación realizar varias tareas concurrentemente. Los distintos threads comparten una serie de recursos tales como el espacio de memoria, los archivos abiertos, situación de autenticación, etc. Esta técnica permite simplificar el diseño de una aplicación que debe llevar a cabo distintas funciones simultáneamente.

Los threads que comparten los mismos recursos, sumados a estos recursos, son en conjunto conocidos como un proceso. El hecho de que threads de un mismo proceso compartan los recursos hace que cualquiera de estos hilos pueda modificar éstos. Cuando un hilo modifica un dato en la memoria, los otros threads acceden a ese dato modificado inmediatamente.

Lo que es propio de cada thread es el contador de programa, la pila de ejecución y el estado de la CPU (incluyendo el valor de los registros).

El proceso sigue en ejecución mientras al menos uno de sus threads siga activo. Cuando el proceso finaliza, todos sus threads también han terminado. Asimismo en el momento en el que todos los threads finalizan, el proceso no existe más y todos sus recursos son liberados.

Los threads son muy buenos para la ejecución de procesos periódicos y concurrentes, ya que comparten ciertos recursos y variables, pero la resolución mínima de tiempo con la que cuenta es de 10 ms.

Timers

Todos los procesos aplicación de Windows, que se derivan de los mensajes del sistema operativo, y un temporizador de Windows es simplemente un objeto que es manejado por el sistema operativo. Cuando el tiempo especificado haya transcurrido, un mensaje WM_TIMER es puesto en la cola de mensajes de la aplicación. Cuando este mensaje WM_TIMER es procesado por la cola de mensajes de la aplicación, uno de los dos posibles modos de activación se producen, dependiendo de cómo este configurado el temporizador.

El Uso de mensajes WM_TIMER sufre de una limitación importante:

- Los mensajes WM_TIMER tienen menor prioridad que otros mensajes. Esto significa que los otros mensajes serán trasladados por delante de WM_TIMER mensajes en una aplicación de la cola de mensajes. Esto sucede incluso con temporizadores que están configurados para utilizar funciones de devolución de llamada el sentido de que los acontecimientos puede demorarse cuando el sistema está muy ocupado.

Los sistemas temporizadores en general tienen los siguientes inconvenientes:

- Los temporizadores se limitan a la resolución del sistema - típicamente 50 milisegundos en los más antiguos sistemas de PC, 15 milisegundos en los más nuevos sistemas de PC, como se muestra en el cuadro siguiente.
- El número de temporizadores disponibles es limitado.

Sistema Operativo	Resolución (ms)	Número máximo de temporizadores multimedia disponibles para las aplicaciones de 32 bits
Windows 95	55	32
Windows 98	55	32
Windows NT 3.51	10	16 por proceso
Windows NT 4.0	10	16 por proceso
Windows 2000	10	16 por proceso
Windows XP	10-15	16 por proceso
Windows Vista	10-15	16 por proceso

Tabla 6-2 Numero de Temporizadores y Resolución en los diferentes S.O.

Multimedia Timers

Los servicios de Multimedia Timers permiten a las aplicaciones planear los temporizadores de eventos con la mayor resolución posible para la plataforma de hardware. Estos servicios permiten al usuario programar temporizadores de eventos con una resolución más alta que otros servicios de temporizador.

Estos servicios de tiempo son útiles para aplicaciones que requieren alta resolución. Por ejemplo, un secuenciador MIDI requiere un temporizador de alta resolución, ya que debe mantener el ritmo de eventos MIDI dentro de una resolución de 1 milisegundo.

En base a los diferentes tipos de temporizadores antes descritos, justificamos el uso de los Multimedia Timers en nuestra aplicación.

6.2.2 PRUEBA DE RESOLUCIÓN DE TIEMPO

Ahora probaremos que tan bien funciona la resolución de tiempo en nuestro sistema, después de implementar los Timers Multimedia. Para ello montamos una red local con 2 equipos y un router como se muestra en la figura 4.2.

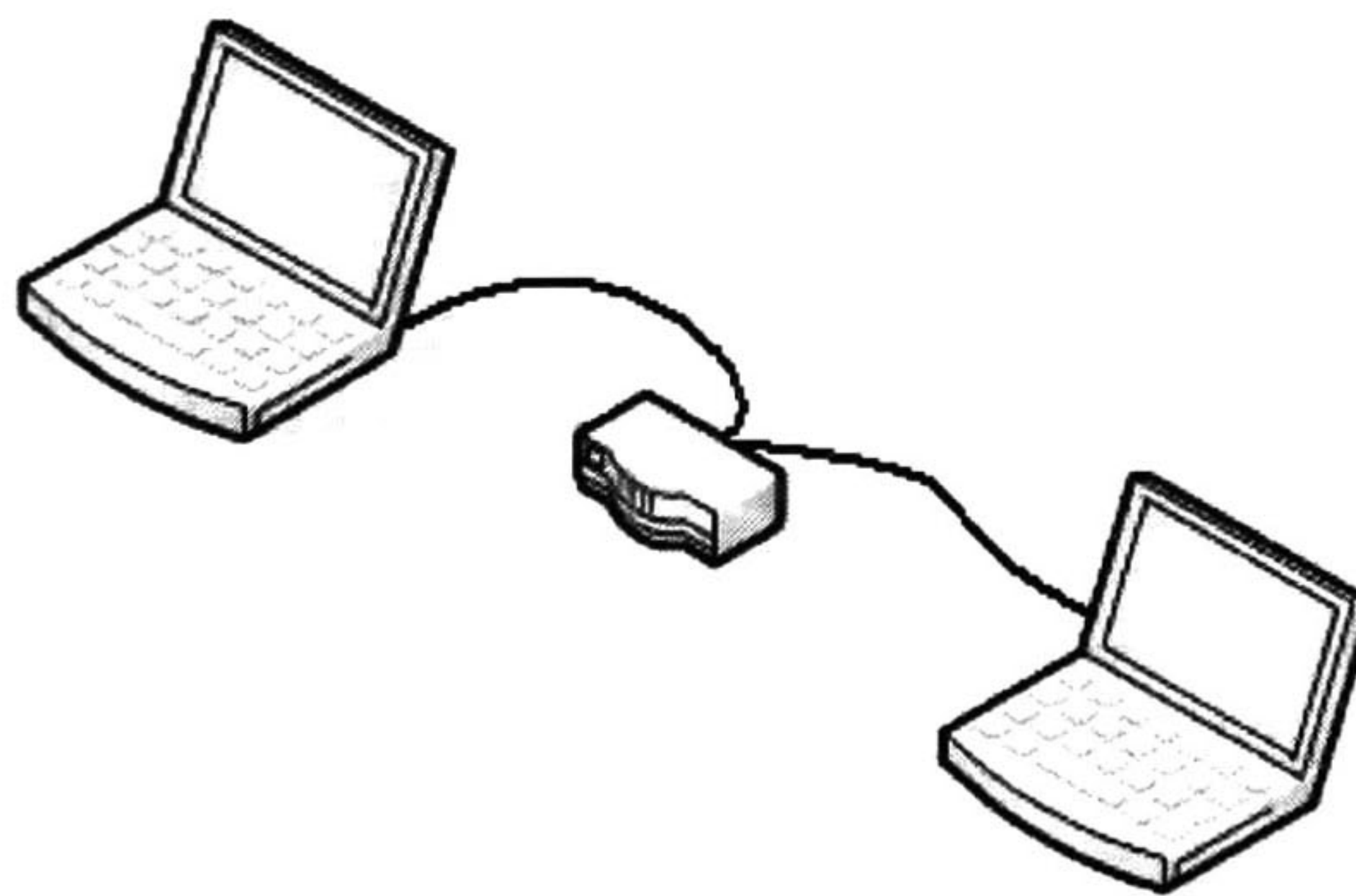


Figura 6-2 LAN de Prueba

La idea es sencilla se usa una simple LAN con 2 pc, cada una de ellas con nuestra aplicación ejecutándose, y realizar un conjunto de mediciones con los diferentes tipos de codecs y tamaños de paquete de nuestro sistema. Cada medición constara de 1000 paquetes, esto para hacer simple la visualización de los resultados esperados.

Una vez obtenidos los resultados se analiza el InterArrival Jitter y el Promedio de este, a lo largo de la medición. El objetivo de la prueba es que, ignorando el retardo que pueda producir la LAN, el InterArrival Jitter y el Average InterArrival Jitter estén sobre el valor del tamaño de paquete con el cual se realizó cada medición. Lo cual es una buena prueba para corroborar que nuestro sistema está enviando los paquetes con una tasa y resolución de tiempo adecuada.

A continuación se muestran las gráficas de los resultados. En la figura 4.3 se observan los resultados obtenidos con el códec G711 Ley A y los tamaños de paquete de 10ms, 20ms, 40ms, 60ms y 80 ms.

