UNIVERSIDAD DE QUINTANA ROO
DIVISIÓN DE CIENCIAS E INGENIERÍA

Emulación de Pérdida de Paquetes mediante una Aplicación VoIP y Caracterización de Parámetros de QoS

TESIS
Para obtener el grado de
Ingeniero en Redes

PRESENTA
Cinthya Addlemy Osorio Flota

DIRECTOR DE TESIS
Dr. Homero Toral Cruz

ASESORES

Dr. Jaime Silverio Ortegón Aguilar

Dr. Freddy Ignacio Chan Puc

Dr. Gliserio Romeli Barbosa Pool

MSI Laura Yésica Dávalos Castilla

Chetumal Quintana Roo, México, Noviembre de 2015

UNIVERSIDAD DE QUINTANA ROO
DIVISIÓN DE CIENCIAS E INGENIERÍA

Trabajo de Tesis elaborado bajo supervisión del Comité de asesoría y aprobada como requisito parcial para obtener el grado de:

INGENIERO EN REDES

Comité de Trabajo de Tesis

Director: _____
Dr. Homero Toral Cruz

Asesor: _____
Dr. Jaime Silverio Ortegón Aguilar

Asesor: _____
Dr. Freddy Ignacio Chan Puc

UNIVERSIDAD DE QUINTANA ROO
DCI DIVISIÓN DE CIENCIAS E INGENIERÍA

UNIVERSIDAD DE QUINTANA ROO
SERVICIOS ESCOLARES
TITULACIONES

Chetumal Quintana Roo, México, Noviembre de 2015

Agradecimientos

Quiero dar el agradecimiento principal y más importante a mi Dios, El ingeniero por excelencia, por todas las bendiciones que recibí mientras estuve en la universidad, gracias por darme la sabiduría y el valor para terminar este trabajo, por siempre estar en mi camino apoyándome y dándome las fuerzas necesarias para seguir adelante en todos los momentos de mi vida.

Un especial agradecimiento a mis padres Celso Rafael Osorio Cocom y Cinthya Flota Ortiz, por darme siempre amor, apoyo moral, económico y la confianza para alcanzar la culminación de mis estudios, sin duda han sido realmente un pilar fundamental en mi vida.

A las personas que siempre estuvieron creyendo y confiando en mis sueños, mi mami Rafaela Guerra Ortiz y mi hermanita Ayareli Aylin Osorio Flota. Gracias por quererme y ayudarme siempre de una manera muy especial.

A José Alfredo Vazquez Gutierrez, gracias por su apoyo, por su preocupación, por sus bendiciones, por sus palabras que me inspiran aliento cada dia.

A mis profesores por su excelente labor en las aulas, por su esfuerzo diario por enseñarnos siempre lo mejor, gracias porque sin sus enseñanzas no sería posible terminar este trabajo. En especial al Dr. Homero Toral Cruz por brindarme su apoyo incondicional para lograr este objetivo.

A una persona muy especial ALP, gracias por su apoyo incondicional, por su amor, cariño, comprensión, cuidado, y sobre todo por soportar mi mal humor en todo el transcurso de este trabajo y por estar siempre conmigo en todo momento.

A mis compañeros y amigos, gracias por su ayuda incondicional, con quienes tuve la oportunidad de pasar momentos agradables y también momentos difíciles en la carrera.

A todos y cada uno de ustedes mil gracias.

Cinthya Addlemy Osorio Flota

RESUMEN

Con el paso del tiempo, las tecnologías de comunicación han obtenido grandes avances, desde la transmisión de paquetes de datos sobre la red de conmutación de paquetes hasta la transmisión de señales de voz humana en forma de paquetes sobre las redes IP. En nuestros días, a esta tecnología se le conoce como Voz sobre IP (VoIP) y se dice que esta tecnología es unas de las responsables de la convergencia de la red de voz con la red de datos.

Décadas atrás existían dos principales modos de comunicación, el telégrafo y el teléfono, que permitían una simple y rápida comunicación de datos y voz, respectivamente. Estos modos de comunicación evolucionaron con la aparición del Internet, el cual permitió la convergencia de datos y voz hacia el usuario final, dando lugar a la creación de muchos modos de comunicarnos que no sólo cambiaron la vida del usuario, sino también la complejidad de los negocios.

La VoIP, más que una tecnología que permite transmitir voz sobre las redes de datos, permite la convergencia de diferentes medios, tales como: voz, datos y video. Esta aplicación proporciona al usuario muchas ventajas, tales como: reducir costos, permitir disponer de un sistema propio y configurarlo de acuerdo a las necesidades de cada usuario, implementar de manera libre y gratuita otros servicios como el buzón de mensajes, identificación de llamadas, buzón de espera, entre otros.

Para lograr que la voz sobre IP se establezca como una tecnología que pueda competir con la red telefónica pública conmutada (PSTN), debe proporcionar ciertos niveles de calidad de servicio (QoS) exigidos por los usuarios finales.

Teniendo en cuenta que Internet es la red de redes o red convergente, sobre la cual se transmitirán los diferentes medios o servicios de voz, video y datos; surgen ciertos inconvenientes derivados de la calidad de servicio que no es capaz de garantizar Internet, por su naturaleza de recursos compartidos.

La principal desventaja de la tecnología IP es que proporciona un sercvicio de mejor esfuerzo (Best-Effort) y por lo tanto, se presentan retardos (fijos y variables) y pérdida de paquetes que afectan considerablemente la calidad de sevicio de la comunicación. Por tal motivo, el estudió de los principales parámtros de QoS juega un papel importante en la tecnología VoIP.

Diversos estudios han demostrado que la pérdida de paquetes presenta naturaleza rafagoza, es decir, si un paquete se pierde, existe una alta probabilidad que el siguiente paquete también se pierda. A este fenómeno se le conoce como pérdida de paquetes a ráfagas. Un modelo para permite capturar este fenómeno es una cadenas de Markov de estados infinitos.

Tambien, algunos estudios han demostrado que el jitter puede modelarse mediante procesos autosimilares o estructuras de correlación. Estas estructuras de correlación pueden ser representadas mediante un único parámetro, llamado parámetro de Hurst, el cual puede ser estimado por diversos métodos.

En base a los puntos mencionados anteriormente, este proyecto de tesis, fue centrado en el estudio del comportamiento del jitter y la calidad de servicio bajo diferentes niveles de rafagosidad y porcentajes de pérdida de paquetes en un sistema de voz sobre el protocolo de Internet. Para realizar dicho estudio, se desarrolló una metodología práctica para emular diferentes niveles de rafagosidad y porcentajes de pérdida de paquetes sobre una aplicación VoIP en software; se realizó la evaluación de la calidad de servicio mediante el MOS y Modelo E y la evaluación del parámetro de Hurst (H) mediante el método de la varianza en las trazas de jitter del conjunto de llamadas emuladas.

Contenido

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

LISTA DE ABREVIATURAS

A/D	Análigico/Digital
ACF	Advance Custom Fields
ACK	acknowledgement
ACR	Absolute Category Rating
ARQ	Automatic Repeat reQuest
ATM	Asynchronous Transfer Mode
CODEC	Coder-Decoder
CSRC	Contributing sorce
D/A	Digital/Analógico
DSP	Digital Signal Processor
FEC	Forward Error Correction
GK	GateKeeper
HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
IP	Internet Protocol
ISDN	Integrated Services Digital Networks
ITU	International Triathlon Union
LRD	Long Range Dependence
MCU	Multipoint Control Unit
MOS	Mean Opinion Score
OWD	One Way Delay
PLR	Packet Loss Rate
PPP	Point to Point Protocol
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RAS	Registration, Admission and Status
RSVP	Resource Reservation Protocol
RTC	Real Time Clock
RTCP	Real Time Control Protocol
RTP	Real-Time Transport Protocol
SIP	Session Initiation Protocol

SNR	Signal to Noise Ratio
SRD	System Reference Document
SRTP	Secure Real-Time Transport Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UIT	Unión Internacional de las Telecomunicaciones
ITU-T	International Telecommunication Union - Telecommunication Standardization Sector
URI	Uniform Resource Identifier
VoIP	Voice over Internet Protocol
VPN	Virtual Private Network

CAPÍTULO

1

1 INTRODUCCIÓN

Voz sobre el Protocolo de Internet (VoIP) es uno de los servicios más atractivos e importantes hoy en día en las redes de telecomunicaciones. Sin embargo, cuando el tráfico de voz es transportado sobre redes IP, la transmisión basada en paquetes puede introducir desperfectos, como pérdida de paquetes y tener influencia en la calidad de servicio (QoS) percibida por el usuario final. La calidad de la voz de sistemas VoIP depende de muchos parámetros de QoS, particularmente, la tasa pérdida de paquetes (PLR), el jitter y el retardo en un sentido (OWD) tienen un importante impacto en la calidad de la voz. Estos parámetros (PLR, jitter y OWD) están estrechamente relacionados unos con otros y pueden ser usados para configurarlos (longitud de los datos de voz, tipo de CODEC, tamaño de redundancia FEC, y tamaño del de-jitter buffer) a valores óptimos para ofrecer buenos niveles de calidad de voz.

Para alcanzar niveles satisfactorios de calidad de voz, las aplicaciones VoIP deben ser diseñadas en base a modelos de tráfico representativos. Para implementar modelos de tráfico representativos, es necesario el monitoreo de los principales parámetros de QoS, mediante mediciones pasivas y la caracterización de dichos parámetros.

Por otro lado, estudios de tráfico de datos en redes de computadoras modernas han mostrado que: (1) el tráfico IP presenta naturaleza auto-similar y de cola pesada, (2) la pérdida de paquetes es de naturaleza rafagoza y presenta dependencia temporal finita.

Motivados por los puntos anteriores, en este trabajo se generará tráfico de voz con una aplicación VoIP y se emularán diferentes condiciones de pérdidas de paquetes durante la transmisión, haciendo uso de los procesos de Markov; de esta manera se obtendrá un sustancioso conjunto de trazas en el cual se analizará el comportamiento de los principales parámetros de QoS, tales como: PLR y jitter.

1.1 Justificación

Como se mencionó anteriormente, para el diseño adecuado de aplicaciones VoIP, es necesario conocer el comportamiento de los principales parámetros que determinan la QoS que percibe el usuario final. Para caracterizar dichos parámetros se requiere el monitoreo constante de dichos parámetros bajo diversos escenarios de red, lo cual implica los siguientes inconvenientes:

- Utilización de equipo especializado de alto costo.
- Recurso humano/tiempo para implementar el escenario de red bajo estudio.
- Recurso humano/tiempo para el monitoreo de la red.

Sin embargo, mediante modelos representativos de tráfico, se pueden crear algoritmos para emular algunos fenómenos que ocurren en la red, tal como la pérdida de paquetes. De acuerdo a estudios previos, es bien sabido que el proceso de pérdida de paquetes puede ser modelado mediante los procesos de Markov; de esta manera es posible crear diversos escenario de red de manera más práctica y de esta manera poder analizar el comportamiento de los principales parámetros de QoS.

1.2 Objetivos

1.2.1 Objetivo General

Emular procesos de pérdida de paquetes mediante cadenas de Markov y caracterizar los principales parámetros que determinan la calidad de servicio en sistemas de voz sobre el protocolo de Internet.

1.2.2 Objetivos Específicos

1. Generar tráfico de voz mediante una aplicación VoIP, emulando diferentes niveles de pérdida de paquetes y niveles de rafagosidad.
2. Realizar el monitoreo de los principales parámetros de QoS.
3. Analizar los patrones de tráfico capturado.
4. Caracterizar el comportamiento de los parámetros de QoS bajo estudio.

1.3 Metodología

La metodología a seguir se basa en las actividades que se describen a continuación:

1. Primero se realizará un estudio del estado del arte referente al funcionamiento de los sistemas VoIP, los protocolos que intervienen en el proceso de comunicación Voz sobre IP, los elementos de calidad y la integración de comunicaciones de voz a Internet.

2. En esta etapa se hará el mapeo del problema físico (pérdida de paquetes) a un modelo representativo. En un sistema de voz que funciona sobre una red de conmutación de paquetes, como la Internet, la pérdida de paquetes suele ser muy común y esperada. Dichas pérdidas, generalmente ocurren en escalas de tiempo pequeñas y mediante cadenas de Markov de estados finitos es posible representar este fenómeno. Una cadena de Markov de dos estados será utilizada para representar una ráfaga de paquetes perdidos (*burst*s) o una ráfaga de paquetes que llegaron correctamente a su destino (*gaps*). Para lograr un comportamiento similar al que ocurre en Internet, es necesario usar una cadena de Markov de 4 estados. Así podremos tener varias transiciones entre *burst* y *gap*s.

3. Se realizará la caracterización del fenómeno pérdida de paquetes como sigue:

Usaremos una descripción de paquetes perdidos en VoIP basados en ventanas de tiempo angostas y amplias. El comportamiento de los paquetes perdidos en una ventana de tiempo angosta será llamada Microscópica, y el comportamiento de pérdida de paquetes en una ventana de tiempo amplia será llamado Macroscópica.

El comportamiento microscópico puede ser modelado por una cadena de Markov de 2 estados, mientras que el comportamiento macroscópico puede ser modelado por una cadena de Markov de 4 estados.

4. Se propondrá una metodología para emular pérdidas de paquetes, basada en los siguientes puntos.

   - Con el fin de representar diferentes niveles de rafagosidad (*bursts),* la descripción de paquetes perdidos en VoIP estará basada en comportamientos microscópicos y macroscópicos.

   - Para representar diversos patrones de pérdidas, se usarán cadenas de Markov de dos y cuatro estados.

   - Se crearan vectores de pérdida de paquetes por medio de un algoritmo que creará una secuencia binaria para representar los estados de paquete perdido (uno) o el estado de paquete recibido (cero). El *estado 1* es el estado de pérdidas y el *estado 2* es el estado de paquetes que llegaron a su destino.

Patrón de pérdidas: 00000110010101011011000000000000000000000001000000000
Estado: 22222112212121211211222222222222222222222221222222222

5. Pruebas: Para poner en práctica esta metodología, se introducirán los vectores generados de pérdida de paquetes a una aplicación de VoIP. En base a los elementos del vector se emularan las pérdidas, es decir, cuando se presente un “uno” el sistema VoIP no enviará el paquete y cuando se presente un “cero” el paquete será enviado.

6. Mediciones y Análisis: Por último, se realizará la captura de los patrones de tráfico y el correspondiente análisis de los principales parámetros de QoS.

CAPÍTULO

2

2 VOZ SOBRE EL PROTOCOLO DE INTERNET (VoIP)

El término VoIP hace referencia a la transmisión en tiempo real de la voz usando la tecnología IP sobre las redes de conmutación por paquetes. Consiste además, en un conjunto de recomendaciones y protocolos para el control y transmisión de paquetes de voz usando IP [1].

Voz sobre IP se puede definir también, como la transmisión de paquetes de voz utilizando redes de datos, la comunicación se realiza por medio del protocolo IP (Internet Protocol), permitiendo establecer llamadas de voz y fax sobre conexiones IP (Redes de Datos Corporativos, Intranets, Internet, etc), obteniendo de esta manera una reducción de costos considerables en telefonía. Una de las grandes desventajas de ésta tecnología es que el protocolo IP no ofrece calidad de servicio (QoS), por lo tanto es posible que se presenten retardos y pérdidas en la transmisión afectando de ésta manera la calidad de la voz.

La telefonía IP es una aplicación inmediata de VoIP que permite realizar llamadas telefónicas sobre las redes de datos sin necesidad de disponer de los circuitos conmutados convencionales provistos a través de la PSTN (Public Switching Telefony Network) que ha sido la red desarrollada a lo largo de los años para transmitir señales vocales.