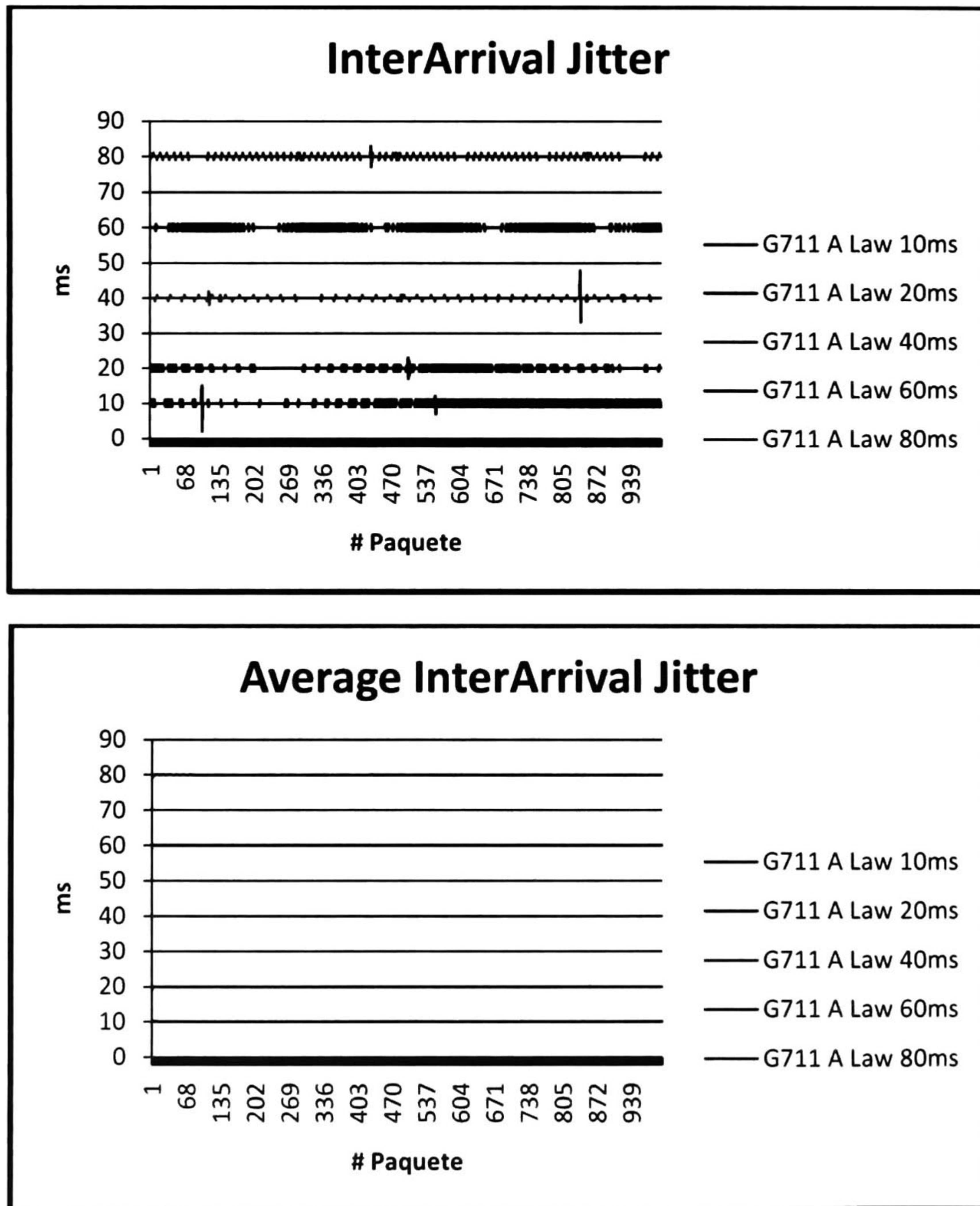


Figura 6-3 G711 A Law

En la figura 4.4 se muestran los resultados obtenidos con el códec G711 Ley Mu y los tamaños de paquete de 10ms, 20ms, 40ms, 60ms y 80 ms.

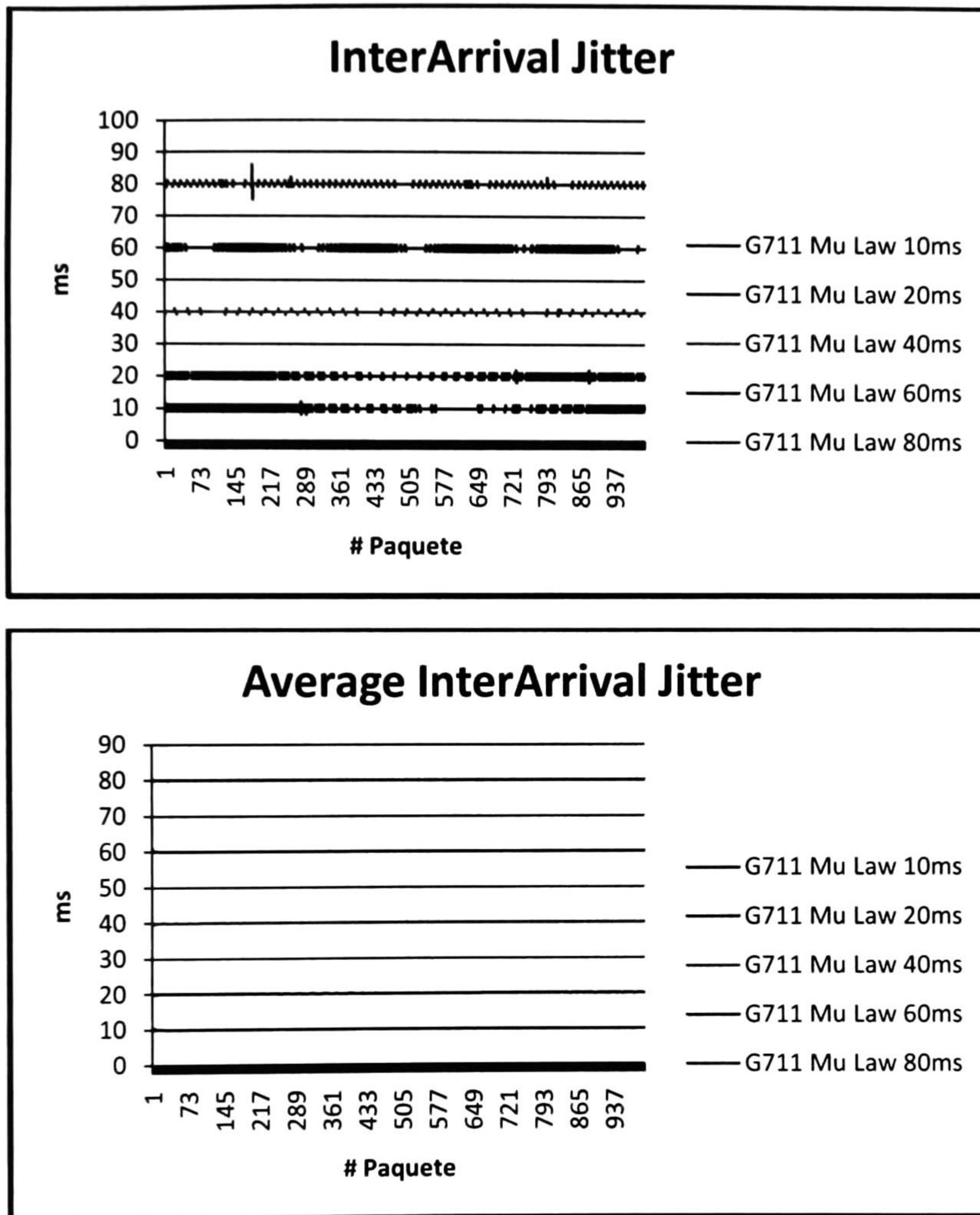


Figura 6-4 G711 Mu Law

Observando las gráficas de los resultados, es obvio que cumplen con la prueba que se había planteado antes, el InterArrival Jitter está sobre el valor del tamaño del paquete, con unas pequeñas variaciones debido a la naturaleza de la red. Sin embargo en las gráficas de Average InterArrival Jitter se corrobora mejor este resultado.

Realizamos una segunda prueba, basada en que en un ambiente ideal el tiempo de inter-arribo está determinado por la siguiente fórmula:

$$IAT_i = (R_i - R_{i-1}) + (PLB_{(i,i-1)})(IDT) \quad \text{Ecuación 6-1}$$

Donde:

IAT_i = Valor del tiempo de inter arribo del i -ésimo paquete.

R_i y R_{i-1} = Estampa de tiempo del paquete i e $i-1$.

IDT (Inter Departure Time) = Tiempo de salida entre paquetes.

$PLB_{(i,i-1)}$ = Numero de paquetes perdidos entre el paquete i e $i-1$.

Despreciando el retardo producido en la LAN podemos hacer la siguiente equivalencia:

$$(R_i - R_{i-1}) = IDT \quad \text{Ecuación 6-2}$$

Sustituyendo la ecuación 6-2 en la ecuación 6-1 y reduciendo la nueva ecuación obtendríamos el siguiente resultado

$$IAT_i = (IDT) + (PLB_{(i,i-1)})(IDT) \quad \text{Ecuación 6-3}$$

$$IAT_i = (PLB_{(i,i-1)} + 1)(IDT) \quad \text{Ecuación 6-4}$$

Tomando en cuenta la ecuación 6-4, se observa que el tiempo de inter arribo es igual a, recordando nuevamente que se desprecia el retardo de la LAN, el numero de paquetes perdidos entre el paquete i e $i-1$, multiplicado por el tiempo de salida entre paquetes. Si no se perdiera ningún paquete entre el paquete i e $i-1$ el tiempo de inter arribo es igual al tiempo de salida entre paquetes.