La telefonía IP dispone de equipos que convierten la señal de voz analógica del teléfono en señal digital, posteriormente comprimen la información y la introducen en paquetes IP para ser transmitidos sobre una red IP, los cuales también tienen la habilidad de realizar el proceso inverso al momento que el paquete llega a su destino.

2.1 Funcionamiento de la VoIP

Básicamente el proceso inicia con la codificación de la voz, posteriormente se lleva a cabo el encapsulamiento o paquetización para transmitir los paquetes a través de una red IP y en el otro extremo de la nube, en el receptor, se realizar exactamente las mismas funciones en un orden inverso.

En las llamadas de VoIP, se utiliza la el esquema de conmutación de paquetes, un escenario en los que múltiples dispositivos comparten una sola red de datos, comunicándose mediante el envío de paquetes entre ellos, que contienen la información de direccionamiento en la que se especifica la dirección del equipo origen y destino, pudiendo seguir diversas rutas en función del estado y congestión de la red [2].

En la conmutación de paquetes, la información a transmitir se divide en unidades de información llamados paquetes, en los cuales se agrega información relevante también llamados encabezados. Estos paquetes viajan por flujos independientes, no existe una ruta predeterminada, es decir, los paquetes pueden viajar por el mejor camino entre dos puntos, donde siempre tienen más de un camino o ruta disponible, con mayores opciones por donde llegar a su destino, esto es una característica intrínseca de las redes IP, cuando los paquetes ya han llegado a su destino son re-ensamblados para reconstruir la información original, por lo tanto, se puede mencionar que la conmutación por paquetes hace uso más eficiente de los recursos de la red.

2.1.1 Proceso de codificación de la voz

Como se menciona anteriormente, el proceso de codificación es el primer paso dentro de una llamada VoIP. Este consiste en convertir la señal de voz analógica a una señal digital que luego se codifica adecuadamente.

En la Figura 2.1 se muestra el proceso de conversión analógico a digital (A/D) y digital a analógico (D/A) [3].

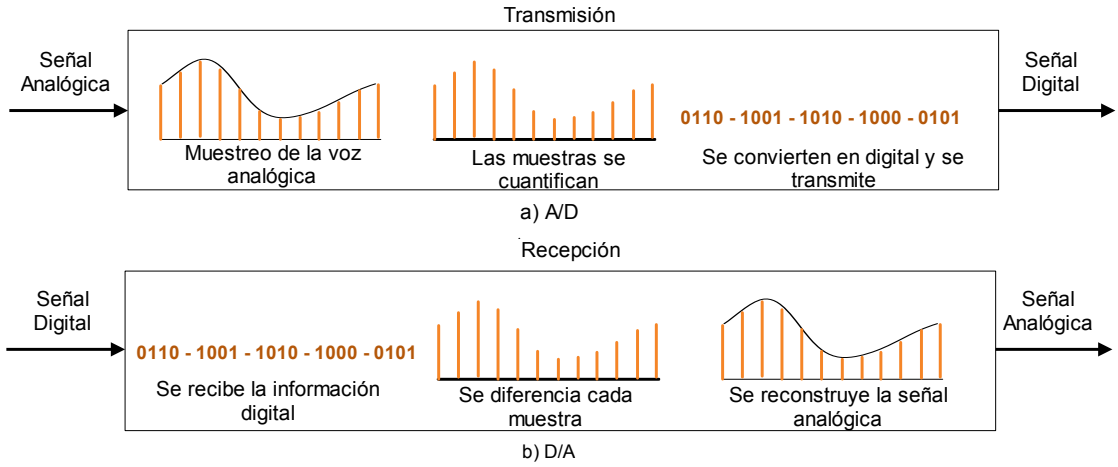


Figura 2.1 Conversión A/D y D/A

El objetivo es conseguir que la señal vocal analógica se digitalice y codifique, de manera que ésta pueda ser transmitida por redes de daos; al otro extremo es necesario recuperar la forma de onda original para lo cual se debe realizar el proceso inverso, es decir, decodificar la señal digital y convertir a una señal audible para que pueda ser escuchada por el ususrio final [4].

La telefonía IP tiene como reto convertir la señal analógica producida por un hablante en una señal digital, a este proceso se le conoce como digitalización de la voz. El proceso de digitalización consiste en tomar una muestra de la voz, cuantificarla y codificarla o convertir este valor en un número binario [3].

Las ventajas de la digitalización son tan evidentes que las redes telefónicas fueron evolucionando durante los años 80 y principios de los 90 hasta convertirse en redes digitales. La única parte analógica que queda en las redes telefónicas es el tramo que va desde el terminal telefónico a la central del operador. Este tramo es conocido como bucle de abonado o bucle local [3].

Para la realización el proceso de codificación, es necesario tres elementos, los cuales son: los conversores A/D y D/A, los procesadores digitales de la voz (DSPs) y los CODECs. Tal y como se muestra en la Figura 2.2 [4].

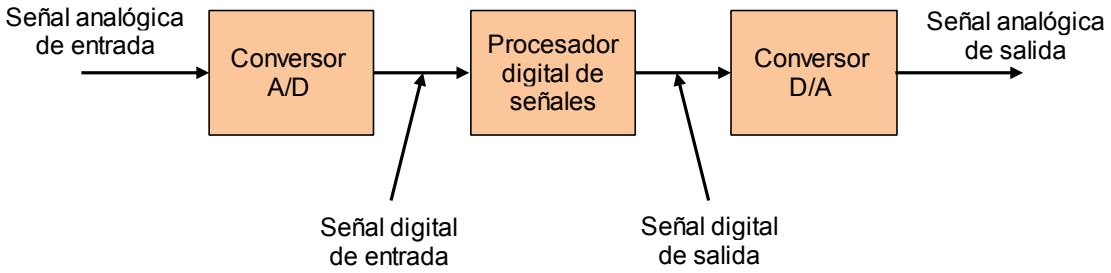


Figura 2.2 Proceso de codificación de la voz

2.1.1.1 Conversores A/D - D/A

Estos dispositivos pueden hacer operaciones importantes de entrada-salida en este proceso, una es la conversión de digital a analógico (D/A) y la otra es la conversión de analógico a digital (A/D).

El proceso de la conversión A/D costa de tres pasos [5], como se ilustra en la Figura 2.3.

- ***Muestreo.-*** Es la conversión de una señal en tiempo continuo a una señal en tiempo discreto obtenida tomando muestras de la señal en tiempo continúo en instantes de tiempo discreto.
- ***Cuantificación.-*** Es la conversión de una señal en tiempo discreto con valores continúos a una señal en tiempo discreto con valores discretos (señal digital). El valor de cada muestra de la señal se representa mediante un valor seleccionado de un conjunto finito de valores posibles.
- ***Codificación.-*** En este paso, cada valor discreto se representa mediante secuencia binaria de *b* bits.

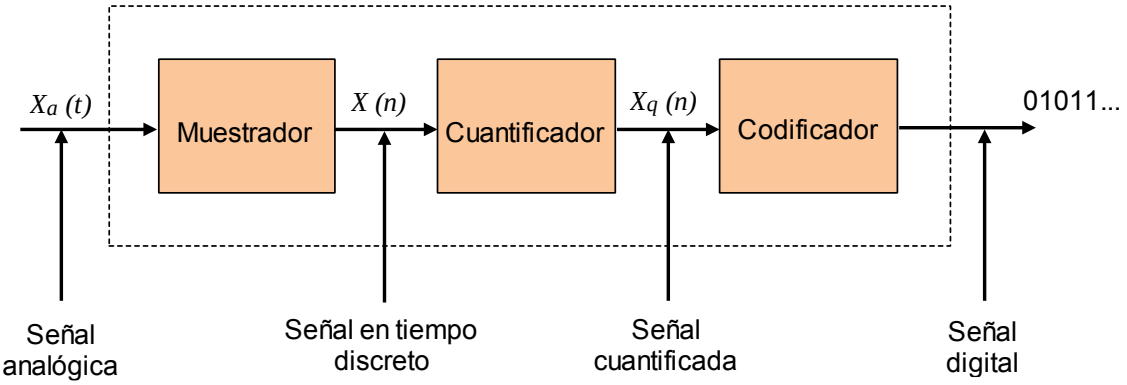


Figura 2.3 Partes básicas de un conversor A/D

Los conversores D/A son dispositivos que se encargan de recuperar la señal analógica original, el funcionamiento ideal de estos dispositivos consiste en hacer pasar el flujo finito de bits representados por pulsos, por un filtro de recuperación el cual reconstruye una aproximación de la señal muestreada por el conversor A/D [4].

2.1.1.2 Procesadores digitales de señal (DSP)

Un sistema de procesador digital de señal se puede entender como cualquier sistema electrónico que realice el proceso aplicando operaciones matemáticas a señales representadas de forma digital.

Los DSPs tienen la función de procesar las secuencias de números o símbolos entregadas por el conversor A/D, estos procesadores de señal realizan una gran cantidad de operaciones matemáticas basadas en algoritmos avanzados de procesamiento de señal que se deben realizar de manera rápida para evitar introducir retardos en la comunicación [4].

2.1.1.3 CODECs

Un CODEC es el software que se encarga de codificar archivos para poder almacenarlos sin que ocupen demasiado espacio, y de decodificarlos para que puedan ser utilizados en su momento. Estos son utilizados para obtener una comprensión de los datos.

Los COdificadores-DECodificadores constituyen una serie de especificaciones basadas en hardware y/o software para la codificación y decodificación de señales digitales. Es decir, un CODEC realiza la tarea de codificar una señal o flujo de datos y posteriormente descifrarla y recuperar así la señal original [4].

La mayoría de los CODECs utilizados para VoIP son estandarizados por la ITU-T y cada uno cuenta con los parámetros siguientes [4].

- Frecuencia de muestreo
- Tasa de bits
- Longitud de la trama
- Tamaño de la trama
- Requerimiento de DPS (MIPS)
- Retardo de extremo a extremo
- Retardo de procesamiento
- Memoria requerida
- MOS (Mean Opinion Score)
- Tramas/muestras por paquete
- Tasa de paquetes
- Trama por paquete
- Periodo de paquetización (sampling rate)
- Supresión de silencios

Existen una gran cantidad de CODECs y se diferencian unos de otros por ciertas características particulares que poseen; tales como el bit-rate ($kbps$),

retardo del algoritmo (ms), complejidad y la calidad de voz. Los algoritmos de codificación pueden ser divididos en tres tipos: CODECs de forma de onda, Vocoders (o codificación paramétrica) y CODECs híbridos [6]. En la Figura 2.4 se muestra una comparativa entre estos tres tipos de CODECs.

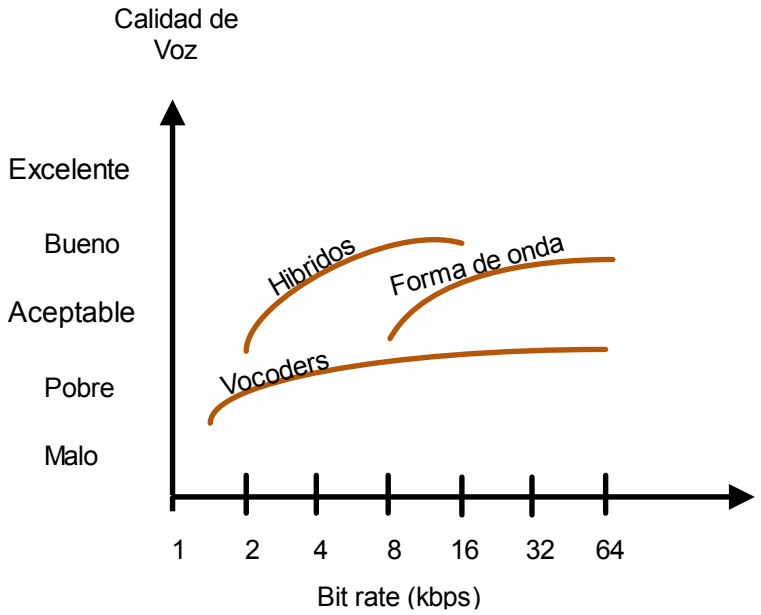


Figura 2.4 Comparación entre tipos de CODECs

Comúnmente los CODECs en forma de onda usan altos niveles de "bit-rates", proporcionan una buena calidad de la voz, pero consumen más ancho de banda. Los Vocoders con bajos "bit-rates" consumen menos ancho de banda, pero reproducen voz sintética y de menor calidad [3]. Los CODECs híbridos usan una combinación de las dos técnicas anteriores con el propósito de obtener una alta calidad de voz a bajos "bit-rates" (inferiores a $8\ kb/s$) [7].

En la Tabla 2.1 [8], se muestran algunos codecs usados en VoIP con sus respectivas características. Los diferentes codecs tienen los nombres que corresponden al nombre estándar de la ITU-T que describe su operación.

Codec	Coding Technique	Sampling Frequency (kHz)	Duration of Frame (ms)	Samples per frame	Bit Rate (kbit/s)	Complexity (MIPS)	Used Algorithms
G.711	Waveform	8	0.125	1	64	0.35	Companded PCM
G.726	Waveform	8	0.125	1	a) 16, b) 24, c) 32, d) 40	12	ADPCM
G.723.1	Hybrid	8	30	240	a) 5.3, b) 6.3	19	a) ACELP, b) MPC-MLQ
iLBC	Hybrid	8	20 o 30	160 o 240	a) 15.2, b) 13.3	a) 15, b)18	FB-LPC
G.729a	Hybrid	8	10	80	8	13	CS-ACELP
G.722	Waveform	16	0.0625	1	a) 48, b) 56, c) 64	10	SB-ADPCM
G722.2	Hybrid	16	20	320	A variety of nine different Bit Rates from 6.600 to 23.85	38	ACELP

Tabla 2.1 Tipos de Codecs de VoIP

2.1.2 Proceso de encapsulamiento de una trama VoIP

Todas las comunicaciones de una red parten de un origen y se envían a un destino. La información que se envía a través de una red es denominada datos o paquetes. Si una computadora le envía datos a otra, en primer término los datos deben empaquetarse a través de un proceso denominado encapsulamiento.

Los paquetes que llevan la voz se transportan empleando la siguiente estructura:

- Carga útil o Payload (muestra de voz).
- RTP
- UDP
- IP
- Nivel Físico (ATM,PPP, Ethernet u otro)

Una vez que la llamada ha sido establecida, la voz será digitalizada y entonces transmitida a través de la red en tramas IP. Las muestras de voz son primero encapsuladas en RTP y luego en UDP antes de ser encapsuladas en una trama IP. En la Figura 2.5 [4], se muestra el proceso de encapsulamiento a través de varios protocolos incluyendo IPv4, UDP y RTP.

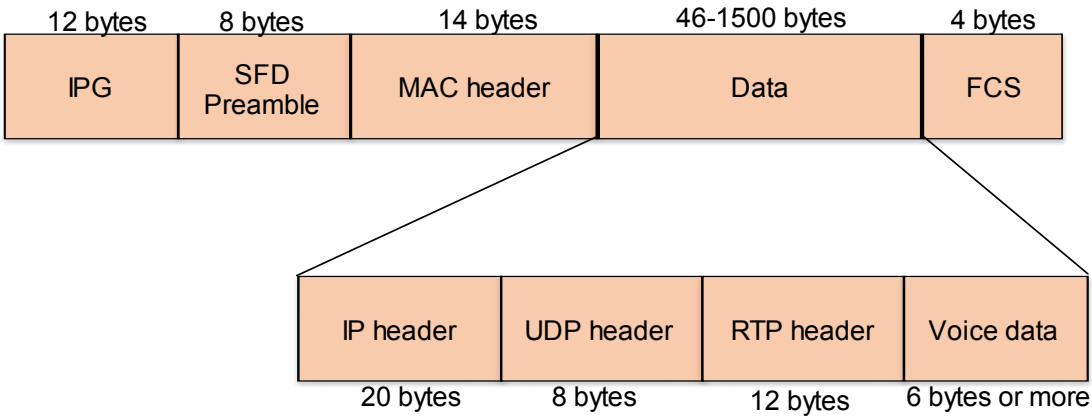


Figura 2.5 Encapsulamiento de una trama VoIP

Después de codificar la voz, es esencial encapsular la información de manera que se puedan transmitir por las redes de datos. Previo a revisar el proceso de encapsulamiento IP se deben estudiar los protocolos asociados a las comunicaciones en tiempo real, para esto se utiliza el protocolo RTP con su protocolo asociado RTCP [4].

2.1.2.1 RTP (Real-Time Transport Protocol)

RTP es un protocolo de nivel de aplicación, es utilizado para la transmisión de información en tiempo real sobre servicios de red multicast o unicast. Este permite transportar información de voz y video a través de Internet, es decir, paquetes encapsulados sobre protocolo IP.

Se usa frecuentemente en sistemas de streaming, videoconferencia y sistemas push to talk (en conjunto con H.323 o SIP), también representa la base de la industria de VoIP. El SRTP (Secure Real-Time Transport Protocol) es una extensión del perfil de RTP para conferencias de audio y vídeo para proporcionar confidencialidad, autenticación de mensajes y protección de reenvió para flujos de audio y vídeo.

El monitoreo del transporte de los datos es realizada mediante el protocolo RTCP, el cual provee descripción de funcionalidades y capacidades mínimas de control. RTP y RTCP están diseñados para trabajar independientemente de las capas de red y del método de transporte [4].

RTP utiliza UDP para el transporte de la información y aprovecha la suma de verificación (checksum) del mismo para verificar la integridad de los datos y aunque no tiene puertos fijos asociados en este protocolo, se utilizan los puertos pares, mientras que los puertos impares consecutivos son destinados para el control con RTCP [4].

Las funciones de RTP son:

- Identificación del tipo de carga útil transportada (CODECs de Audio/Vídeo).
- Verificar la entrega de los paquetes en orden (marca de tiempo) y si resulta necesario reordenar los paquetes fuera de orden.
- Transporte de información de sincronismo para la codificación y decodificación.
- Monitoreo de la entrega de información.

En la Figura 2.6 [4], se muestra el encabezado de un paquete RTP que presenta un formato fijo. Los primeros 12 octetos son fijos en todo el paquete RTP, mientras los identificadores CSRC (Contributing sorce) son insertados en elementos específicos en la red.

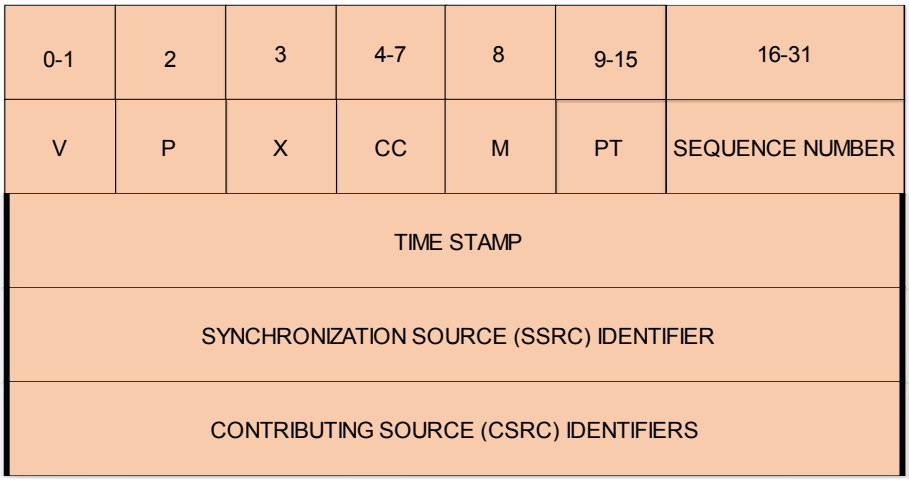


Figura 2.6 Estructura del encabezado RTP

2.1.2.2 RTCP (Real-Time Control Protocol)

Es el protocolo asociado a RTP y proporciona información acerca de la calidad de los paquetes RTP entregados. La función principal de este protocolo es la de proveer una retroalimentación de la calidad de servicio que provee RTP. RTCP recopila información asociada a RTP como paquetes entregados, paquetes perdidos, Jitter y round trip delay. Este protocolo está basado en la transmisión periódica de mensajes de control hacia todos los participantes de la sesión. Con este método de distribución de la información de control, es posible permitir a una entidad, como el proveedor de servicio actuar como un monitor externo de la calidad del servicio para diagnosticar problemas en la red [4].

Aunque la aplicación de estos dos protocolos no permite proveer calidad de servicio, RTCP tiene mecanismos para asegurar la entrega ordenada de los paquetes además RTCP provee información de la calidad de la recepción, información que se puede utilizar para tomar medidas externas relacionadas con la calidad [4].

Por otro lado, RTCP utiliza el mismo protocolo que RTP para enviar paquetes de control hacia todos los participantes de una sesión.

Los servicios que provee RTCP son los siguientes:

- Dar seguimiento a la calidad en la distribución de los datos, asi como mantener el control de los CODECs activos.
- Transportar un identificador constante para la fuente RTP (CNAME) para anunciar el número de participantes por sesión con el fin de ajustar la tasa de transmisión de datos.

2.1.3 Protocolos de señalización VoIP

Los protocolos de señalización son los encargados de crear, mantener y terminar una llamada en VoIP.

Existen varios protocolos de señalización, algunos con más popularidad que otros, pero son incompatibles entre sí. Es decir, que nuestro teléfono IP no sólo acepta un protocolo, afortunadamente los servidores de telefonía IP como los propios teléfonos IP pueden ser capaces de hablar más de un protocolo, lo cual amplía su interoperatividad.

Algunos de los protocolos más importantes para el desarrollo e implementación de un sistema que emplee VoIP, tanto sobre infraestructuras fijas como inalámbricas, se muestran en la Figura 2.7 [2].

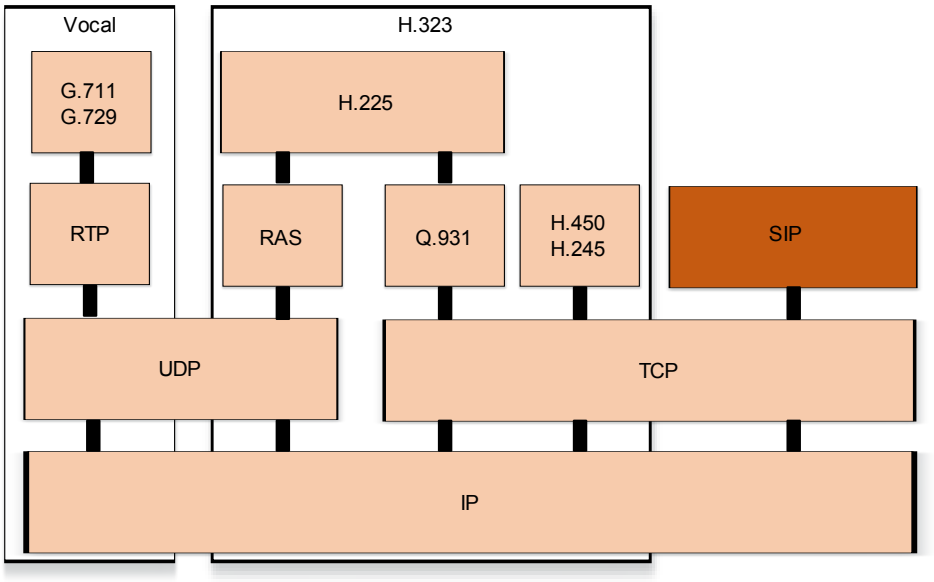


Figura 2.7 Familia de protocolos para VoIP

De la gran variedad de protocolos que son importantes en la telefonía IP, los más utilizados son los siguientes:

- H.323 que hace referencia a un conjunto de protocolos definidos por la ITU-T (International Telecommunication Union).
- SIP (Session Initiation Protocol, Protocolo de inicio de sesión) desarrollado por la IETF (Internet Engineering Task Force) y definido en el RFC3261.

2.1.3.1 H.323

Estándar de la unión internacional de las telecomunicaciones ITU-T 34 aprobado en mayo 1996 como un medio para transmitir comunicación de voz, video, data y fax a través de la redes IP mientras mantenían la conectividad con el PSTN [9]. Este es el encargado de definir el modo de interactuar de varios protocolos entre sí, debido a que este estándar cubre con la mayor parte de las necesidades requeridas para integrar la voz en redes IP se tomó la decisión que fuera la base de la tecnología VoIP.

El estándar H.323 es relativamente seguro y no requiere muchas consideraciones de seguridad de las que son comunes para cualquier comunicación de redes con el Internet. H.323 usa el protocolo RTP como medio de comunicación, y este no soporta de forma nativa las rutas de comunicación encriptadas. El uso de VPN u otros túneles encriptados entre puntos finales es la manera más común de encapsular la comunicación de forma segura [9].

La principal aportación de este estándar fue el desarrollo de un conjunto de protocolos de señalización que permiten controlar el establecimiento, mantenimiento y liberación de conexiones multimedia sobre redes de paquetes, ya que los protocolos para la transmisión de estos medios fueron adoptados de trabajos previos, principalmente desarrollados por el IETF a través de los protocolos RTP y RTCP.

En la Figura 2.8, se muestran los elementos fundamentales que forman parte de una red H.323, se divide en cuatro bloques, los cuales son: Terminales, MCUs, Gatekeepers y Gateways [10].

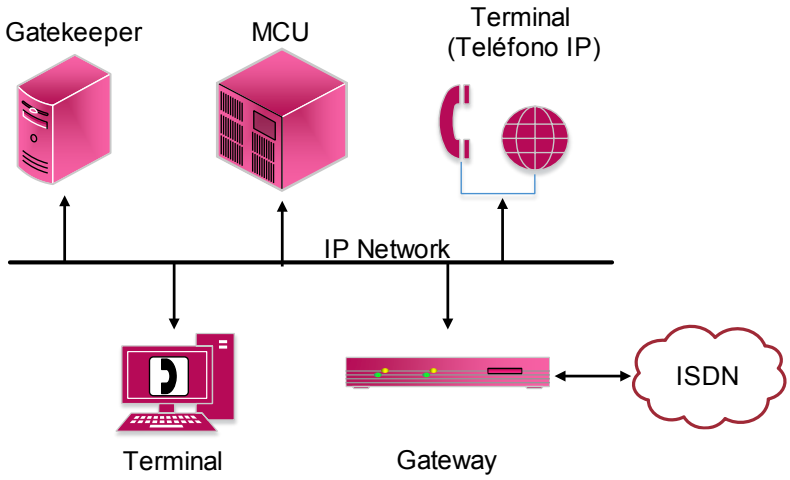


Figura 2.8 Elementos de una red H.323

- ***Terminales.-*** Son los puntos finales, es decir, serían el equivalente a los teléfonos actuales, su principal funcionamiento es el de realizar el tratamiento necesario de la señal para su correcto envío por la red de datos, proporcionando una comunicación bidireccional con otra terminal

H.323, cabe mencionar que cada una de estos terminales debe soportar al menos transmisión de voz, voz y datos, voz y video o voz, datos y video.

- ***Gatekeeper.-*** Se podría considerar como el punto central de la topología de una red H.323, su función principal es la de control y gestión de los recursos de la red.

  En términos generales el Gatekeeper define el concepto de zona H.323, es decir, toda terminal antes de realizar una llamada debe de consultar con el gatekeeper, una vez que obtiene el permiso, el GK es quien realiza el mapeo entre el identificador de usuario destino (dirección alias o número telefónico) y la dirección IP (dirección de transporte) equivalente, una vez lograda la comunicación entre los terminales, la tarea del GK concluyó y se reparte la carga del sistema entre las terminales.

  De esta manera el GK centraliza y mantiene un control de todo el tráfico generado en la red, y apoyándose en el control del ancho de banda puede fijar un límite de autorización y de ser necesario rechazar llamadas internas o externas.

- ***Gateway.-*** El gateway (GW) es un elemento esencial en la gran mayoría de las redes, es el encargado de conectar una red VoIP con una red PSTN, realizando la función de traductor. El GW se encarga de la traducción de diversos formatos de comunicación entre redes heterogeneas, los gateways H.323 se usan en entornos en los que se necesita salida a la telefonía convencional o para comunicar diferentes redes con otros gateways usando los protocolos H.245 y Q.931.

- ***Unidad de control multipunto.-*** Es el encargado de dar soporte a las conferencias entre tres o más puntos finales permitiendo la comunicación multipunto, los MCUs se dividen en dos partes:
  - Controlador multipunto
  - Procesador multipunto

La señalización de una llamada H.323 se caracteriza por las siguientes fases [11]:

1) ***Establecimiento de llamada.-*** En esta fase uno de los terminales se registra en el gatekeeper utilizando el protocolo RAS solicitando la identificación del usuario llamante enviando un mensaje ARQ; el GK aceptará la llamada y enviará al terminal llamante un mensaje de confirmación ACF o bien rechazará la llamada ARJ. Posteriormente utilizando el protocolo H.225 para establecer el canal de señalización, se manda un mensaje de SETUP para iniciar la llamada H.323 (utilizará la dirección IP y puerto recibidos del GK a través del mensaje ACF). El terminal llamado contesta con un CALL PROCEDING 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.
2) ***Intercambio e capacidades (H.245).-*** En esta fase se abre una negociación mediante el protocolo H.245, las entidades llamante y llamada determinaran los parámetros de la comunicación, es decir, que establecen quién será el maestro y quién será el esclavo, codificadores a utilizar, número de conexiones y direcciones a utilizar, puertos, número de muestras por trama, en esta negociación se abre el canal de comunicación (direcciones IP, puerto). Los principales mensajes de H.245 que se utilizan para esta fase son: *TerminalCapability Set* (mensaje de intercambio de capacidades soportadas por los terminales que intervienen en una llamada) y *OpenLogicalChannel* (mensajes 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).
3) ***Intercambio de información multimedia.***- Los terminales inician la comunicación y el intercambio de audio o video mediante la arquitectura RTP/UDP/IP, así como canales de control a través de la arquitectura RTP/UDP/IP para los canales de realimentación, al objeto de controlar la

calidad de los flujos de información recibida por el otro extremo de la comunicación.

4) ***Terminación de llamada.***- En esta fase cualquiera de las entidades H.323 pueden iniciar el proceso de finalización de llamada mediante mensajes *CloseLogicalChannel* y *EndSessionComand* a través del canal 2.245. posteriormente utilizando H.225 se cierra la conexión con el mensaje RELEASE COMPLETE; y por último se liberan los registros con el gatekeeper utilizando mensajes del protocolo RAS.