Se realizaron un grupo de 6 mediciones, que se describen en la tabla 6.3, haciendo que se perdieran cierto número de paquetes aleatorios, durante la medición.

Codec	Tamaño de Paquete	Duración
G711 A Law	10 ms	1 minuto
	20 ms	1 minuto
	80 ms	1 minuto
G711 Mu Law	10 ms	1 minuto
	20 ms	1 minuto
	80 ms	1 minuto

Tabla 6-3 Parámetros de Medición

A continuación se muestran las graficas de las ráfagas de los paquetes perdidos y del Jitter del Inter-Arribo de las diferentes mediciones.

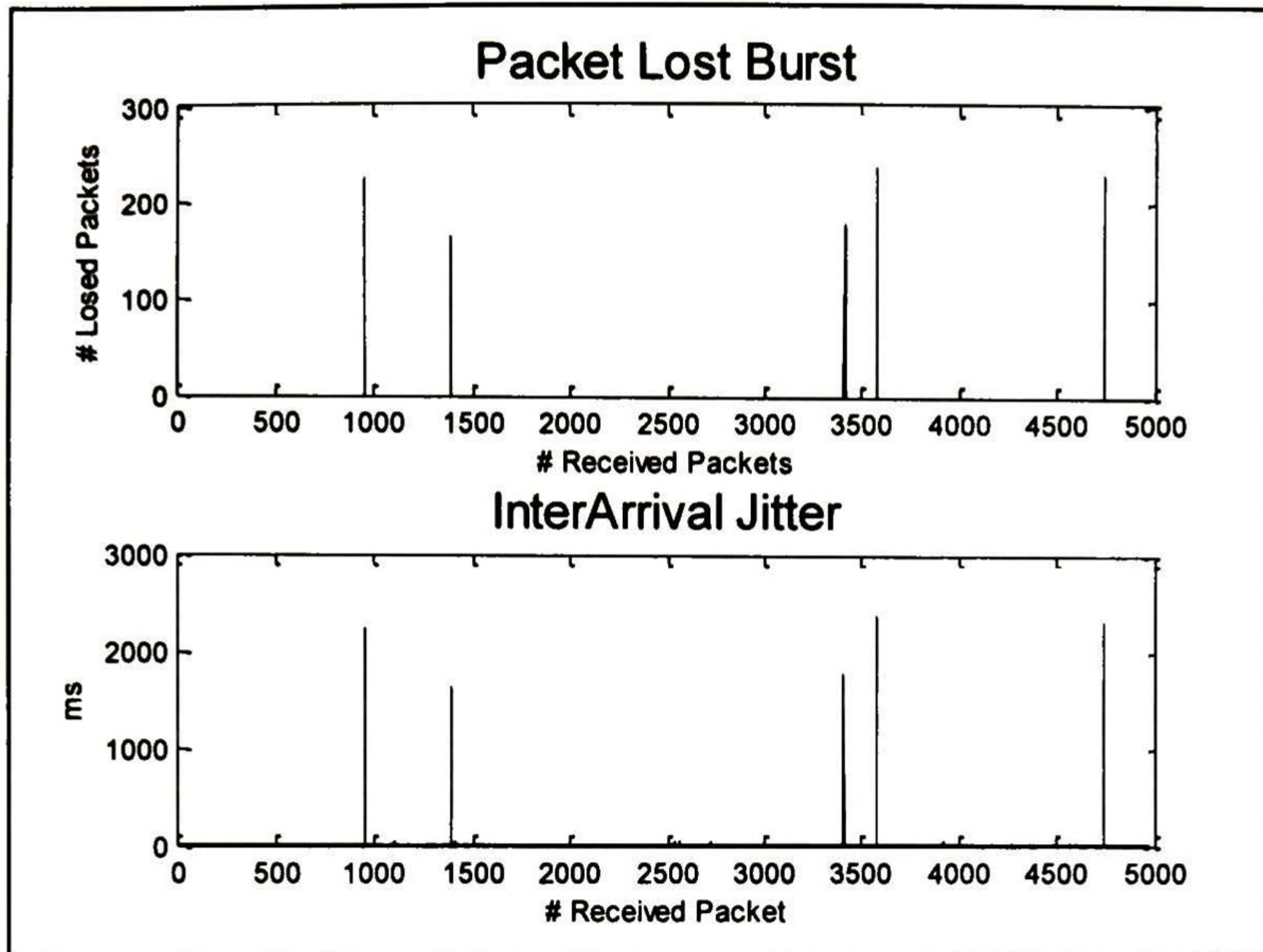


Figura 6-5 G711 A Law 10 ms

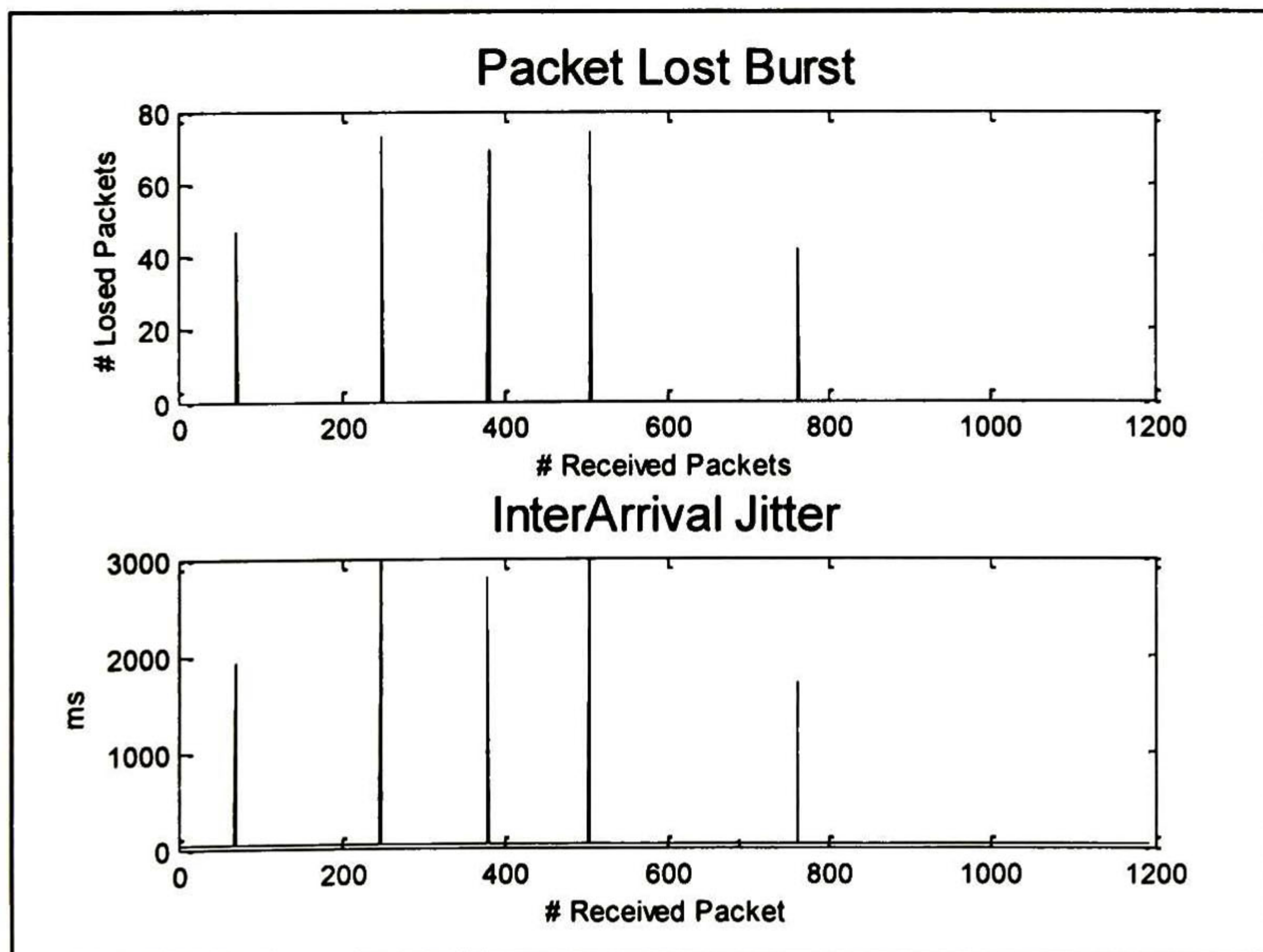


Figura 6-6 G711 A Law 40 ms

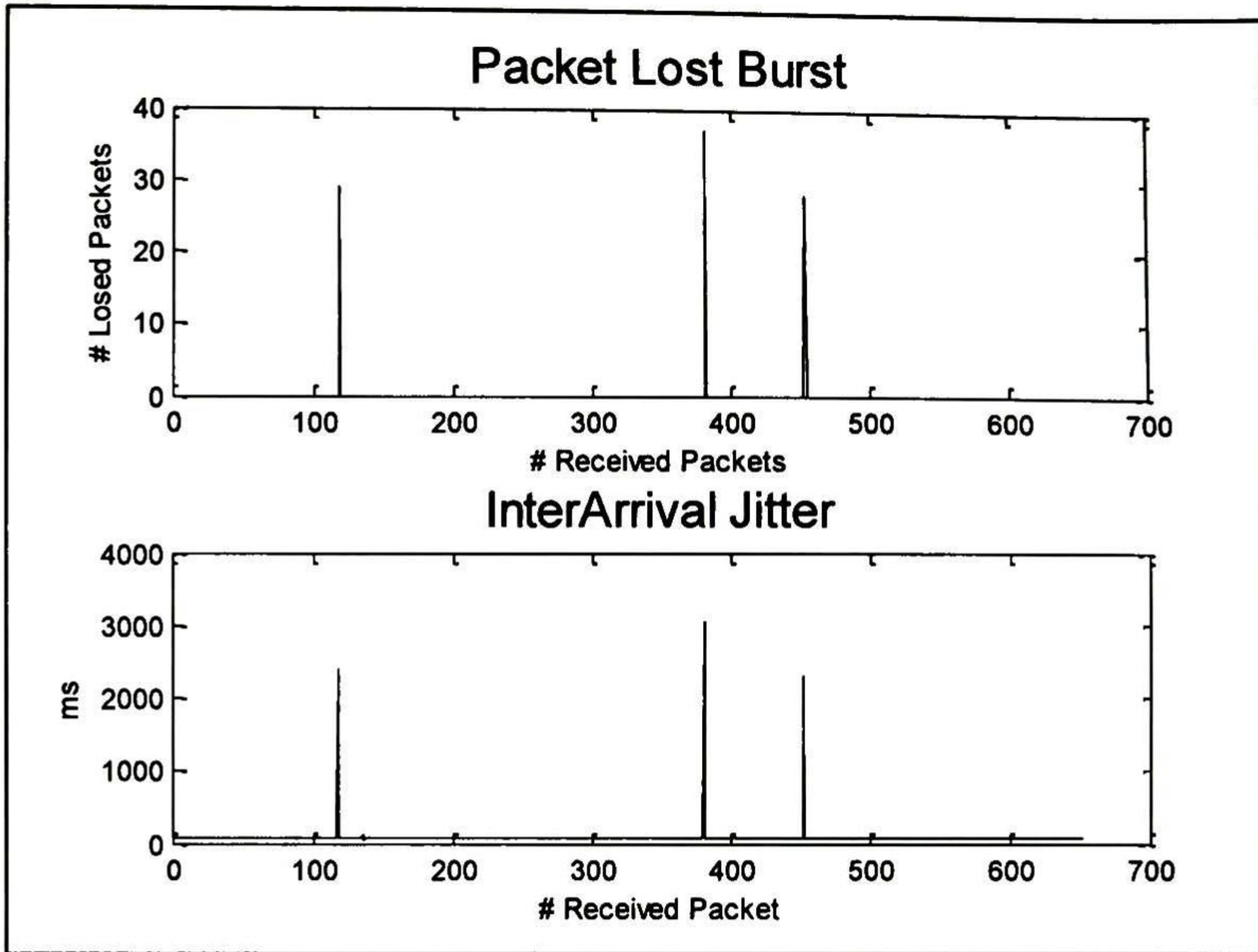


Figura 6-7 G711 A Law 80 ms

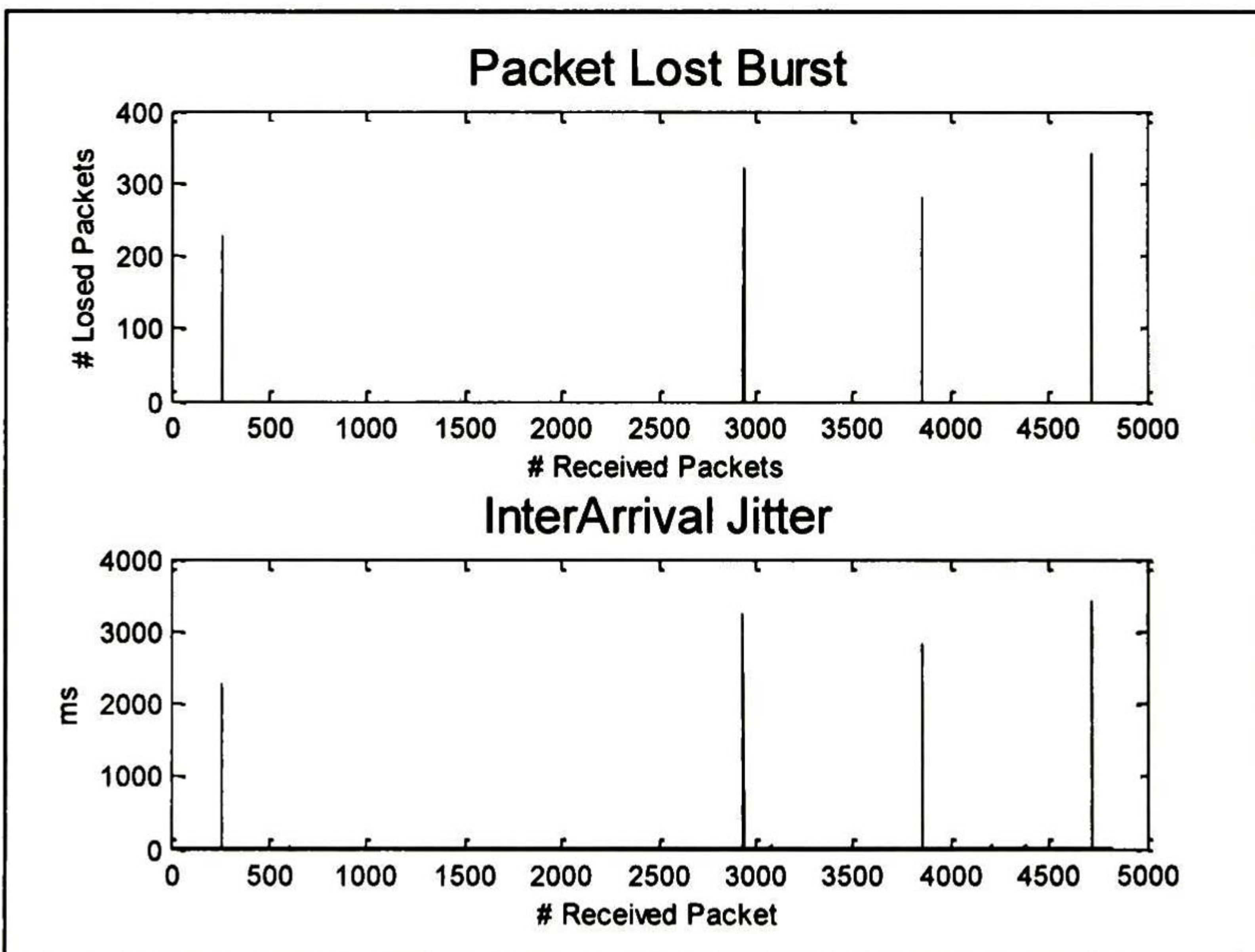


Figura 6-8 G711 Mu Law 10 ms

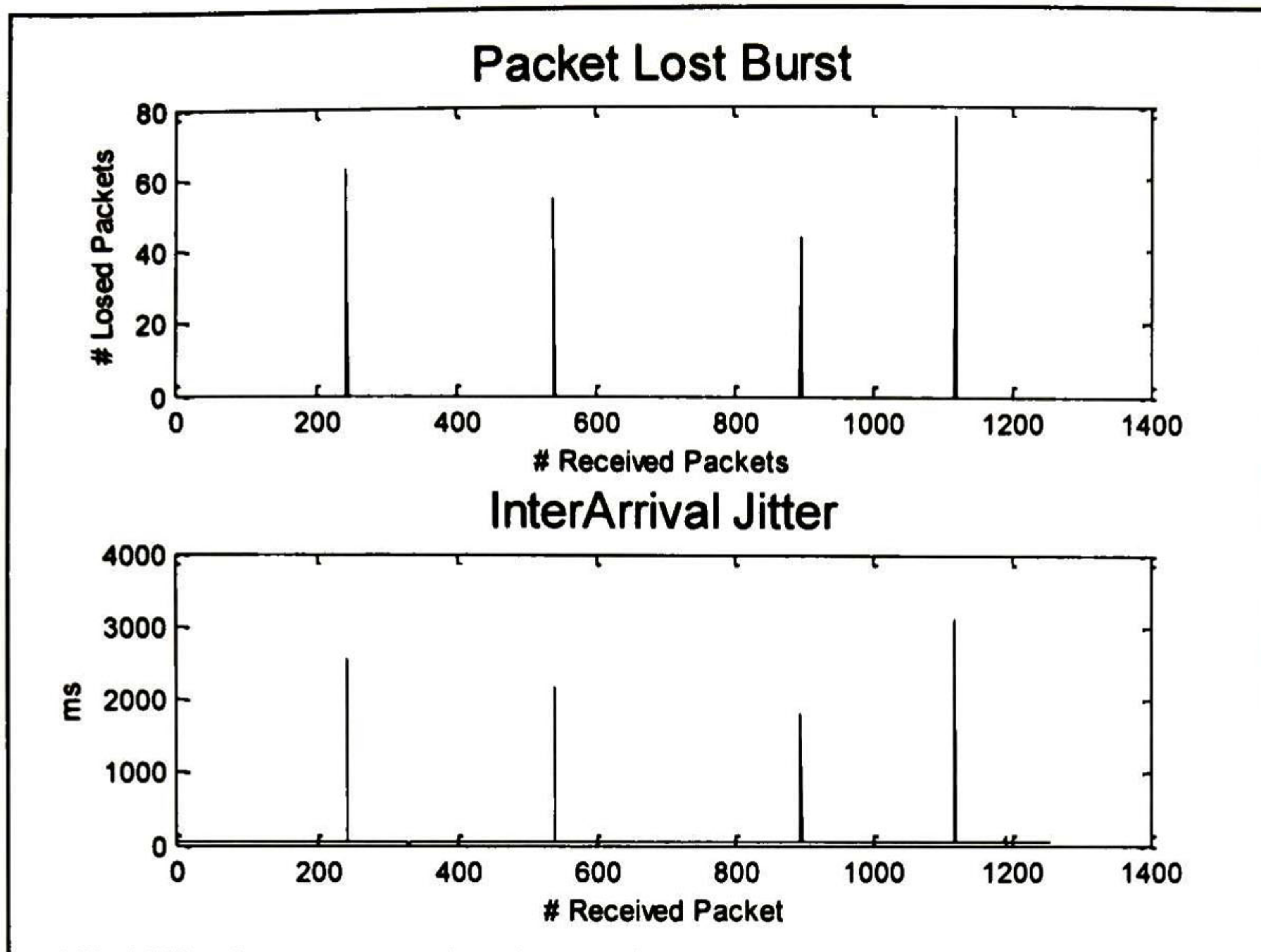


Figura 6-9 G711 Mu Law 40 ms

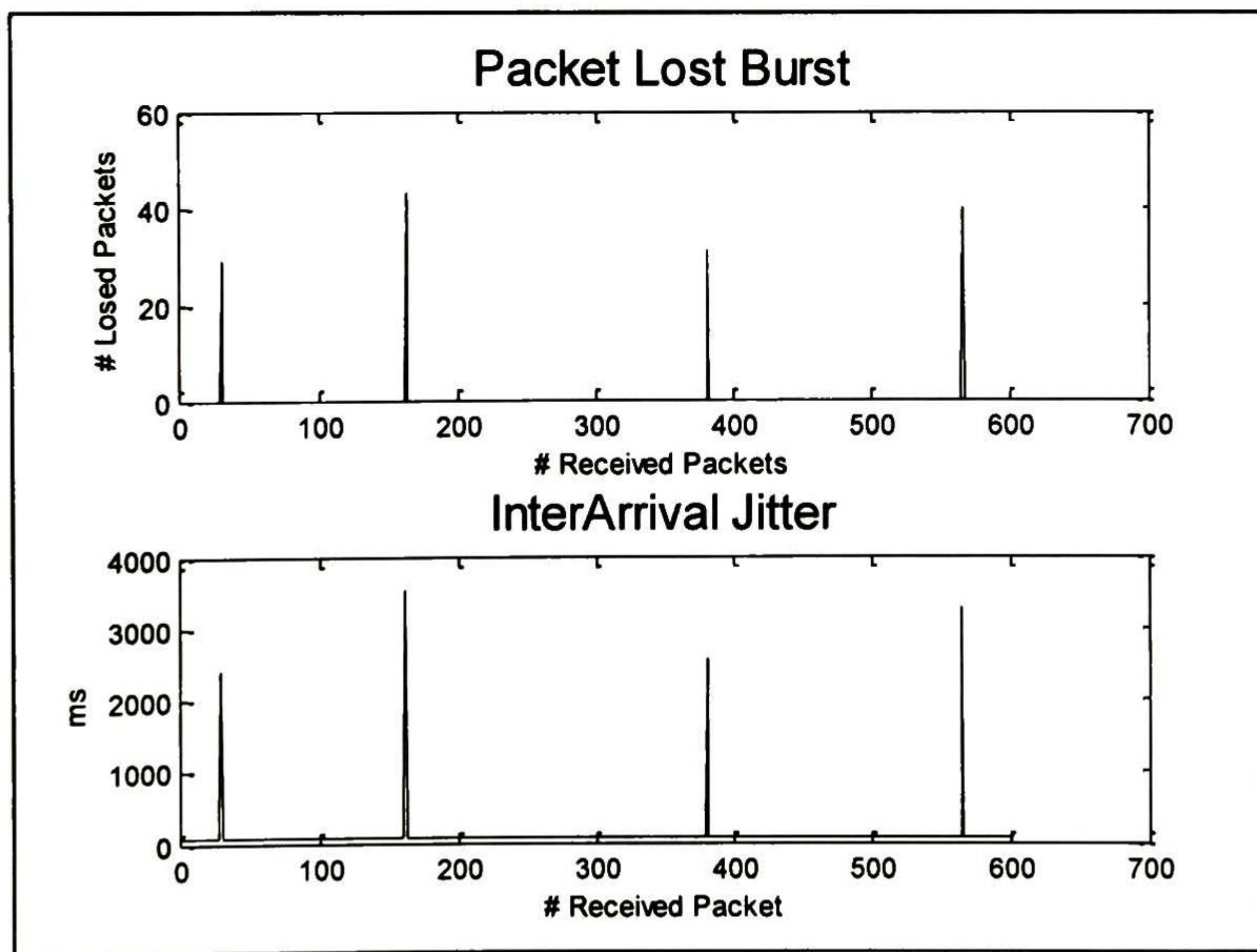


Figura 6-10 G711 Mu Law 80 ms

Observando las graficas anteriores se llega a la conclusión que cumple con lo establecido. Ya que todas las graficas del tiempo de de inter arribo son similares a las de Ráfagas de Pérdidas de Paquetes escaladas por el tamaño de paquete. En las gráficas de G711 A Law de 10ms y G711 Mu Law de 10ms es donde se puede observar mejor lo antes mencionado.

Con estos nuevos resultados y los mencionados en la primera prueba podemos concluir que el sistema satisface la prueba de resolución de tiempo y el envió periódico de paquetes correctamente.

7 CONCLUSIÓN Y TRABAJO FUTURO

En este trabajo se describió de forma detallada el proceso de desarrollo del sistema, iniciando por la justificación, diseño, desarrollo y pruebas del sistema.

Las características y aportes más importantes son:

- Contar con una herramienta más en el laboratorio de Telecomunicaciones
- Establecer una comunicación de VoIP
- Realizar mediciones de los parámetros de QoS
 - OWD
 - Jitter OWD
 - Valor Absoluto del Jitter de OWD
 - Promedio del Valor Absoluto del Jitter de OWD
- MOS
 - MOS WOD
 - MOS RTT
- Parámetro de Hurts
 - H de OWD
 - H de RTT
 - H del Jitter de OWD
 - H del Jitter de RTT
- Graficar Resultados
- Guardar Trazas de los valores medidos
- Mediciones Automáticas
- Ahorro de tiempo en mediciones y procesamiento
- Ahorro de esfuerzo humano
- Contar con una plataforma para desarrollar un sistema más completo

Se ha logrado desarrollar un sistema completamente funcional como puede apreciarse, pero como primera versión de todo sistema se pueden llevar a cabo versiones posteriores, es decir, mejorar funcionalidades y agregar funcionalidades. Las principales propuestas de trabajo futuro se listan a continuación:

- Implementar nuevos codecs
- Implementar la medición de nuevas métricas
- Desarrollar una versión para Linux
- Realizar una serie de mediciones y un análisis detallado.

- Agregarle la funcionalidad de multiconferencia
- Añadirle inteligencia, que en base a los parámetros elija el códec y tamaño de paquete para mejorar la calidad de servicio
- Probar diferentes modelados que se han realizado

Estas son unos trabajos futuros que se proponen, sin embargo hay infinidad de posibilidades para implementar y explotar este sistema.

8 BIBLIOGRAFÍA

- [1] Xianglin, Wang; C.-C. Jay Kuo (May 1998). "An 800 bps VQ-based LPC voice coder" *The Journal of the Acoustical Society of America* **103**. Retrieved on 24-03-2007.
- [2] L.R. Rabiner, R.W. Schafer, "Digital Processing of Speech Signals", Prentice-Hall, ISBN 0-13-213603-1, 1978.