Hay una gran variedad de codecs que utiliza H.323, pero los utilizados con mayor frecuencia son los siguientes [12]:

- **G.711.-** Se emplea principalmente en la telefonía clasica. Utiliza 56 kbps o 64 kbps para la señalización y 64 kbps para la transmisión. Opera a una tasa de 8.000 muestras por segundo, que codifica logarítmicamente y emplea las frecuencias de la voz humana (300 6y 3,4 kHz). Usa códigos para el silencio y la repetición de valores. Existen varios tipos:
  - **U-law:** G.711u empleado en Norteamérica y Japón.
  - **A-law:** G.711a empleado en Europa y en el resto del mundo, fue diseñado para utilizarse en computadoras.
- **G.729.-** Comprime fragmentos de audio de 10 ms, no sirve para tonos DTMF o fax-Gestiona los silencios y el ruido. Usa 8 kbps 0 13 kbps para señalización y 31,5 para la transmisión.

2.1.3.2 SIP

A diferencia de H.323, SIP (Session Initiation Protocol) tiene su origen en la comunidad IP, específicamente en la IETF y no en la industria de las Telecomunicaciones ITU-T. SIP es un protocolo que se diseñó con el objetivo de que fuera muy simple. De cierto modo es similar al protocolo HTTP en muchos sentidos, ambos usan un esquema de petición-respuesta, ambos usan una

codificación en ASCII para presentar los mensajes y en ambos casos existen distintos métodos y códigos de respuesta [13].

El protocolo SIP es un estándar para la inicialización, modificación y finalización de sesiones interactivas de usuario (con uno o más participantes) donde intervienen elementos multimedia (audio, vídeo, datos, mensajería instantánea, juegos en línea y realidad virtual). Se emplea para la señalización VoIP y 3G y permite determinar la ubicación de los usuarios. Por defecto utiliza el puerto 5060 [12].

Los elementos incluidos en SIP son [12]:

- **Agentes de usuario (user agent).-** Son los terminales o clientes manejados por una persona o un software, que emiten o reciben los mensajes de dicho protocolo. Los User Agent se comportan de dos maneras: como clientes al realizar una petición (User Agent Clients); y como servidores al recibirla (User Agent Servers).
- **Servidores de registro.-** Para saber en qué punto de la red el usuario está conectado, se necesita un mecanismo de registro. El usuario tiene un URI (es como un e-mail) independientemente de su ubicación, el usuario hace una petición *Register* al servidor, para que este asocie la dirección lógica a una IP (esta asociación se llama binding, y es caduca y no renovable).
- **Servidores proxy y de redirección.-** Para encaminar el mensaje se necesitan servidores intermedios, que pueden ser proxy (encaminadores) o de redirección, generando una respuesta que indica la dirección del destino o de otro servidor que lo acerque al destino.

La señalización de una llamada SIP, se resume en un esquema sencillo entre dos teléfonos IP llamándose directamente, la cual produce una secuencia, como se observa en la Figura 2.9 [13].

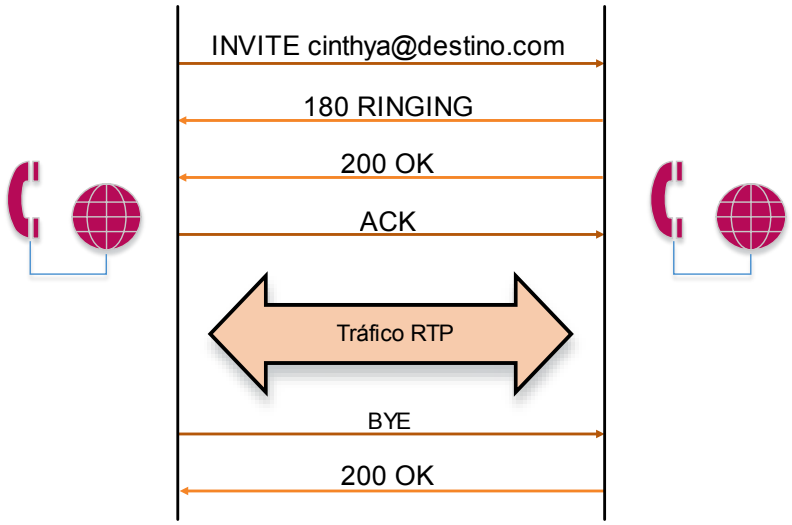


Figura 2.8 Llamada directa entre dos teléfonos IP usando SIP

El usuario del teléfono A está llamando al teléfono B. La secuencia es como sigue [13]:

1) El usuario del teléfono A llama al usuario cinthya@destino.com, para ello se envía una petición usando el método *INVITE*. En esa petición se indica no solo el destino sino también los codecs que soporta el teléfono.

2) El teléfono del usuario B, si está activo, indicará al teléfono A que la llamada está en curso, es decir, que el teléfono del usuario B está sonando. Para ello responde, en primera, con *"180 RINGING"* con lo que indica un código de respuesta provisional a la vez que indica al usuario del teléfono A que está avisando al usuario B.

3) Si el usuario del teléfono B levanta el teléfono, se enviará un mensaje de respuesta desde B a A indicando que la llamada se ha aceptado. Es el mensaje *"200 OK*", además se envían parámetros de configuración de lo que va a ser el canal para enviar la información de audio.

4) El teléfono A confirma que ha recibido esa indicación de *OK* y para ello envía una petición usando el método *ACK*.

5) En este momento se va a producir el intercambio de información de audio, la llamada se ha establecido y los usuarios están hablando entre ellos.

6) En el instante en que cualquiera de los dos teléfonos cuelgue, se enviará desde este una petición al otro teléfono usando el método *BYE*, con lo que se indicará el deseo de terminar la conversación.

7) El otro extremo responde con *OK*.

El teléfono del usuario A enviará la petición *INVITE* al servidor proxy que tenga configurado y este se encargará de buscar la ubicación del teléfono B. la secuencia se muestra en la Figura 2.10 [13]:

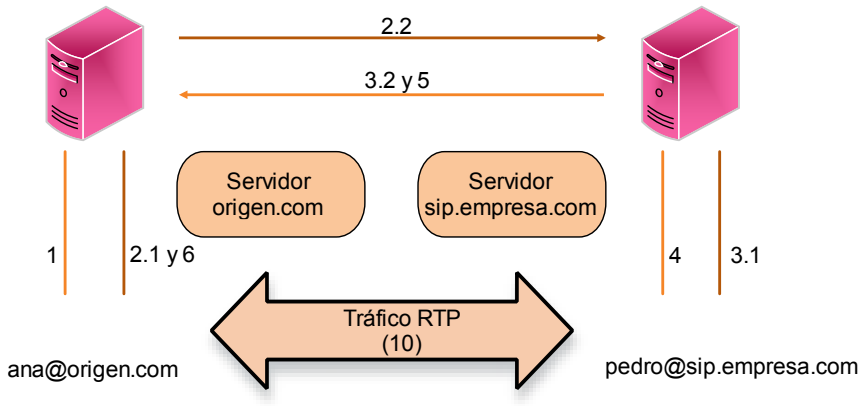


Figura 2.9 Llamada a través de servidores usando SIP

2.2 Calidad de servicio (QoS)

En la actualidad las redes IP soportan un sin número de aplicaciones distintas, muchas de las cuales requieren bajo retardo, de otra manera se verá afectada la calidad, o causará que la aplicación simplemente no funcione. Para poder dar un tratamiento diferenciado a la voz con respecto a los datos, es necesario implementar técnicas de calidad de servicio a nivel de red.

Con el paso del tiempo, se podría pensar que VoIP lograría reemplazar a la red de telefonía tradicional (PSTN), por lo tanto, los clientes necesitan recibir la misma calidad de transmisión de voz que reciben con los servicios de telefonía básica, esto significa una consistente alta calidad en la transmisión de voz. Al igual

que otras aplicaciones en tiempo real, VoIP es sensible a la pérdida, a los retardos y ancho de banda.

De acuerdo a la ITU-T, la calidad de servicio (QoS) es el efecto global de la calidad de funcionamiento de un servicio que determina el grado de satisfacción de un usuario de un servicio [14].

La calidad de servicio (QoS) extremo a extremo necesita de contribuciones que aporten los componentes, tales como se muestran en la Figura 2.11 [14]:

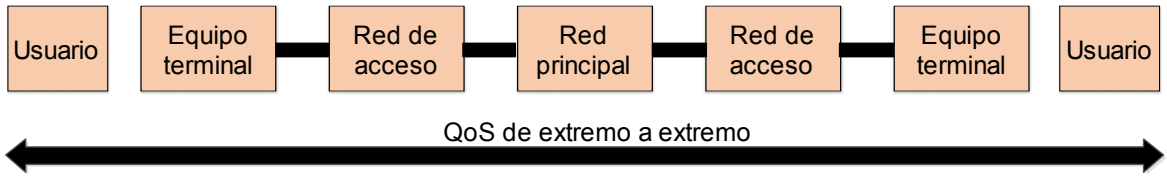


Figura 2.10 Esquema de contribuciones a la QoS de extremo a extremo

Cada contribución se explica a continuación [14]:

- La configuración mostrada pertenece al servicio convencional con usuarios a cada extremo de la conexión. De igual modo, se pueden aplicar a los servicios ofrecidos por un proveedor de servicios, en un extremo, y un usuario en el otro.
- Equipo terminal: este puede depender de la variabilidad de la calidad de funcionamiento del equipo terminal.
- Red de acceso: esta contribución depende de la combinación de medios de acceso y de tecnologías utilizadas para un servicio concreto.
- Red principal: esta puede ser la de un único proveedor o la concatenación de redes de distintos proveedores.

La QoS se puede separar en cuatros puntos de vista (Recomendación G.1000), tal y como se muestra en la Figura 2.12 [14]:

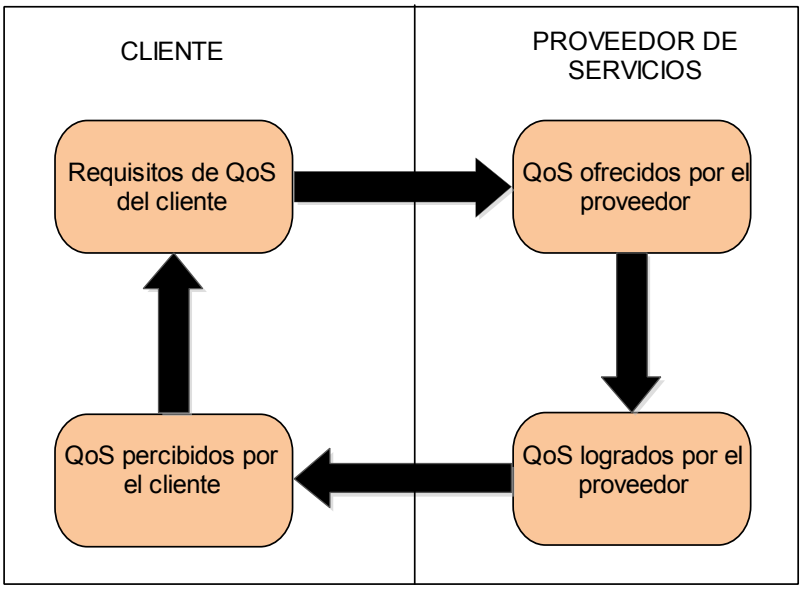


Figura 2.11 Cuatro polos de la calidad de servicio

2.2.1 Parámetros de desempeño

Los problemas de QoS en VoIP vienen derivados de dos factores principalmente: a) Internet es un sistema basado en conmutación de paquetes 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 perjudiquen la comunicación.

2.2.1.1 Retardo

Llamado también latencia, es la variación temporal o retraso introducido por la transmisión de los paquetes de datos desde la fuente hasta el destino. Este parámetro depende de muchos elementos como el número de nodos por los cuales tienen que pasar los paquetes hasta alcanzar el destino, el tráfico de la red, los protocolos de enrutamiento, etc.

La latencia son los huecos o gaps que aparecen en la conversación como consecuencia de los retardos introducidos en cada una de las fases de la transmisión de una conversación VoIP y por la congestión de la red. Una latencia inferior a $150\,ms$ no es perceptible inmediatamente, pero a partir de ese valor comienza a afectar negativamente a la conversación; un valor de $300\,ms$ se hace intolerable en una conversación de voz [15].

El primer retardo es en la matriz de switch y luego el retardo de procesamiento. A todo esto se suman los retardos propios del proceso de comprensión vocal.

Los servicios en tiempo real y multimedia son sensibles a retardos. En aplicaciones como la videoconferencia es necesario que este parámetro sea reducido al mínimo.

El retardo, o retardo total de transmisión, resulta debido a los diferentes fenómenos que ocurren dentro de un sistema VoIP y el medio de transmisión, que pueden ser provocados por la codificación aplicada, por el esquema de corrección de errores, retardos introducidos por la red, los buffers, y decodificadores [6]. El retardo en un solo sentido (OWD), que ocurre en Internet, es el tiempo necesario para que un paquete atraviese la red desde la fuente al host destino [16]. Esto es descrito analíticamente por la ecuación 2.1.

$$D^K(L)_{OWD} = \delta + \sigma + \sum_{h=1}^{s}\left(\frac{L}{C_h} + X_h^K(t)\right) \qquad 2.1$$

Donde:

- $D^K(L)_{OWD}$ es el retardo OWD de un paquete k de tamaño L
- δ representa el retardo de propagación
- σ es el retardo de procesamiento
- s el número de saltos
- L/C_h el retardo de transmisión

- $X_h^K(t)$ el retardo de encolamiento de un paquete K de tamaño L en el salto h $(h = 1, \dots, s)$ con capacidad C_h [17]

La Tabla 2.2 muestra los diferentes retardos que se producen en milisegundos en un sistema VoIP.

Fuente de retraso	*Rango Típico*
Recording	10-40 ms
Codificación	10-20 ms
Retraso de extremo a extremo	70-120 ms
Buffer	50-200 ms
Decodificador	10-20 ms
TOTAL	**150-400 ms**

Tabla 2.2 Tipos de retardos que se producen en un sistema VoIP

2.2.1.2 Eco

Las características de latencia y Jitter pueden producir eco sobre la señal telefonía, lo cual hace necesario el uso de canceladores de eco.

Existen dos tipos de eco: Uno de ellos tiene alto nivel y poco retardo y se produce en el circuito híbrido de 2 a 4 hilos local. El otro, es de bajo nivel y gran retardo y se produce en el circuito híbrido remoto [18].

Los ecos se producen cuando la persona que llama escucha su propia voz retrasada por tan sólo 30 ms. El tiempo que toma para transmitir de una persona que llama de audio para viajar a la persona que llama la recepción y volver a su auricular del teléfono de la llamada el retardo de ida y vuelta y es importante en lo que respecta a eco. Cuanto mayor es el retardo de ida y vuelta y la falta de concordancia, peor es el eco potencial. Hay dispositivos de procesamiento de señales digitales muy complejas llamadas canceladores de eco que puede ayudar en la cancelación del eco en sistemas VoIP. G.164, G.165, y G.168 están

aprobados por la UIT-T para asegurar el correcto desarrollo de los canceladores de eco [19].

El cancelador de eco se construye mediante la técnica de ecualización transversal autoadaptativa, que consiste en usar una parte de la señal de transmisión para cancelar el eco producido por la desadaptación de impedancias en el circuito híbrido que convierte de 4 a 2 hilos [18].

2.2.1.3 Pérdida de paquetes

La pérdida de paquetes es producida por los errores en alguno de los medios de transmisión o a la congestión de la red.

Indica el número de paquetes perdidos durante la transmisión. Normalmente se mide en tanto por ciento. Por ejemplo, los routers pierden/niegan/descartan paquetes por muchas razones, en general en función del estado de la red.

La pérdida de paquetes aceptable esta una función del tamaño del paquete. Un tamaño de paquete de alto valor disminuirá el retardo extremo a extremo, pero si el paquete se pierde, la degradación será proporcinal al tamaño del paquete. Por otro lado, un mayor porcentaje de pérdida de paquetes puede ser tolerado si el tamaño de paquete es reducido, por otro lado, el tamaño de paquete más pequeño tiene el potencial de aumentar el retardo de extremo a extremo [19]. Por tal motivo existe un compromiso entre los retardos y la pérdida de paqutes.

Hay otras muchas razones que pueden causar la pérdida de paquetes en entornos inalámbricos: enlaces de red saturados, colisiones, rotura de enlace, etc. La eliminación de paquetes depende únicamente del estado de la red, y esto no puede ser previsto.

2.2.1.4 Jitter

Los paquetes enviados pueden llegar al destino siguiendo diferentes caminos, por lo tanto el retardo de los paquetes puede variar. El jitter es la variación o diferencia de retardo que existe entre los paquetes, causada por congestión de red, pérdida de sincronización o por las diferentes rutas seguidas por los paquetes para llegar a su destino.

El jitter es debido a la secuencia desigual de llegada de paquetes UDP. Esta variación de retardo entre paquetes provoca inestabilidad y, dada la naturaleza de las redes de conmutación de paquetes, es difícil conseguir los paquetes de VoIP en llegar a su destino a una velocidad constante. Los paquetes de VoIP son enviados a una velocidad constante y llegan a su destino en el extremo de la red TCP/IP en momentos ligeramente diferentes [19].

El jitter es el efecto que se produce debido a un retardo o latencia variable entre los paquetes. En otras palabras, es la consecuencia por el cual el retardo entre paquetes no es constante. Esto hace que los paquetes no lleguen en orden, dejando huecos en la secuencia de tramas de la conversación [15].

Se trata de una latencia variable producida por la congestión de tráfico en el backbone de red. En este caso, se puede emplear un búffer para distribuir los paquetes y reducir el jitter, pero esto introduce un retardo adicional. Lo ideal es incrementar el ancho de banda del enlace. Otra posibilidad es la formación de colas de menor prioridad al tráfico de telefonía sobre los datos [18].

En la Figura 2.13 nos muestra el comportamiento que sigue el jitter durante su trayectoria a través de la red IP.

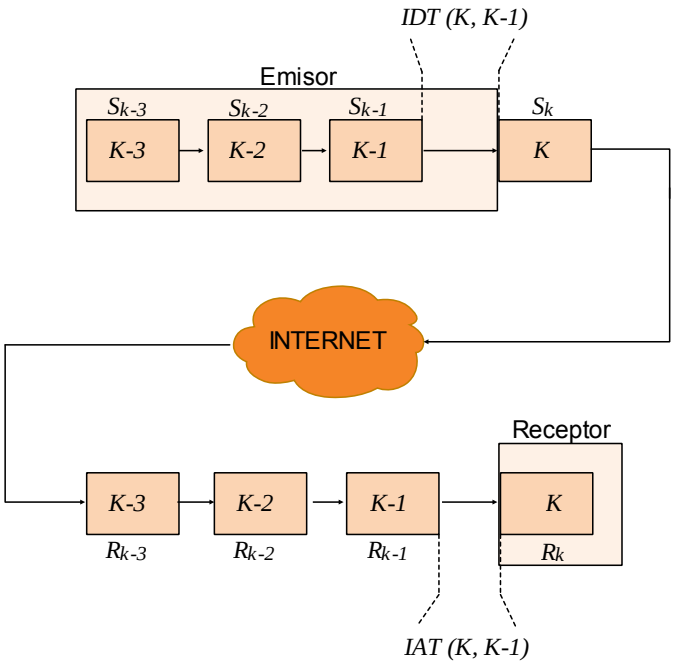


Figura 2.12 Comportamiento del jitter a través de una red IP

El RFC 1889 define al jitter como la diferencia de los "*relative transit time*" de dos paquetes consecutivos; el "*relative transit time*" de un paquete, 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 de tiempo [7].

La diferencia en el "*relative transit time*" entre dos paquetes adyacentes J^K, se calcula de acuerdo a la acuación 2.2:

$$J^K(L) = (R_K - S_K) - (R_{K-1} - S_{K-1}) \qquad 2.2$$

Donde:

- R_K es la hora de llegada del paquete K en el receptor
- S_K es la hora en que se envió el paquete K (estampa de tiempo RTP)
- R_{K-1} es la hora en que se recibió el paquete $K-1$
- S_{K-1} es la hora de llegada del paquete $K-1$ en el receptor

Derivado de lo anterior, el jitter de arribo (jitter utilizado en este trabajo) es calculado de acuerdo a la siguiente ecuación:

$$IAT(K, K-1) = J^K(L) + IDT(K, K-1) \quad 2.3$$

Donde:

- $J^K(L)$ es el valor calculado mediante la ecuación 2.2
- $IDT(K, K-1) = (S_K - S_{K-1})$ es la diferencia del tiempo de partida de dos paquetes consecutivos
- $IAT(K, K-1) = (R_K - R_{K-1})$ es la diferencia del tiempo de arribo de dos paquetes consecutivos o jitter de arribo para los paquetes K y $K-1$

En este trabajo, llamaremos jitter a $IAT(K, K-1)$.

2.2.1.5 Relación entre Pérdida de Paquetes y Jitter

El efecto que tiene las pérdidas de paquetes en el jitter, se muestra en la Figura 2.14 y de la ecuación 2.3. Derivado de este efecto, se puede establecer una relación analítica entre el jitter y la pérdida de paquetes como sigue:

Sea $IAT(K, K-1) = J^K(L) + IDT(K, K-1)$ (ecuación 2.3) el jitter entre el paquete K y $K-1$.

Si el paquete $K-1$ y $K-2$ se pierden, la ecuación 2.3 toma la siguiente forma:

$$IAT(K, K-3) = J^K(L) + (3)IDT \quad 2.4$$

Por lo tanto, si n paquetes consecutivos se pierden, observe la Figura 2.15, tendremos:

$$IAT(K, K-n-1) = J^K(L) + (n+1)(IDT) \quad 2.5$$

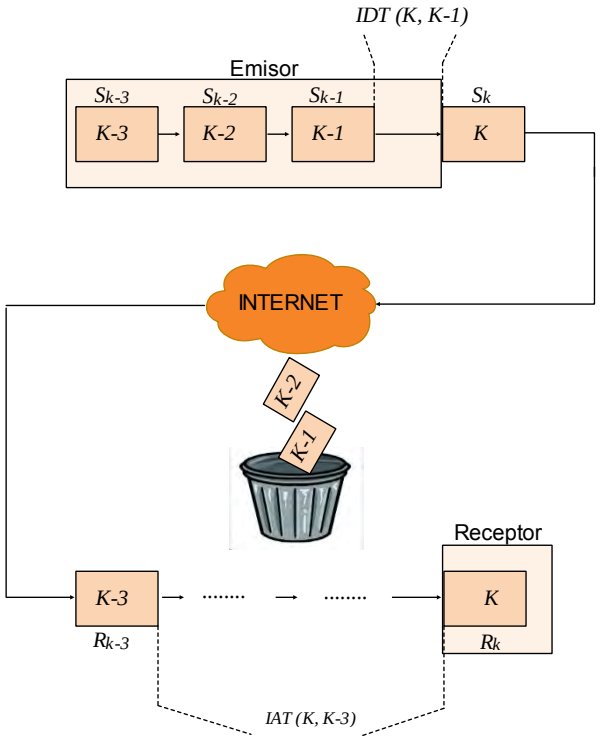


Figura 2.13 Comportamiento del jitter ante 2 paquetes perdidos

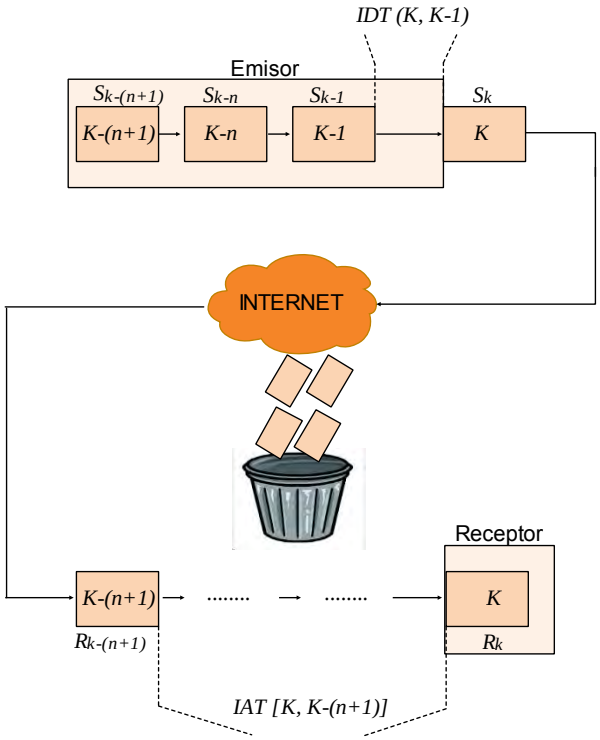


Figura 2.14 Comportamiento del jitter ante una ráfaga de "n" paquetes perdidos

2.2.2 Medidas de la Calidad de Servicio

Los usuarios finales son los que determinan la calidad de servicio de manera objetiva y/o subjetivamente en función de los desperfectos en la red. La medida de calidad usada en este trabajo, será el MOS [21], y puede ser obtenido a partir del cálculo del factor R del modelo E [20-21]. A continuación se explica detalladamente cada una de estas medidas.

2.2.1.1 Medidas Subjetivas: MOS

El MOS es un método derivado del Índice de Categoría Absoluta (ACR de sus siglas en inglés), es un método para evaluar los sistemas de transmisión de voz [1].

En el MOS se requiere que algunas personas, llamadas oyentes, evalúen la calidad de voz de varias muestras de la comunicación; así, éste método está basado sobre promedios de puntuaciones provistas por muchos oyentes que van desde $1, 2, 3, 4, y\ 5$, y tienen una calidad "mala", "pobre", "relativamente buena", "buena" y "excelente", respectivamente [1].

2.2.1.2 Medidas Objetivas: Modelo E

El *modelo E* es un modelo computacional diseñado para producir el MOS sin la realización de pruebas subjetivas. Las pruebas subjetivas no son prácticas porque consumen mucho tiempo. Estas pruebas pueden ser sustituidas por el *Modelo E* [1].

En el modelo E, se combinan algunos desperfectos (los efectos de los retardos, los paquetes perdidos, entre otros) dentro de una red, para proporcionar un valor llamado factor R, que varía de 0 (en el peor de los casos) hasta 100 (en el mejor de los casos). El factor R se expresa de la siguiente manera:

$$R = R_0 - I_s - I_d - I_e + A \quad (2.6)$$

Donde:

- R_0 representa la relación señal a ruido (SNR) que puede llegar a tomar hasta un valor de 100
- I_s es la combinación de todos los desperfectos que aparecen de forma casi simultánea con la señal de voz
- I_d representa los errores causados por los retardos
- I_e son las degradaciones causadas por los CODECs
- A es el factor de expectación, el cual captura el hecho que los usuarios pueden aceptar algo de degradación de la calidad por el hecho que están usando VoIP

Para opciones prácticas se puede reducir la ecuación 2.6 como sigue [20]:

$$R = 93.2 - I_d(T) - I_e(CODEC, PLR) \quad (2.7)$$

$$I_d = 0.024(T) + 0.11(T - 177.3)y(T - 177.3)$$

$$T = OWD = \frac{1}{2}RTT, \quad y(x) = \begin{cases} 0, & x < 0 \\ 1, & x > 0 \end{cases} \quad (2.8)$$

$$I_e(G.711) \sim 0 + 30\ln(1 + 15 \cdot PLR)$$

$$I_e(G.729) \sim 11 + 40\ln(1 + 10 \cdot PLR) \quad (2.9)$$

Habitualmente, los valores del factor R son categorizados como se muestra en la Tabla 2.3.

Factor R	*MOS*	*Satisfacción del Usuario*
$\mathbf{90 \leq R < 100}$	$4.34 - 4.50$	Muy satisfecho
$\mathbf{80 \leq R < 90}$	$4.03 - 4.34$	Satisfecho
$\mathbf{70 \leq R < 80}$	$3.60 - 4.03$	Algunos usuarios insatisfechos
$\mathbf{60 \leq R < 70}$	$3.10 - 3.60$	Muchos usuarios insatisfechos
$\mathbf{0 \leq R < 60}$	$1.00 - 3.10$	Todos los usuarios insatisfechos

Tabla 2.3 Factor R y MOS y su relación en los usuarios

2.2.3 Modelos de servicios

El objetivo es evitar que la congestión en determinados nodos de la red afecte a algunas aplicaciones que requieren un determinado ancho de banda, o un mínimo retardo, sin la necesidad de ocupar excesivo ancho de banda. Para evitar dicha congestión existen dos tecnologías o modelos que proporcionan una determinada Calidad de Servicio (QoS), ya que estos ofrecen propiedades que debe tener un servicio y que este ofrece a las aplicaciones que lo usan.

Los dos tipos de modelo son: servicios integrados (IntServ) y servicios diferenciados (DiffServ).

2.2.3.1 Modelo de servicios integrados

Este modelo es basado en la utilización de algún protocolo de reserva de recursos (RSVP), el cual nos permite realizar una reserva de recursos en todos los routers que participan en la comunicación.

El modelo IntServ intenta integrar todos los tipos de tráficos posibles en una misma red de uso general. Este ofrece servicios cuantificables y mesurables en el sentido que son definidos para proporcionar una determinada calidad de servicio para un tipo de tráfico cuantificado. Este modelo está asociado a mecanismos de admisión y reserva de recursos en la red [22].

El protocolo de reserva describe cómo una aplicación negocia el nivel de calidad de servicio. El más simple es que una aplicación pida una calidad de servicio particular y que la red se lo proporcione o lo deniegue. Sin embargo, más que rechazar la petición, la red podría conceder un nivel de recursos menor que el pedido. Un esquema más complejo es el modelo de reserva de “doble pasada”. En este esquema, se propaga la especificación del tráfico inicial desde el origen a los posibles destinos [22].

El principal inconveniente que se puede presentar en esta tecnología, radica en la necesidad de mantener información sobre cada flujo en todos los routers de la red, lo cual conduce a problemas de escabilidad.

2.2.3.2 Modelo de servicios diferenciados

En este modelo a diferencia del IntServ, se basa en la división del tráfico en diferentes clases, y en la asignación de prioridades a los agregados. Para ello los paquetes se marcan a la entrada de la red, según diferentes categorías, de las cuales se establecen diferentes parámetros de QoS. En una misma clase se agregan diferentes flujos que recibirán el mismo tratamiento.

En este modelo, la red clasifica el tráfico en distintas clases y les aplica una disciplina de servicio diferenciada con el objetivo de proporcionar distintos niveles de calidad de servicio. En este caso no se reservan recursos por lo que no se puede garantizar a priori una calidad de servicio [22].

De este modo, se pueden tener varias clases de servicio para tiempo real, con varios niveles de retardo. También habrá niveles con servicio predictivo y otros sólo con garantía de entrega. El cliente escogerá el tipo de servicio en función del tráfico a transmitir y por supuesto, el precio que quiera pagar. Otra de las ventajas de este modelo es su menor complejidad de implementación y su fácil integración con los protocolos IP, en el que cada paquete puede ser marcado con la clase de servicio que requiere [22].