- [3] W. C. Chu, "Speech Coding Algorithms: Foundation and Evolution of Standardized Coders", John Wiley & Sons, ISBN 0-471-37312-5, 2003.
- [4] B. Atal , M. Schroeder, "Code-excited linear prediction (CELP): High quality speech at very low bit rates", ICASSP'85, Vol. 10, pp.937-940, Apr 1985.
- [5] J.P. Adoul, C. Lamblin, "A Comparison of Some Algebraic Structures for CELP Coding of Speech", IEEE ICASSP'87, pp. 1953-1956, 1987.
- [6] J.P. Adoul, P. Mabillean, M. Deprat, S. Morissette, "Fast CELP Coding Based on Algebraic Codes", IEEE ICASSP'87, pp. 1957-1960, 1987.
- [7] A. S. Tanenbaum, "Redes de Computadoras", 3a ed., Prentice Hall, 1997.
- [8] J. B. Postel, RFC 791, "Internet Protocol: DARPA Internet Program Protocol Specification", IETF, Sep 1981.
- [9] J. Postel, "TCP: Transmission Control Protocol" RFC 793, Septiembre 1981.
- [10] J. Postel, "UDP: User Datagram Protocol", RFC 768, Agosto 1980.
- [11] V. Jacobsen, R. Fredrick, S. Casner, and H. Schulzrinne, "RTP: A transport Protocol for Real-Time Applications", RFC 1889, Enero 1996.
- [12] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, RFC 3550, "RTP: A Transport Protocol for Real-Time Applications", IETF, Jul 2003.
- [13] ITU-T Recommendation H.323, "Packet-Based Multimedia Communications Systems", Julio 2003.
- [14] D. Rizzetto, and C. Catania, "A Voice over IP Service Architecture for Integrated Communications", IEEE INTERNET Computing, Mayo-Junio 1999, pp. 53-62.
- [15] Rec. H.323, "Packet-based multimedia communications systems", ITU-T, Jul 2006.
- [16] H. Toral and D. Torres, "IP Telephony: An Overview", Acapulco, Guerrero, Septiembre 2004, Memorias de CIE 2004, pp. 23-28.