Referencias

[1] Homero Torral, Modelado de los Parámetros de QoS de Tráfico VoIP Auto-similar y una Mejora al Modelo E, Tesis de Doctorado, CINVESTAV Unidad Guadalajara, 2010.

[2] J. M. Huidobro, Telefonía sobre IP. Baja la factura del teléfono, Autores científico-técnicos y académicos, 2009.

[3] J. A. Carballar, VoIP: la telefonía IP, Madrid, Paraninfo, 2008.

[4] Andrés. A. Domínguez Hernández, Análisis de implementación y recomendación de soluciones corporativas de comunicaciones unificadas NGN Based con Business Communications Manager (BCM) de Nortel, Tesis de Licenciatura, Escuela Politécnica Nacional, 2009.

[5] J. G. Proakis, D. G. Manolakis, Tratamiento digital de señales, Madrid, Pearson, 1998.

[6] Alexander Raake, Speech Quality of VoIP: assessment and prediction, John Wiley & Sons Inc., 2006, ISBN: 0-470-03060-8

[7] Jesús Argaez, Software para el análisis de QoS en VoIP, Tesis de Maestría, CINVESTAV Unidad Guadalajara, Julio 2009.

[8] Michail Papanikolaou, Speech Codecs analysis, basic arithmetic operations, Tesis de Diploma, Universidad Técnica Nacional de Atenas, 2013.

[9] B. Pérez, Asterisk: Instalación, Configuración y puesta en marcha, México: Asterisk, 2012.

[10] ITU, Diseño y Configuración de dos Plataformas de Interfonía H.323, 2010.

[11] J. I. Moreno, I. Soto, D. Larrabeiti, Protocolos de Señalización para el transporte de Voz sobre redes IP, Madrid, 2008.

[12] J. A. Gómez, Servicios en red, Madrid, Editex, 2012.

[13] F. Sivianes Castillo, G. Sánchez Antón, J. Ropero Rodríguez, Servicios en red, Madrid: Paraninfo, 2010.

[14] UIT-T "E.800": Serie E: Explotación general de la red, servicio telefónico, explotación del servicio y factores humanos, Definiciones de términos relativos a la calidad de servicio, Ginebra, 2009.

[15] M. M. Vallina, Infraestructuras de redes de datos y sistemas de telefonía, Madrid, Paraninfo, 2013.

[16] ITU-T, “G.114: One way transmission time”, Telecommunication Standardization Sector, Geneva, Switzerland, 2003.

[17] R. Prasad, C. Dovrolis, M. Murray and K.C. Claffy, “Bandwidth estimation: metrics, measurement techniques, and tools”, IEEE network, vol. 17, no. 6, 2003, pp. 27-35.

[18] R. C. P. José Manuel Huidobro Moya, Sistemas de Telefonía, Madrid, Paraninfo, 2006.

[19] J. F. Ransome, J. R. Rittinghouse, VoIP Security, Estados Unidos de América, Elsevier Digital Press, 2005.

[20] ITU-T, “G.107: The E-Model, a computational model for use in transmission planning”, Telecommunication Standardization Sector, Geneva, Switzerland, 2009.

[21] ITU-T, “G.108: Application of the E-Model: A planning guide”, Telecommunication Standardization Sector, Geneva, Switzerland, 1999.

[22] C. A. Vega Lebrún, D. Arvizu Gutiérrez, A. García Santillán, Algoritmos para Encriptación de datos, Veracruz, 2008.

CAPÍTULO

3

3 NATURALEZA DEL TRÁFICO IP

La teoría de tráfico consiste en la aplicación de modelos matemáticos para explicar la relación que existe entre la capacidad de una red de comunicaciones, la demanda de servicio que los usuarios le imponen y el nivel de desempeño que la red puede alcanzar. Como dicha demanda es de naturaleza probabilística y estadística, se suele representar mediante algún proceso estocástico adecuado, con lo que se constituyen diferentes modelos de tráfico. Así pues, dado un modelo de tráfico particular, el desempeño de la red se podría predecir, en principio, aplicando herramientas adecuadas proporcionadas principalmente por la teoría de procesos estocásticos y otros recursos matemáticos. Los resultados de dicho análisis d e desempeño son los puntos de partida para el diseño de mecanismos de control de la red en variados aspectos.

En redes modernas de comunicaciones, es importante poder encontrar relaciones entre el tráfico y el desempeño, con las cuales se pueda determinar que tipos de garantías de servicio pueden ofrecerse. Por supuesto, no podemos esperar que dichas relaciones se puedan expresar de una manera tan compacta, pero sí debemos ser capaces de encontrar procedimientos de diseño de redes y de administración de los recursos de la red en los que se tengan en cuenta las características esenciales del tráfico que afectan significativamente las medidas de desempeño y en los que se ignoren las características irrelevantes. Con este propósito, resulta de fundamental importancia desarrollar modelos de tráfico que capturen dichas características.

A lo largo del desarrollo de las redes de comunicaciones, se han propuesto diferentes modelos de tráfico, cada uno de los cuales ha resultado útil dentro del contexto particular para el que se propuso. Este aspecto es importante, pues un modelo puede ser tan bueno como otro si ambos satisfacen pruebas de hipótesis adecuadas. A partir de la necesidad de prestar servicios integrados con una única estructura de red, el modelado de tráfico se ha convertido en una extensa área de

investigación, en la que el objetivo es desarrollar modelos que predigan el impacto de la carga impuesta por las diferentes aplicaciones sobre los recursos de la red, de manera que se pueda evaluar la calidad de servicio (QoS) ofrecida.

El tráfico se modela mediante un proceso estocástico que representa la demanda que los usuarios de una red de comunicaciones imponen sobre los recursos de la misma. En un principio, se consideró que los tiempos entre llegadas de las demandas de los usuarios eran independientes entre sí, así como la cantidad misma de la demanda. Después, se vio la necesidad de incluir el efecto de la correlación existente entre estas variables, para lo cual se desarrollaron modelos más elaborados en los que la correlación decaía exponencialmente con el tiempo. Sin embargo, recientemente se ha evidenciado que, en las redes modernas de comunicaciones, la correlación entre estas variables no decae tan rápidamente y puede persistir a través de muchas escalas de tiempo. Este fenómeno, que afecta significativamente el desempeño de las redes de comunicaciones, se puede representar adecuadamente mediante modelos de tráfico autosimilar.

Los fenómenos de autosimilitud en el tráfico de las redes actuales de comunicaciones se descubrieron hace ya más de una década [1, 2], sin embargo, los efectos que este fenómeno produce en el desempeño de las redes han obligado a desarrollar numerosas investigaciones hasta nuestros días.

Los modelos tradicionales de tráfico permiten fácilmente controlar la variabilidad de la demanda y, por consiguiente, con ellos resulta relativamente fácil ejercer control de tráfico de manera que se puedan garantizar algunos niveles mínimos de calidad de servicio. Desafortunadamente, el fenómeno de la autosimilitud puede conducir a estructuras complejas de correlación en las que la variabilidad se extiende a muchas escalas de tiempo, invalidando las técnicas de control diseñadas para dichos modelos tradicionales de tráfico[3].

3.1 Modelos clásicos de Poisson

El modelo de Poisson, a semejanza del binomial, consta de varios ensayos de Bernoulli. La diferencia consiste en que el modelo binomial sirve para calcular la probabilidad de ocurrencia de un resultado particular en un número finito de repeticiones, mientras que con el modelo de Poisson se determina la probabilidad de ocurrencia de un determinado evento en el tiempo o el espacio y no en un número definido de repeticiones del experimento. En estos eventos que se producen aleatoriamente en el espacio o el tiempo, la frecuencia de ocurrencia de un evento es tan baja con relación a la frecuencia de no ocurrencia que se consideran como sucesos raros.

El proceso de Poisson establece que la probabilidad de que haya un número de llegadas k en un tiempo T dado está dado por [4]:

$$P(k|T) = \frac{(\lambda T)^k}{k!} e^{(-\lambda T)} \qquad 3.1$$

Donde:

- k es el número de llegadas en un tiempo T
- λ es el número medio de llegadas

Y la probabilidad de que haya un número de finalizaciones j en un tiempo dado está dado por:

$$P(j|T) = \frac{(\mu T)^j}{j!} e^{(-\mu T)} \qquad 3.2$$

Donde:

- j es el número de finalizaciones en un tiempo T
- μ es el número medio de finalizaciones

Los intervalos entre llegadas (λ) o entre finalizaciones (μ) son independientes y tienen una distribución exponencial negativa.

$$f(t) = \mu e^{(-\mu|t|)} \mu > 0 \qquad 3.3$$

Sin embargo, a pesar de su sencillez, estos modelos no se ajustaban al tráfico de paquetes. A partir de estudios [2], se demuestra que el proceso que genera el tráfico de paquetes no es el mismo que el de la Red Telefónica Pública Conmutada ya que presenta propiedades de autosimilitud. Por lo tanto, el proceso de Poisson no era adecuado para modelar el comportamiento de la red de paquetes.

3.2 Procesos Autosimilares

A partir de los años 80`s se han venido desarrollando modelos de tráfico que tienen en cuenta la correlación existente entre llegadas de paquetes a *corto plazo*. Es decir, al calcular la función de autocovarianza de los procesos estocásticos correspondientes se encontró que la velocidad a la que esta decae en el tiempo es exponencial, como se muestra en la ecuación 3.3. Sin embargo, mediciones y estudios recientes han demostrado que la función de autocorrelación en muchas trazas de tráfico real, decae más lentamente que una exponencial con el tiempo, es decir, de acuerdo a una ley de potencia.

En el clásico estudio presentado por Leland, Willinger, Taqqu y Wilson en ACM SIGCOMM `93 [1] y posteriormente la versión extendida [2] se mostró, tras la observación de exhaustivas mediciones realizadas sobre una red Ethernet en Bellcore, que el tráfico era de naturaleza “autosimilar” (self-similar).

En esta sección se presenta la definición de los principales conceptos involucrados en el tema general del tráfico autosimilar en redes de comunicaciones de una manera manera relativamente sencilla, con el fin de ofrecer una herramienta para abordar el tema del tráfico autosimilar en la literatura especializada.

La autosimilitud describe el fenómeno en el que ciertas propiedades de un objeto se preservan sin importar el escalamiento en el tiempo o en el espacio.

- ***Proceso Autosimilar:*** Un proceso estocástico continuo de valores reales $\{Y(t), -\infty < t < \infty\}$ es llamado *autosimilar* si para una constante $\alpha > 0$ existe un $H > 0$, llamado índice de autosimilitud, tal que:

$$\{Y(at), t \in \Re\} \stackrel{d}{=} \{a^H Y(t), t \in \Re\} \forall \alpha > 0, \alpha \in \Re \quad 3.4$$

Donde:

- $Y(at)$es la versión escalada de $Y(t)$.
- a^H es un parámetro de normalización.
- $\stackrel{d}{=}$ representa igualdad de la distribución finita.

Lo anterior quiere decir que $Y(t)$ y $a^{-H}\, Y(at)$ tienen idéntica distribución en diferentes escalas de tiempo. Para una serie de tiempo discreta la definición se da en términos de la serie agregada.

Así, una segunda definición de autosimilitud, muy próxima a la primera, aunque no equivalente, es la usada por la siguiente serie agregada:

$$X(m) = \left\{X_k^{(m)}; k = 1,2,3, \ldots\right\} \quad 3.5$$

Dónde:

Cada término$X_k^{(m)}$ se define como:

$$X_k^{(m)} = \frac{1}{m}\sum_{i=(k-1)m+1}^{km} X_i; k = 1,2,3, \ldots \quad 3.6$$

Dónde:

m representa el nivel de agregación; es decir, cada nueva serie es obtenida partiendo la original en bloques disjuntos de tamaño m y promediando cada bloque para obtener los k valores de la nueva serie. Como se muestra en la F'igura 3.1.

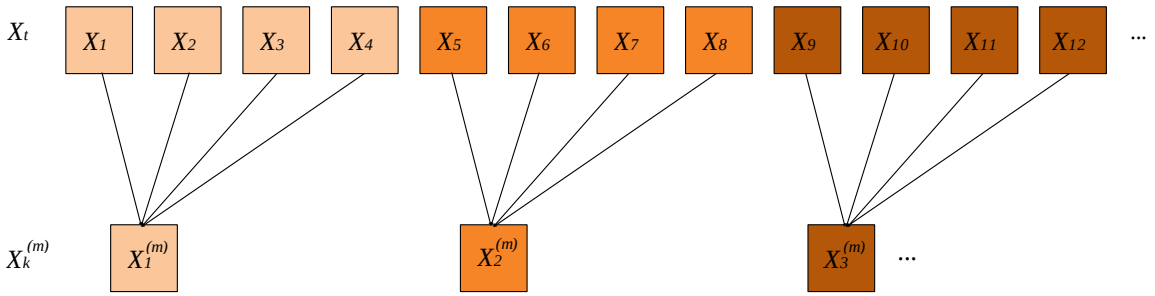


Figura 3.1 Ejemplo de Autosimilitud

- ***Autosimilitud de una serie agregada:*** se dice que X es auto-similar si satisface la siguiente ecuación:

$$X^{(m)} \overset{d}{=} m^{H-1} X \qquad 3.7$$

- ***Autosimilitud de Segundo Orden:***se dice que X es auto-similar de segundo orden con parámetro de Hurst (H), si satisface la siguiente ecuación:

$$var\left(X^{(m)}\right) = var(X) \cdot m^{2H-2} \qquad 3.8$$

- ***Dependencia de corto rango (SRD):*** Se dice que un proceso es SRD si su función de autocovarianza decae exponencialmente. Para un proceso SRD, $0 < H < 0.5$.
- ***Dependencia de largo rango (LRD):*** La dependencia de largo rango implica que la función de autocorrelación de un proceso decae más lentamente que una exponencial, es decir, de acuerdo a una ley de potencia. Este tipo de comportamiento nos indica la presencia de valores muy altos con probabilidad no despreciable. Para un proceso LRD, $0.5 < H < 1$.

Modelos como el de Poisson no contemplan el fenómeno de LRD y aplicarlos al tráfico actual puede tener como consecuencia mal diseño de elementos de red, por ejemplo, la subestimación del tamaño del buffer de encolamiento.

3.3 Estimación del Parámetro H: Método de la varianza

Existen muchos métodos para evaluar el parámetro H en una serie de tiempo, sin embargo, uno de los más utilizados y fáciles de implementar es el método de la varianza, el cual está basado en la definición de la autosimilitud de segundo orden.

De acuerdo a la ecuación 3.8, una serie de tiempo es autosimilar de segundo orden si se cumple:

$$var\left(X^{(m)}\right) = var(X) \cdot m^{2H-2} = \frac{var(X)}{m^{\beta}}$$

Donde $H = 1 - \frac{\beta}{2}$.

Si aplicamos logaritmos a ambos miembros de la expresión obtenemos:

$$\log var\left(X^{(m)}\right) = \log var(X) - \beta \log m$$

Si se grafican en un par de ejes los distintos puntos que surgen de ir escalando por m, la variable X, donde las abscisas están representadas por $\log m$, y las ordenadas por $\log var\left(X^{(m)}\right)$, se puede observar que los puntos se agrupan en torno a una línea recta de pendiente $-\beta$ y cuya ordenada al origen es la constante $\log var(X)$. Entonces si se realiza una regresión a la colección de puntos y se calcula el valor de β, se puede estimar el valor de H de la siguiente forma:

$$H = 1 - \frac{\beta}{2}$$

Referencias

[1] W.E. Leland, M.S. Taqqu, W. Willinger, and D.V. Wilson. (1993). “On the Self-Similar Nature of Ethernet Traffic”, Proc ACM SIGCOMM, Pages 183–193.

[2] W.E. Leland, M.S. Taqqu, W. Willinger, and D.V. Wilson. (1994). “On the Self-Similar Nature of Ethernet Traffic” (Extended Version), IEEE/ACM Transactions on Networking, 2:1–15, 1-15.

[3] K. Park K and W. Willinger. (2000). “Self-Similar Network Traffic and Performance Evaluation”, Jhon Wiley Interscience, New York.

[4] J. F. Hayes and T. V. J. Ganesh Babu. (2004). “Modeling and Analysis of Telecommunications Networks”, Jhon Wiley Interscience, New Jersey.

CAPÍTULO

4

4 MODELADO Y SIMULACIÓN DE PAQUETES PERDIDOS

La pérdida de paquetes es una de las más importantes degradaciones variables en el tiempo, puesto que degrada la calidad de la voz, y está directamente vinculado a las técnicas de transmisión (protocolos usados) de paquetes [1].

Las aplicaciones como la voz y video son más susceptibles a los cambios en las características de transmisión de redes de datos. La pérdida de paquetes se produce a lo largo del trayecto de los datos, lo cual degrada seriamente la calidad de la voz.

En las redes de datos, la pérdida de paquetes es muy común y esperada debido a su naturaleza de recursos compartidos. Algunos protocolos de datos utilizan la pérdida de paquetes para conocer las condiciones de la red y poder reducir el número de paquetes que se envían [2]. La pérdida de paquetes marca la diferencia entre la tecnología VoIP y la red telefónica clásica [1].

4.1 Cadenas de Markov

Una *cadena de Markov* o *proceso de Markov* es aquel en el cual, la probabilidad de que el sistema esté en un estado particular en un periodo de observación dado, depende solamente de su estado en el periodo de observación inmediato anterior [3].

Supongamos que el sistema tiene n estados posibles. Para cada $i = 1,2,\dots,n$, y cada $j = 1,2,\dots,n$, sea t_{ij} la probabilidad de que si el sistema se encuentra en el estado j en cierto periodo de observación, estará en el estado i en

el siguiente: t_{ij} recibe el nombre de *probabilidad de transición*. Además, t_{ij} se aplica a cada periodo; es decir, no cambia con el tiempo [3].

Como t_{ij} es una probabilidad, debemos tener que:

$$0 \leq t_{ij} \leq 1 \qquad (1 \leq i, j \leq n)$$

Asimismo, si el sistema está en el estado j en cierto periodo de observación, entonces debe estar alguno de los n estados (ya que también podría permanecer en el estado j en el siguiente). Por lo tanto, tenemos:

$$t_{1j} + t_{2j} + \cdots + t_{nj} = 1 \qquad 4.1$$

Es conveniente disponer las probabilidades de transición como la matriz $T = [t_{ij}]$ de n x n, llamada matríz de transición de la cadena de Markov. Otros nombres para una matríz de transición son *matríz de Markov*, *matríz estocástica* y *matríz de probabilidades*. Como podemos ver, las entradas en cada columna de T son no negativas y, de acuerdo con la ecuación 4.1, suman 1 [3].

4.1.1 Cadena de Markov de 2 estados

Consideramos una cadena de Markov con dos estados, 1 y 2, de tal manera que podemos ir de 1 a 2 con probabilidad p y de 2 a 1 con probabilidad q.

Una ráfaga de paquetes perdidos y recibidos sigue el comportamiento de una cadena de Markov de dos estados (también conocido como modelo de Gilbert) y puede ser descrita mediante dos parámetros (p y q) [1], como se muestra en la Figura 4.1.

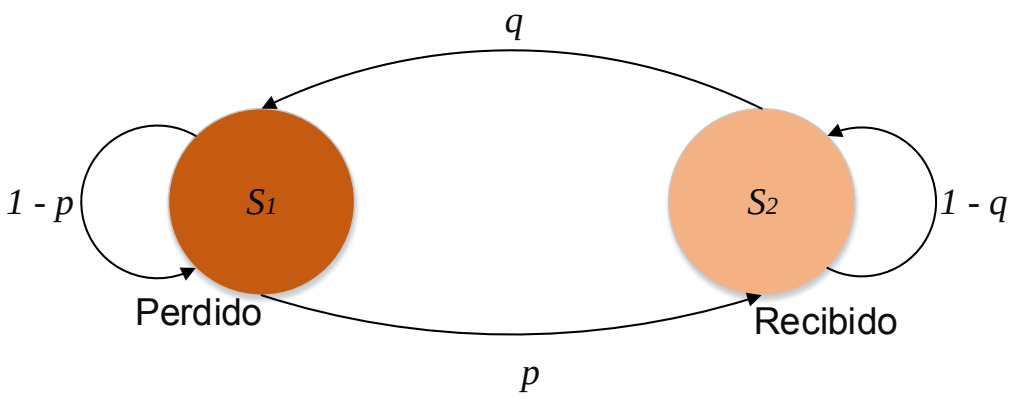


Figura 4.1 Cadena de Markov de 2 estados

Donde p y q son las dos probabilidades de transición, p (transición perdido→recibido) y q (transición recibido→perdido). El estado S_1 representa los paquetes perdidos y S_2 representa el estado en el que los paquetes son recibidos [4].

La probabilidad total de pérdidas [1] o probabilidad de estado estable [5], S_1, puede ser representada por la ecuación 4.2, normalmente llamado PLR.

$$S_1 = \frac{q}{p+q} \qquad 4.2$$

Donde S_1 es la probabilidad de ocupar el estado de pérdidas [5].

Claramente podemos observar que [4]:

$$S_2 = 1 - S_1 \qquad 4.3$$

La distribución de pérdida descrita con el modelo de Markov de dos estados, puede también ser caracterizado por un vector o secuencia binaria; este vector estará compuesto por paquetes perdidos (unos) y paquetes recibidos (ceros) [1].

La distribución del número consecutivo de paquetes perdidos $\left(f_b(k)\right)$ y paquetes recibidos $\left(f_g(k)\right)$, pueden ser expresados en términos de p y q [1].

$$f_b(k) = q(1-q)^{k-1} \qquad 4.4\ (a)$$

$$f_g(k) = p(1-p)^{k-1} \qquad 4.4\ (b)$$

Estas dos distribuciones geométricas nos indican el número de pasos k para cambiar de un estado a otro, en este caso, del estado de paquetes perdidos al estado de paquetes recibidos $f_b(k)$, o viceversa $f_g(k)$.

El promedio de paquetes perdidos y paquetes recibidos pueden ser calculadas como la esperanza de k y deducidas a partir de las ecuaciones 4.4 (a) y 4.4 (b). El promedio de paquetes perdidos quedaría como sigue [6]:

$$\bar{b} = E\{k\} = \sum_{k=1}^{\infty} k * q(1-q)^{k-1} \qquad 4.5$$

$$= q \sum_{k=1}^{\infty} k\, p^{k-1}$$

Donde $p = 1 - q$.

También es fácil de ver que

$$= q \sum_{k=1}^{\infty} \frac{d}{dp}\left(p^k\right)$$

$$= q \frac{d}{dp}\left[\sum_{k=1}^{\infty} p^k\right] = q \frac{d}{dp}\left[\frac{p}{1-p}\right] = q\left[\frac{(1-p)\frac{d}{dp}(p) - p\frac{d}{dp}(1-p)}{(1-p)^2}\right]$$

$$= q\left[\frac{(1-p) - p(-1)}{(1-p)^2}\right] = \frac{q}{(1-p)^2}$$

Pero $q = 1 - p$

$$= \frac{q}{q^2} = \frac{1}{q} \qquad 4.6$$

De igual forma podemos calcular el promedio de los paquetes encontrados consecutivamente, quedando de la siguiente forma:

$$\bar{g} = E\{k\} = \frac{1}{p} \quad 4.7$$

También se puede llegar a la siguiente deducción [1]:

$$S_1 = \frac{\bar{b}}{\bar{b} + \bar{g}} \quad 4.8$$

4.1.2 Cadena de Markov de 4 estados

Cuando se usa un gran número de estados, es decir, una cadena de Markov de n estados; es para lograr una mayor precisión en el modelado de una serie de tiempo de PLR de pérdida de paquetes de tráfico VoIP. En este trabajo utilizaremos únicamente dos tipos de cadenas de Markov; las cuales son las cadenas de Markov de dos estados en la parte práctica y las cadenas de Markov de cuatro estados en la parte teórica.

El modelo de la cadena de Markov de cuatro estados explica que existen dos niveles de rafagosidad; el *estado malo,* en el cual hay muchos paquetes perdidos y el *estado bueno,* en el cual hay pocos paquetes perdidos. En la Figura 4.2 se muestra este modelo.

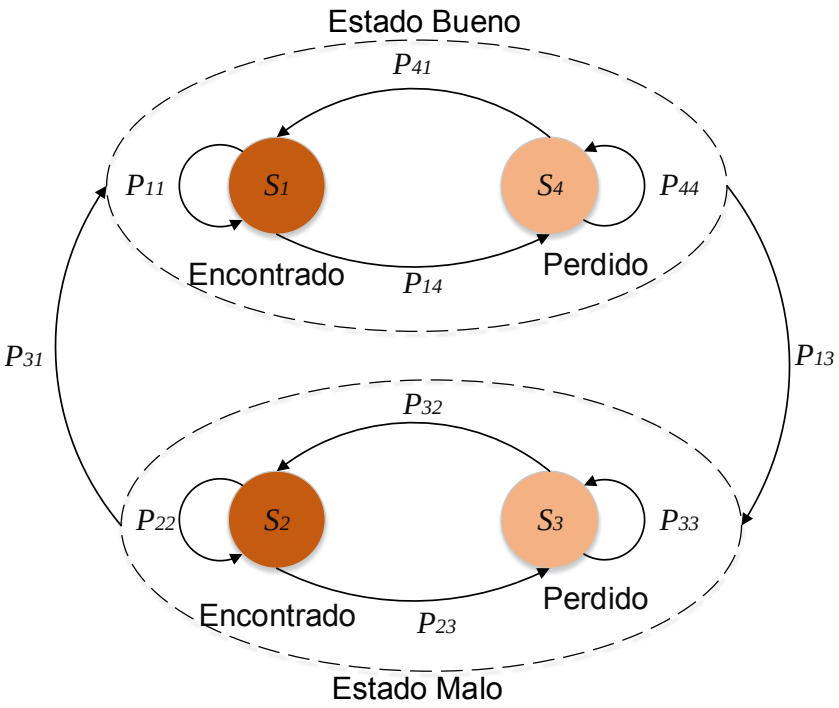


Figura 4.2 Cadena de Markov de 4 estados

En la cadena de Markov de 4 estados, es necesario hallar todas las probabilidades de transición, los cuales son los estados S_1 y S_2 (pertenecen a los paquetes recibidos), los estados S_3 y S_4 (pertenecen a los paquetes perdidos) y los parámetros $[p_{14}, p_{41}, p_{23}, p_{32}, p_{13}, p_{31} \in (0,1)]$.

La probabilidad de estado estable cuando está en el estado bueno es [5]:

$$\pi_B = \frac{P_{13}}{P_{13} + P_{31}} \qquad 4.9$$

y la probabilidad de estado estable de estar en el estado malo es [5]:

$$\pi_G = \frac{P_{31}}{P_{31} + P_{13}} \qquad 4.10$$

En el *estado bueno* (G) la pérdida de paquetes ocurre con probabilidad baja P_G, mientras que en el *estado malo* (B) ocurre con probabilidad alta P_B.

Todos los paquetes perdidos en un estado bueno o malo, pueden ser calculados siguiendo las ecuaciones que a continuación se presentan.

$$P_G = \frac{p_{14}}{p_{14}+p_{41}}; \quad \text{4.11 (a)}$$

$$P_B = \frac{p_{23}}{p_{23}+p_{32}} \quad \text{4.11 (b)}$$

La probabilidad total de todos los paquetes perdidos para una cadena de Markov de 4 estados está dada por la ecuación:

$$PLR = \pi_G P_G + \pi_B P_B \quad \text{4.12}$$

4.2 Distribución de paquetes perdidos

Debido al comportamiento variable en el tiempo de la red de paquetes, los paquetes perdidos pueden mostrar una gran variedad de distribuciones. La distribución más usada para estudiar la calidad de voz es la distribución de Bernoulli [1].

Consideremos la llegada exitosa y la pérdida de un paquete hacia un destino, como los resultados exclusivos A (es el suceso elemental que tiene uno de los dos posibles resultados como su elemento) y $\bar{A}$ (es el otro único posible suceso elemental) que ocurren con probabilidades p y q, respectivamente. Ahora, establecemos que el suceso A ocurre en cualquier prueba dada con la probabilidad $P(A) = p$; el suceso $\bar{A}$ entonces tiene la probabilidad $P(\bar{A}) = 1 - p$. Definimos la variable aleatoria discreta X, donde $X = 1$ para cuando A ocurre y $X = 0$ para cuando $\bar{A}$ ocurre. Se describe de la siguiente manera [7]:

$$\begin{aligned} p(1) &= p \\ p(0) &= q \end{aligned} \quad \text{4.13}$$

Donde $q = 1 - p$.

La media y la varianza para X estan dadas por las ecuaciones 4.14 (a) y 4.14 (b), respectivamente:

$$\mu = p \qquad \text{4.14 (a)}$$

$$\sigma^2 = p\,(1-p) \qquad \text{4.14 (b)}$$

En este trabajo, el uso de esta distribución no será de mucha utilidad, porque esta establece que las pérdidas se presentan de manera independientes, es decir, que la pérdida de un paquete no depende si el paquete anterior llegó o se perdió; a esto llamaremos pérdidas homogéneas. Esta distribución no representa el comportamiento real de internet; cuando un paquete se pierde es muy probable que el siguiente también se pierda, y los consiguientes también, creando una ráfaga de paquetes perdidos en un periodo de tiempo dado [5].

Para referirnos a una ráfaga de paquetes perdidos usaremos el término de *burst* [1], esto quiere decir, que si el paquete n se pierde es muy probable que el siguiente paquete $n-1$ también se pierda, esto significa que se crea una dependencia entre n y $n-1$.

4.3 Modelado de pérdida de paquetes

Una serie de tiempo de PLR o traza de pérdida de paquetes de tráfico VoIP, puede modelarse usando modelos de estados finitos en tiempo discreto, así como, las cadenas de Markov. Se puede obtener un modelo simplificado de una traza de pérdida, utilizando únicamente una cadena de Markov de dos estados. Para lograr mayor precisión, es conveniente utilizar más números de estados [1].

Observemos una secuencia de variables aleatorias $X_0, X_1, \ldots X_n$, y supongamos que el conjunto de todos los posibles valores son $\{0, 1, \ldots, M\}$. Donde X_n es el estado de algún sistema en el tiempo n, y, de acuerdo con la interpretación, decimos que el sistema está en estado j en el tiempo n si $X_n = j$.

La secuencia de variables aleatorias nos dice que una cadena de Markov está formada por:

$$P_{ij} = P(X_n = j | X_{n-1} = i) \quad 4.15$$

Una cadena de Markov es un proceso aleatorio donde el valor de la variable aleatoria en un instante n depende *sólo* de su valor en su pasado inmediato $n-1$. [8].