- [17] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, RFC 3261, "SIP: Session Initiation Protocol", IETF, Jun 2002.
- [18] Rec. Y.1540, "IP Packet Transfer and Availability Performance Parameters", ITUT, Dec 2002.
- [19] ITU-T Recommendation G.114, "One-Way Transmission Time," Febrero 1996.
- [20] P. Goyal, A. Greenberg, C. R. Kalmanek, W. T. Marshall, P. Mishra, D. Nortz, and K. K. Ramakrishnan, "Integration of Call Signaling and Resource Management for IP Telephony," IEEE INTERNET Computing, Mayo-Junio 1999, pp. 44-52.
- [21] B. Irineo, D. Torres, A. Veloz and E. García, "'In Band" FEC Encoder for SONET/SDH At 2.5 Gbit/s And 10 Gbit/s", México, D.F., Septiembre 2003, CIE 2003.
- [22] B. Bouallegue, R. Djemal, H. Guesmi, R. Tourki and J. P. Diguët, "A Flow Control Approach and Interleaving Method for Real-time Application in High-speed Network," Dedicated Systems Magazine 2003 (<http://www.dedicated-systems.com>), pp. 13-18.
- [23] R. Prasad and C. Dovrolis, "Bandwidth Estimation: Metrics, Measurement Techniques, and Tools," IEEE Network, Noviembre-Diciembre 2003, pp. 27-35.
- [24] Rec. G.107, "The E-model, a computational model for use in transmission planning", ITU-T, Mar 2005.
- [25] Rec. P.800, "Methods for subjective determination of transmission quality", ITUT, Aug 1996.
- [26] Steve McQuerry, Kelly McGrew, Stephen Foy "CISCO voice over frame relay ATM and IP", CISCO Press, 2001
- [27] COLE R. G., ROSENBLUTH J.H., "Voice over IP performance monitoring" AT&T Laboratories, Miiddletown, NJ.
- [28] R. G. Clegg, "Statistics of Dynamic Networks"; PhD thesis, Dept. of Math., Uni.of York, 2004.
- [29] M. S. Taqqu and V. Teverovsky, "On Estimating the Intensity of Long-Range Dependence in Finite and Infinite Variance Time Series"; A Practical Guide to Heavy Tails: Statistical Techniques and Applications, Birkhauser, Boston, 1998, p.p. 177-217.
- [30] M. S. Taqqu, V. Teverovsky and W. Willinger, "Estimators for Long-range Dependence: An Empirical Study"; Fractals, 1995, Volume 3, No. 4, p.p. 785-788.

- [31] S. Bregni and L. Primerano, "Using the Modified Allan Variance for Accurate Estimation of the Hurst Parameter of Long-Range Dependent Traffic"; IEEE Transactions on Information Theory, 2005.
- [32] M. S. Taqqu and V. Teverovsky, "Robustness of Whittle-type estimates for time series with long - range dependence"; Stochastic Models, 1997, Volume 13, p.p. 723-757.
- [33] P. M. Robinson, "Gaussian semiparametric estimation of long-range dependence"; The Annals of Statistics, 1995, Volume 23, p.p. 1630-1661.
- [34] M.S., Taqqu, and V. Teverovsky, "Semi-Parametric Graphical Estimation Techniques for Long-Memory Data"; Time Series Analysis in Memory of E.J. Hannan, New York, Springer-Verlag, 1996, p.p. 420-432.
- [35] O. Rincon y D. Torres, "Software Avanzado para Análisis de Métricas de Internet con Aplicaciones a Series de Tiempo", Tesis de Maestría, CINVESTAV Unidad Guadalajara, Dic 2006.
- [36] H. Toral y D. Torres, "Mediciones de las métricas de tráfico de vos en redes IP", Tesis de Maestría, CINVESTAV Unidad Guadalajara, Dic 2006.
- [37] Ian Somerville "Ingeniería de Software", 6ta Edición, Pearson Educación, México, 2002

9 ANEXO A – DOCUMENTACIÓN DEL CÓDIGO

Este software fue desarrollado, siguiendo un proceso de desarrollo de software y aplicando diversos conocimientos de ingeniería de software. Una etapa fundamental de todo proceso es la documentación, además se pretende que este sistema pueda servir como plataforma para desarrollar un sistema más completo y potente.

Como ya se ha mencionado en la sección 5.6 del capítulo 5, la documentación se generó con la herramienta llama **Doxygen**. La cual es un sistema muy completo y de uso libre. Sin embargo para lograr una buena documentación es necesario usar toda las características de este sistema de documentación.

Como resultado la documentación generada es muy completa pero muy extensa por lo que no fue posible incluirla en este documento, pero sin embargo, será añadida a este en formato electrónico grabada en un medio de almacenamiento digital para su consulta.

10 ANEXO B – PLAN DE PRUEBAS

1. Identificador

PP-VoIPAS-01

2.- Introducción

2.1 Objetivos

El plan de pruebas para el proyecto VoIPAS deberá cubrir los siguientes objetivos:

- Detallar las actividades requeridas para preparar y llevar a cabo las pruebas de operación del sistema.
- Establecer la secuencia y planificación de las tareas de prueba.
- Establecer las fuentes de información necesarias para preparar el plan.
- Determinar las herramientas necesarias para realizar las pruebas.
- Definir el entorno necesario para llevar a cabo las pruebas.

2.2 Justificación

El sistema VoIPAS es desarrollado como una herramienta más para el laboratorio de Telecomunicaciones, también se pretende que sirva como una plataforma de desarrollo en la cual se puedan implementar nuevas funcionalidades.

Por ello es importante realizar unas series de pruebas que nos permitan identificar anomalías en el sistema para ser corregidas y obtener un sistema estable, el cual pueda evolucionar de manera satisfactoria.

2.3 Ámbito

Este plan de pruebas propone la verificación de las operaciones individuales del sistema VoIPAS utilizando la metodología de caja negra (evaluación del sistema a través de entradas y resultados) combinada con pruebas exploratorias para los módulos de menor prioridad.

Para el enfoque de caja negra, los resultados de las operaciones se evaluarán utilizando diferentes técnicas como la comparación con otras herramientas,

replicación de resultados provenientes de la literatura y algunas propiedades de los procesos; todo esto con el objetivo de asegurar que el producto de software que se desarrolló es robusto, proporciona resultados confiables y lo hace dentro del marco de los requerimientos del sistema.

2.4 Referencias

En este caso los documentos a los que hará referencia y que en un momento dado fungirán como fuente de información para el desarrollo tanto del plan como de las pruebas mismas son los siguientes:

- Capítulo 4 “Especificación de Requerimientos”, de esta Tesis.
- Estándar para la documentación de las pruebas de software IEEE 829 – 1998

3.- Productos de prueba

Debido a que las pruebas que se llevarán a cabo son sólo de caja negra, el único producto de prueba que será verificado es el archivo ejecutable que representa al sistema, el cual deberá ser corregido conforme se vayan hallando errores, de manera que el módulo que se verifique siempre corresponda a la última versión.

4.- Características que serán probadas

A continuación se listan las características del sistema que deberán ser probadas:

- 1. Señalización entre aplicaciones.*
- 2. Transmisión y recepción de audio.*
- 3. Resolución de tiempo.*
- 4. Realización de mediciones de forma automática y el guardado de las trazas de las métricas calculadas.*
- 5. Cálculo correcto del MOS.*

5.- Características que no serán probadas

Debido al tipo de pruebas por realizarse, el código del sistema no será probado en ningún caso. Así mismo, como se reutilizó algunas clases, especificadas en la sección 5.3 de este trabajo, no se generarán casos de prueba para verificar el funcionamiento de estos. Ya que las clases reutilizadas, fueron probadas y verificadas [35].

Debe hacerse la aclaración que tampoco se harán pruebas exhaustivas del sistema sino que se escogerán casos representativos de cada situación que permitan evaluar la mayor cantidad de características con el número mínimo de casos de prueba.

6.- Estrategia

Debido a que los módulos cubren características muy específicas y de alguna manera, independientes entre sí, las pruebas serán realizadas de acuerdo al siguiente modelo:

- ***Pruebas de Comunicación.***
Las pruebas de comunicación van encaminadas a asegurar la correcta captura, transmisión y recepción de audio.
- ***Pruebas de Cálculo y Guardado de parámetros.***
Pretende probar que todos los parámetros sean calculados correctamente por cada paquete recibido.
- ***Pruebas de Resolución de Tiempo.***
Un criterio más riguroso se basa en que se cumplan la transmisión periódica de los paquetes con la resolución de tiempo adecuada. Con la resolución de tiempo adecuada nos referimos al tamaño de paquete elegido para transmisión y el codec elegido.
- ***Pruebas de automatización de las mediciones.***
Tiene como objetivo principal verificar que una vez configurado un conjunto de mediciones estas se realicen continuamente de manera automática, sin errores.
- ***Pruebas de MOS.***
Las pruebas de comunicación van encaminadas a asegurar la correcta captura, transmisión y recepción de audio.

6.1 Regresión

Cuando exista una nueva versión que probar debido a la corrección de algún incidente en pruebas anteriores, se deberá realizar una prueba de regresión a las operaciones afectadas por la corrección, es decir, repetir las pruebas adecuadas que ya hubieran sido superadas por la versión anterior, con el objetivo de verificar que no hayan sido introducidos nuevos errores durante la corrección del proceso.

7.- Criterio de Paso / Fallo de los productos

Como el sistema se compone de pocos módulos los criterios de paso / fallo serán estrictos para asegurar la estabilidad y el adecuado comportamiento del producto ante todos los casos que se le impongan.

Por ello, un módulo no se considerará exitoso sino hasta que sea capaz de manejar todas las posibles entradas a las que sea sometido y proveer una respuesta adecuada a cada una de ellas.

Las excepciones a este criterio se evaluarán de acuerdo a las necesidades, considerando las restricciones de tiempo y la posibilidad de depuración de los errores.

8.- Criterio de suspensión y requerimientos de reanudación

Criterio de suspensión: A menos que se halle un error que requiera atención inmediata y que impida continuar con las pruebas del resto del sistema, las actividades de verificación no se verán suspendidas por ninguna causa.

Criterio de reanudación: Retomando la premisa anterior, en el momento en que se haya liberado la versión con el error corregido se reanudarán las actividades de prueba empezando con la correspondiente prueba de regresión.

9.- Documentos de prueba que se entregarán

Los siguientes documentos serán generados durante la fase de verificación:

- Plan de pruebas del sistema.
- Especificación de operaciones del sistema.
- Especificación de diseño de pruebas.
- Reportes de ejecución de pruebas.

10.- Tareas de pruebas

Ver planeación.

11.- Necesidades del entorno

Las pruebas se realizarán en una máquina con las siguientes características:

- Computadora Portátil Acer Aspire 5100.
- Memoria (RAM) de 1278 MB.
- Procesador AMD Turion(tm) 64 X2 Mobile Technology TL-50 1.60 GHz.
- Memoria de Video Dedicada ATI Radeon de 256 MB.
- Sistema Operativo de 32 bits Windows Vista Ultimate.

12.- Planificación

Id	Nombre de tarea	Duración	21 jun '09							28 jun '09										
			D	L	M	X	J	V	S	D	L	M	X	J	V	S				
1	Preparar el plan de pruebas	1 día																		
2	Preparar las especificaciones del diseño de pruebas	1 día																		
3	Ejecutar el procedimiento de las pruebas	1 día																		
4	Revisar los resultados de la ejecución de las pruebas	1 día																		
5	Acordar los cambios necesarios de acuerdo a los resultados	1 día																		
6	Realizar los cambios necesarios en el sistema	2 días																		
7	Repetir los pasos 3, 4, 5 y6. Hasta cumplir todas la pruebas	2 días																		
8	Generar el reporte de "Resumen de Pruebas"	1 día																		

13.- Responsabilidades en la organización y realización de las pruebas.

Pruebas de software: Jesús Antonio Argáez Xool

14.- Aprobaciones

Ing. Jesús Antonio Argáez Xool

Desarrollador del Proyecto

Dr. Deni Librado Torres Román

Asesor del Proyecto

11 ANEXO C – ESPECIFICACIÓN DE OPERACIONES

ID	OP-VOIPAS-CONFIG-01
Nombre	Configuration
Descripción	<p>Permite configurar los parámetros generales del sistema como son:</p> <ul style="list-style-type: none">• My User Name Nombre con el que la aplicación cliente será identificado durante una llamada.• Server IP Address Dirección IP del servidor el cual registra los usuarios que se encuentren disponibles.• Control Port Número de puerto a través del cual las aplicaciones se comunicaran, mediante señales de control.• Server Response Time: Tiempo, en segundos, en el que se espera respuesta del servidor cuando se solicite la dirección IP de algún usuario con el cual deseemos comunicarnos, pasado este tiempo se tomará como una incomunicación con el servidor.• Update Frequency: Tiempo, en segundos, en que los datos de las estadísticas se actualizaran visualmente en la tabla Measurement de la ventana principal de la aplicación.• Output Directory Es la ruta en la cual se guardaran los resultados cuando se realicen las mediciones automáticas.• Automatic Reply Es una opción que habilitada, la aplicación responderá de manera automática a la petición de comunicación de un usuario, de lo contrario mostrara una alerta de que alguien se intenta comunicar con nosotros, pudiendo aceptarla o rechazarla.

ID	OP-VOIPAS-SESSION-01
Nombre	New Session
Descripción	Crea una nueva sesión de comunicación de VoIP, permitiéndonos elegir el usuario con el que nos deseemos comunicar, el número de puerto de la comunicación, el tipo y tamaño de codec, el tamaño de la ventana y el método para el cálculo del parámetro de Hurst y el modo de captura de los paquetes.

ID	OP-VOIPAS-Start-01
Nombre	Solicitud
Descripción	Inicia el intento de comunicación enviando una solicitud al usuario y esperando su respuesta.

ID	OP-VOIPAS-Start-02
Nombre	Automatic Response
Descripción	El usuario destino cuando reciba una solicitud este deberá responder de manera automática, si la opción Automatic Reply esta activada.

ID	OP-VOIPAS-Start-03
Nombre	Manual Response
Descripción	Se mostrara una alerta para confirma si se acepta o rechaza la comunicación, si la opción Automatic Reply no esta activada

ID	OP-VOIPAS-Comunication-01
Nombre	Audio Transmission
Descripción	La aplicación captura paquetes de audio, en base al tamaño de paquete y codec elegido, y los transmitirá.

ID	OP-VOIPAS-Communication-02
Nombre	Audio Reception
Descripción	La aplicación cada vez que reciba un paquete de audio, extrae la parte de datos, lo decodifica y reproduce el audio adecuadamente.

ID	OP-VOIPAS-Communication-03
Nombre	Statistics
Descripción	La aplicación cada vez que reciba un paquete de audio extrae la cabecera del paquete, calcula y estimá los parámetros correctamente.

ID	OP-VOIPAS-Measurement-01
Nombre	Manual Measurement
Descripción	Cuando se inicie una medición de modo manual el usuario es el encargado de terminarla.

ID	OP-VOIPAS-Measurement-02
Nombre	Measurement by Minutes
Descripción	En este tipo de medición el sistema la terminar una vez que haya transcurrido el número de minutos que se haya configurado.

ID	OP-VOIPAS-Measurement-03
Nombre	Measurement by Packets
Descripción	El sistema termina la medición una vez que se hayan capturado el número de paquetes que se hayan configurado en la medición.

ID	OP-VOIPAS-MOS-01
Nombre	MOS
Descripción	El sistema en base a los parámetros calculados calificara la QoS de servicio a través del Modelo E y el MOS.

ID	OP-VOIPAS-Stop-01
Nombre	Stop
Descripción	Cuando el botón Stop sea activado el sistema detiene la medición en curso sea esta de tipo manual, por paquetes o por minutos y no guardara los resultados, será el usuario la persona encargada de hacerlo.

ID	OP-VOIPAS-Save-01
Nombre	Autosave from Parameters
Descripción	Cuando se termine una medición cuya forma de captura sea por paquetes o por minuto, el sistema guarda de forma automática las trazas de los parámetros en el directorio de salida previamente configurado.

ID	OP-VOIPAS-Save-02
Nombre	Saved Manual of Parameters
Descripción	Cuando una medición de forma manual termine, el sistema no guarada los datos, el usuario será encargado de guardar los datos si así lo desea.

ID	OP-VOIPAS-Graph-01
Nombre	Graphic of Parameters
Descripción	El sistema grafica los parámetros que se seleccionen correctamente.

12 ANEXO D – ESPECIFICACIÓN DE DISEÑO DE PRUEBAS

1. Identificador de la especificación del diseño de pruebas.

EDP-VoIPAS-01

2. Características a Probar.

Los módulos principales del sistema son:

- a. El módulo de Transmisión y recepción de datos.
- b. El módulo de cálculo y estimación de parámetros.

Es por ello que las pruebas se enfocaran a estos módulos para obtener un funcionamiento óptimo del sistema.

3. Refinamiento del enfoque.

La generación del conjunto de pruebas se basara en los criterios de:

- Cobertura de la especificación de operaciones.

4. Identificación de las pruebas.

ID de Operación	OP-VOIPAS-CONFIG-01
ID de Caso de Prueba	CP-OP-VOIPAS-CONFIG-01
Método de Evaluación	Archivo de configuración VoIPAS.ini.
Descripción	Cada vez que se configuren los parámetros del sistema se deben actualizar los valores en el archivo VoIPAS.ini
Resultado	Los parámetros del archivo VoIPAS.ini deben corresponder con los ingresados en la configuración realizada.

ID de Operación	OP-VOIPAS-SESSION-01
ID de Caso de Prueba	CP-OP-VOIPAS-SESSION-01
Método de Evaluación	Tabla de sesiones de la ventana principal.
Descripción	Cuando se configure una sesión esta debe ser agregada a la tabla de sesiones con los valores que se hayan elegido.
Resultado	Los valores de la sesión en la tabla de sesiones deben ser los mismos que elegidos durante la creación de una nueva sesión.

ID de Operación	OP-VOIPAS-Start-01
ID de Caso de Prueba	CP-OP-VOIPAS-Start-01
Método de Evaluación	Inicio de una sesión.
Descripción	Se configurara una sesión con una dirección IP destino determinada y se iniciara.
Resultado	La aplicación destino debe de recibir la solicitud de comunicación.

ID de Operación	OP-VOIPAS-Start-02
ID de Caso de Prueba	CP-OP-VOIPAS-Start-02
Método de Evaluación	Inicio de una sesión.
Descripción	Se configurara una sesión con una dirección IP destino determinada, la opción auto reply activa y se iniciara.
Resultado	La aplicación destino debe de recibir la solicitud de comunicación y aceptar automáticamente la solicitud de comunicación si mostrar ningún mensaje.

ID de Operación	OP-VOIPAS-Start-03
ID de Caso de Prueba	CP-OP-VOIPAS-Start-03
Método de Evaluación	Inicio de una sesión.
Descripción	Se configurara una sesión con una dirección IP destino determinada, la opción auto reply desactivada y se iniciara.
Resultado	La aplicación destino debe de recibir la solicitud de comunicación y mostrar un mensaje, sin iniciar la comunicación al menos que sea aceptada por el usuario.

ID de Operación	OP-VOIPAS-Communication-01
ID de Caso de Prueba	CP-OP-VOIPAS-Communication-01
Método de Evaluación	Realizar una medición.
Descripción	Se configurara una sesión y se iniciara.
Resultado	La aplicación transmisora debe de capturar paquetes, de acorde al tamaño de paquete y codec elegido, y trasmitirlos; el receptor debe reproducir el audio capturado.

ID de Operación	OP-VOIPAS-Communication-02
ID de Caso de Prueba	CP-OP-VOIPAS-Communication-02
Método de Evaluación	Realizar una medición.
Descripción	Se configurara una sesión y se iniciara.
Resultado	La aplicación receptora debe recibir los paquetes enviados por el receptor y reproducirlos, estos deben corresponder al audio trasmitido.

ID de Operación	OP-VOIPAS-Communication-03
ID de Caso de Prueba	CP-OP-VOIPAS-Communication-03
Método de Evaluación	Realizar una medición.
Descripción	Cuando se reciba un paquete se debe generar el cálculo de los parámetros. La estimación del parámetro de Hurst se realizara si la ventana que se configuro está llena.
Resultado	Las longitudes de las trazas de los parámetros calculados y estimados deben corresponder de acuerdo a el número de paquetes transmitidos.

ID de Operación	OP-VOIPAS-Measurement-01
ID de Caso de Prueba	CP-OP-VOIPAS-Measurement-01
Método de Evaluación	Realizar una medición.
Descripción	Se creara una sesión cuyo modo de captura será manual.
Resultado	La medición no se detendrá por algún número de paquete capturado o tiempo transcurrido, sino hasta que el usuario a detenga, y las trazas no se guardaran al menos que el usuario lo elija.

ID de Operación	OP-VOIPAS-Measurement-02
ID de Caso de Prueba	CP-OP-VOIPAS-Measurement-02
Método de Evaluación	Realizar una medición.
Descripción	Se creara una sesión con modo de captura por minutos.
Resultado	La medición se detendrá una vez transcurrido el número de segundos con el cual se haya configurado la sesión.

ID de Operación	OP-VOIPAS-Measurement-03
ID de Caso de Prueba	CP-OP-VOIPAS-Measurement-03
Método de Evaluación	Realizar una medición.
Descripción	Se creara una sesión con modo de captura por paquetes.
Resultado	La medición se detendrá una vez capturado el número de paquetes que se haya configurado en la sesión.

ID de Operación	OP-VOIPAS-MOS-01
ID de Caso de Prueba	CP-OP-VOIPAS-MOS-01
Método de Evaluación	Realizar una medición.
Descripción	Se realizara una medición. Durante la medición se verá la calidad de la comunicación y se simularan perdidas de paquetes.
Resultado	La calificación del MOS debe ser du acuerdo a lo esperado con poco retraso y sin perdidas se espera que la calificación sea buena; cuando se pierda paquetes la calificación del MOS se debe de degradar.

ID de Operación	OP-VOIPAS-Stop-01
ID de Caso de Prueba	CP-OP-VOIPAS-Stop-01
Método de Evaluación	Realizar una medición.
Descripción	Detener la comunicación mediante el botón Stop.
Resultado	La comunicación se detendrá sin importar el tipo de medición que se haya elegido.

ID de Operación	OP-VOIPAS-Save-01
ID de Caso de Prueba	CP-OP-VOIPAS-Save-01
Método de Evaluación	Realizar una medición automática, es decir, por minutos y por paquetes.
Descripción	La aplicación se detendrá una vez capturado el numero de paquetes elegido o una vez transcurrido el tiempo determinado en la sesión.
Resultado	Por cada medición automática debe existir una carpeta con las trazas de los parámetros calculados.

ID de Operación	OP-VOIPAS-Save-02
ID de Caso de Prueba	CP-OP-VOIPAS-Save-02
Método de Evaluación	Realizar una medición manual.
Descripción	Detendremos la aplicación y guardaremos los parámetros calculados que deseemos a través del botón Save Traces.
Resultado	Las trazas de los parámetros que elijamos se deben guardar correctamente.

ID de Operación	OP-VOIPAS-Graph-01
ID de Caso de Prueba	CP-OP-VOIPAS-Graph-01
Método de Evaluación	Realizar una medición manual, por paquetes y por minutos.
Descripción	Detendremos la aplicación y guardaremos los parámetros, para posteriormente grafica graficar los parámetros calculados desde la aplicación y las trazas guardadas en Matlab y compararemos las gráficas.
Resultado	La graficas generadas en la aplicación y en Matlab deben ser muy semejantes.

13 ANEXO E – REPORTE DE EJECUCIÓN DE PRUEBAS

Reporte de Ejecución de Pruebas		
Sistema: VoIPAS 1.0	ID: EDP-VoIPAS-01	
Elaborado por: Jesús Antonio Argáez Xool	Fecha: 28/06/2009	
Id de caso de prueba	Resultado	Comentarios
CP-OP-VOIPAS-CONFIG-01	Aprobada	
CP-OP-VOIPAS-SESSION-01	Aprobada	
CP-OP-VOIPAS-Start-01	Aprobada	
CP-OP-VOIPAS-Start-02	Aprobada	
CP-OP-VOIPAS-Start-03	Aprobada	
CP-OP-VOIPAS-Communication-01	Aprobada	
CP-OP-VOIPAS-Communication-02	Aprobada	
CP-OP-VOIPAS-Communication-03	No Aprobada	La cantidad de parámetros calculados no coincide y en la estimación de Hurst marca unos errores.
CP-OP-VOIPAS-Measurement-01	Aprobada	
CP-OP-VOIPAS-Measurement-02	Aprobada	
CP-OP-VOIPAS-Measurement-03	Aprobada	
CP-OP-VOIPAS-MOS-01	No Aprobada	Un factor constante para el MOS se encuentra desactualizado
CP-OP-VOIPAS-Stop-01	Aprobada	
CP-OP-VOIPAS-Save-01	Aprobada	
CP-OP-VOIPAS-Save-02	Aprobada	
CP-OP-VOIPAS-Graph-01	Aprobada	

Reporte de Ejecución de Pruebas

Sistema: VoIPAS 1.0	ID: EDP-VoIPAS-02
Elaborado por: Jesús Antonio Argáez Xool	Fecha: 29/06/2009

Id de caso de prueba	Resultado	Comentarios
CP-OP-VOIPAS-CONFIG-01	Aprobada	
CP-OP-VOIPAS-SESSION-01	Aprobada	
CP-OP-VOIPAS-Start-01	Aprobada	
CP-OP-VOIPAS-Start-02	Aprobada	
CP-OP-VOIPAS-Start-03	Aprobada	
CP-OP-VOIPAS-Communication-01	Aprobada	
CP-OP-VOIPAS-Communication-02	Aprobada	
CP-OP-VOIPAS-Communication-03	Aprobada	
CP-OP-VOIPAS-Measurement-01	Aprobada	
CP-OP-VOIPAS-Measurement-02	Aprobada	
CP-OP-VOIPAS-Measurement-03	Aprobada	
CP-OP-VOIPAS-MOS-01	Aprobada	
CP-OP-VOIPAS-Stop-01	Aprobada	
CP-OP-VOIPAS-Save-01	Aprobada	
CP-OP-VOIPAS-Save-02	Aprobada	
CP-OP-VOIPAS-Graph-01	Aprobada	



**CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL I.P.N.
UNIDAD GUADALAJARA**

El Jurado designado por la Unidad Guadalajara del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional aprobó la tesis

Software para el análisis de QoS en VoIP

del (la) C.

Jesús Antonio ARGÁEZ XOOL

el día 17 de Julio de 2009.

Dr. Deni Librado Torres Román
Investigador CINVESTAV 3A
CINVESTAV Unidad Guadalajara

Dr. Mario Angel Siller González
Pico
Investigador CINVESTAV 2A
CINVESTAV Unidad Guadalajara

Dr. Arturo Veloz Guerrero
Gerente de Diseño Electrónico
Intel