Los valores $P_{ij},\ 0 \leq i \leq M,\ 0 \leq j \leq M$, son llamados probabilidades de transición de la cadena de Markov y satisfacen lo siguiente

$$P_{ij} \geq 0 \qquad \sum_{j=0}^{M} P_{ij} = 1 \qquad i = 0, 1, \ldots, M \quad 4.16$$

Las probabilidades de transición P_{ij} entre estados pueden ser representadas mediante una matríz cuadrada de la siguiente forma:

$$T = \begin{bmatrix} P_{00} & P_{01} & \ldots P_{0M} \\ P_{10} & P_{11} & \ldots P_{1M} \\ \vdots & \vdots & \vdots \\ P_{M0} & P_{M1} & \ldots P_{MM} \end{bmatrix} \quad 4.17$$

Para obtener el enésimo paso de la matriz de transición es necesario multiplicar la matriz por sí misma n veces, es decir:

$$T = T^n \quad 4.18$$

P_{ij} representa las probabilidades en estado estable S_i. El estado estable (steady-state) de la matríz de transición, puede ser obtenido a partir de lo siguiente [4]:

$$\bar{S} = \bar{S}T \quad 4.19$$

o, equivalentemente a:

$$(I - T')\bar{S} = 0 \qquad 4.20$$

Donde $\overline{S} = [s_1 \ s_2 \ \dots \ s_M]$, T' es la transpuesta de la matriz T e I la matriz identidad.

4.4 Comportamiento Microscópico y Macroscópico

En este trabajo, emplearemos una descripción de paquetes perdidos en un sistema VoIP basados en ventanas de tiempo angostas y amplias. Donde, el comportamiento de los paquetes perdidos en una ventana de tiempo angosta será llamada *Microscópica*, y el comportamiento de pérdida de paquetes en una ventana de tiempo amplia será llamado *Macroscópica* [5].

El *comportamiento microscópico* se entiende como al periodo de pérdida de paquetes observado en una ventana de tiempo W_1 de una traza de tráfico VoIP; donde este periodo de paquetes perdidos tiene un específico PLR_1 [5].

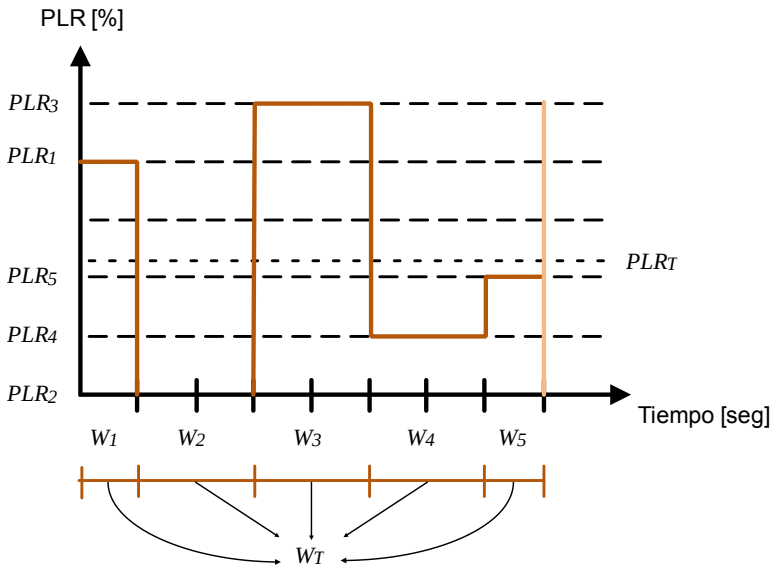


Figura 4.3 Comportamiento microscópico y macroscópico

Por otra parte, el *comportamiento macroscópico* se refiere al conjunto de periodos microscópicos $(W_1, W_2, W_3, \dots, W_n)$ observado en toda la traza de pérdida de paquetes de tráfico VoIP; en el que cada periodo microscópico tiene su propio PLR $(PLR_1, PLR_2, PLR_3, \dots, PLR_n)$ como se muestra en la Figura 4.3. En este comportamiento existen diferentes niveles de PLR para cada periodo microscópico, es decir, las pérdidas de paquetes no suceden de manera homogénea, sino que se concentran en algunos intervalos de tiempo (los paquetes se pierden por ráfagas). El PLR_T es calculado sobre el conjunto de trazas $(W_T = W_1 + W_2 + W_3 + W_4 + W_5)$ [5].

El *comportamiento microscópico* puede ser modelado por una cadena de Markov con pocos números de estados, mientras que el *comportamiento macroscópico* puede ser modelado por una cadena de Markov de varios estados [5]. Por lo tanto, usaremos las cadenas de Markov de 2 y 4 estados propuesto en la sección 4.1 para modelar los comportamientos microscópicos y macroscópicos, respectivamente.

Con el fin de simplificar esta descripción de paquetes perdidos, los periodos microscópicos pueden ser clasificados en dos conjuntos; uno para bajas pérdidas

y otra para altas pérdidas de paquetes. El umbral usado para delimitar los dos conjuntos, es una función de la percepción de la calidad, buena o mala, de acuerdo al cálculo de valores MOS.

4.5 Metodología para Simular Pérdida de Paquetes

Las actuales metodologías para simular paquetes perdidos consisten solamente en generar patrones de pérdidas por medio de las cadenas de Markov de diferentes estados [2,9-13]. La metodología propuesta para simular paquetes perdidos estará basada en dos etapas; primero se generara un patrón de perdida de paquetes y segundo se aplicará este patrón de pérdidas a una aplicación llamada *VoIPAS* [14].

La metodología está basada en los siguientes puntos.

1. La descripción de paquetes perdidos en VoIP estará basada en comportamientos microscópicos y macroscópicos, con el fin de representar diferentes niveles de *burst.*

2. Una cadena de Markov de dos estados será usada para representar los patrones de pérdidas.

3. Se creará un vector de pérdida de paquetes, para que sea introducido a una aplicación VoIP. Esto se obtiene por medio de un script en MatLab.

4. Para concluir, se pone en práctica esta metodología, sobre una aplicación VoIP, llamada VoIPAS. La emulación se realiza de la siguiente manera: Cuando la aplicación de voz encuentra un “uno” el

paquete no se envía, y cuando detecta un “cero” el paquete es enviado.

4.5.1 Metodología usando cadenas de Markov de 2 estados

Para incorporar periodos microscópicos con altos niveles de PLR, se generaron patrones de pérdidas con diferentes niveles de *burst* sobre una ventana de tiempo. Los patrones de pérdidas fueron generadas a partir de una cadena de Markov de dos estados, y está representada por una secuencia binaria:

$$P = \{P_t : t = 1, \dots, N\} \qquad 4.21$$

Donde:

- $P_t = 1$ quiere decir que un paquete se perdió
- $P_t = 0$ que un paquete llegó a su destino
- N es la longitud del patrón de pérdidas [5]

Por ejemplo, usando la ecuación 4.25 podríamos formar un patrón de pérdidas como sigue:

Patrón de pérdidas	0000011001010101101100000000000000000000000001000000000
Estados de la cadena de Marvok de 2 *estados*	**22222112212121211211222222222222222222222221222222222**

Los parámetros de entrada al modelo son el tamaño de la ventana de tiempo y las probabilidades de transición p y q.

Sea $X = \{X_t : t = 1, \dots, N\}$ una comunicación de voz con una duración de N paquetes.

Puesto que los paquetes se pierden en Internet a ráfagas, se aplicaran T diferentes patrones de pérdidas sobre una ventana de tiempo W_l^u de tamaño wN $(w = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1)$ sobre cada comunicación X para emular paquetes perdidos con varios niveles de rafagosidad (w).

$$W_l^u = \{X_l, X_{l+1}, \dots, X_u : l = 1,2, \dots, N - [wN] + 1 \quad u = l + [wN] - 1 \quad l < u\} \qquad 4.23$$

Donde X_l y X_u son el l-ésimo y u-ésimo elemento de la serie de tiempo X_t y representa la ventana de tiempo desde su comienzo hasta su fin.

Los patrones de paquetes perdidos usados en esta metodología son generados por el algoritmo mostrado en la Tabla 4.1.

FOR $\boldsymbol{i = 1}$ **TO** *N*
IF ($\boldsymbol{i >= 1}$ **AND** $\boldsymbol{i = u}$ **AND** *paquete perdido*) **THEN**
$\boldsymbol{P(i) = 1}$
ELSE
$\boldsymbol{P(i) = 0}$
END IF
END FOR

Tabla 4.1 Algoritmo para generar patrones de pérdida: cadena de Markov de dos estados

Referencias

[1] Alexander Raake, Speech Quality of VoIP: assessment and prediction, John Wiley & Sons Inc., 2006, ISBN: 0-470-03060-8

[2] Alexander Raake, “Short-and long-term packet loss behavior: towards speech quality prediction for arbitrary loss distributions”, IEEE Transactions on Audio, Speech, and Language Processing, vol. 14, no. 6, 2006, pp. 1957-1968.

[3] B. Kolman, D. R. Hill, Álgebra lineal, México, Pearson, 2006.

[4] Leopoldo Estrada, Deni Torres, Homero Torral; Analytical Description of a Parameter-based Optimization of the Quality of Service for VoIP Communications, WSEAS Transactions on Communications, Vol. 9, No. 9, 2009, pp. 1042-1052

[5] Homero Torral, Modelado de los Parámetros de QoS de Tráfico VoIP Auto-similar y una Mejora al Modelo E, Tesis de Doctorado, CINVESTAV Unidad Guadalajara, 2010.

[6] Rafael Ubal Tena, Estudio y evaluación de técnicas FEC para la recuperación frente a errores, 06 Mayo 2013, http://www.ece.neu.edu/~ubal/index.html

[7] Fayez Gebali, Analysis of Computer and Communication Networks, Springer, 2008, ISBN 978-0-387-74436-0

[8] Sheldon Ross, A First Course in Probability, Pearson, 2012, ISBN: 032179477X

[9] Y.C. Su, C.S. Yang and C.W. Lee, “The analysis of packet loss prediction for Gilbert model with loss rate uplink”, Information Processing Letters, vol. 90, no. 3, 2004, pp. 155159.

[10] H. Lee, “A Packet-Loss Recovery Scheme Based on the Gap Statistics”, Information networking: convergence in broadband and mobile networking, vol. 3391, LNCS

Springer, 2005, pp. 627-634.

[11] M. Yajnik, S.B. Moon, J. Kurose and D. Towsley, “Measurement and modeling of the temporal dependence in packet loss”, Proc. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), IEEE, New York, NY , USA, 21-25 March, 1999, pp. 345-352.

[12] T. Tao, J. Lu, K. Gong and J. Gu, “A Four-States Markov Model for Burst Error Analysis in Satellite Communications,” Proc. International Conference on Communication Technology (WCC-ICCT), Beijing , China, 21 - 25 august, 2000, pp. 930-934.

[13] Y. Yu and S.L. Miller, “A four-state Markov frame error model for the wireless physical layer,” Proc. IEEE Wireless Communications and Networking Conference (WCNC), IEEE, Kowloon, Hong Kong, 11-15 March, 2007, pp. 2053-2057.

[14] Jesús Argaez, Software para el análisis de QoS en VoIP, Tesis de Maestria, CINVESTAV Unidad Guadalajara, Julio 2009.

CAPÍTULO

5

5 EMULACIÓN DE PLR SOBRE UNA APLICACIÓN VOIP Y MEDICIÓN DE JITTER

En este capítulo, se aborda el análisis de las principales métricas de desempeño medidas bajo diferentes escenarios de red. Cada escenario tendrá distintos niveles de PLR desde 0.1% hasta 5% y con diferentes niveles de rafagosidad. Si hablamos de una comunicación real se sabe que es imposible obtener un comportamiento de la red con ciertos niveles de PLR deseados. Imaginemos una red, en la cual se desa obtener en una comunicación de 10 minutos con un PLRde 2%; estamos consientes de que esto podría tardar mucho tiempo para que se presentara esta situación o posiblemente no pasaría nunca. Sin embargo, es posible emular diferentes escenarios de red con diversos niveles de PLR deseados por medio del modelo descrito en el capítulo 4.

5.1 Creación de vectores

Para la creación de vectores de PLR, nos basaremos en el algoritmo propuesto en el capítulo 4 que corresponde a la Tabla 4.1. Cada vector tendrá la longitud correspondiente a una llamada de 10 minutos. En este trabajo se utilizará el CODEC G.711, ya que es el CODEC más utilizado en los servicios de telefonía.

El número de paquetes creados en una comunicación de 10 minutos de duración, depende del tamaño de paquete utilizado; es decir, se obtendrá un mayor número de paquetes cuando se escoge un tamaño de $20ms$ que un tamaño de $40\ ms$ o $60\ ms$. Para este trabajo utilizaremos dos tamaños de paquetes: $20\ ms$ y $40\ ms$.

Los vectores creados serán introducidos a la aplicación VoIPAS v1.5 [8]; donde cada elemento del vector corresponde a un paquete de voz generado por

la aplicación, cuando se presenta un "1" no se envía el paquete generado (paquete perdido), pero si se presenta un "0" el paquete es transmitido a la red para que alcance su destino (paquete recibido exitosamente). En la Figura 5.1, se visualiza la interfaz de usuario principal de la aplicación VoIPAS v1.5 utilizada.

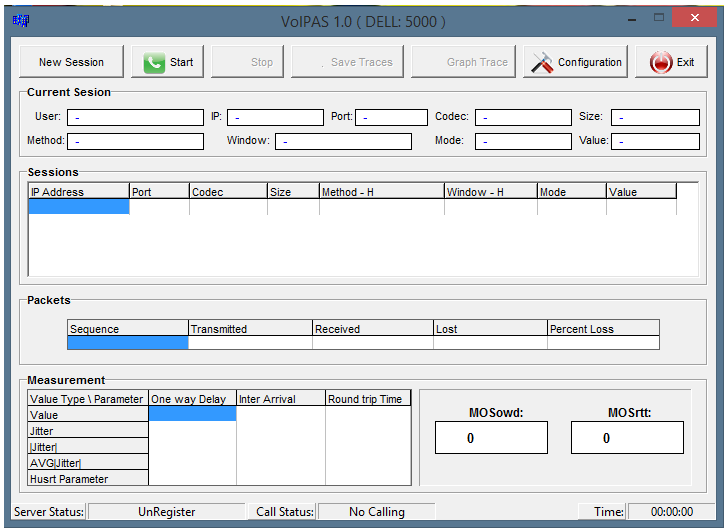


Figura 5.1 Interfaz VoIPAS v1.5

Con la metodología explicada con anterioridad, podemos generar los vectores de pérdida, con diferentes niveles de PLR, según su ventana de tiempo W_l^u de la siguiente manera:

Sea W el vector que contiene los diferentes tamaños de ventana, sobre las cuales, se insertarán vectores de pérdida P con diversos niveles de PLR mediante una cadena de Markov de 2 estados:

$$W = wN$$

Donde $w = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$, N es el número de paquetes generados en las llamadas de prueba, $P = \{P_t : t = 1, \dots, wN\}$ (si $P_t = 1 \Rightarrow Paquete\ Perdido$ y si $P_t = 0 \Rightarrow Paquete\ Recibido$) y $PLR = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0\}$ es el vector que contiene los diferentes valores de pérdida que contendrán los vectores P.

Cada vector P tomará la posición W_l^u (ver ecuación 4.27) dentro de la comunicación $Y = \{Y_n : n = 1, \dots, N\}$ que contiene N paquetes, donde Y_n representa el $n - esimo$ paquete en una comunicación.

5.2 Escenario de medición

En las Figuras 5.2 a) y 5.2 b) se muestran los escenarios de medición, tanto el real así como el emulado, respectivamente. Cabe mencionar, que el escenario real consta de dos computadoras (A y B), las cuales se encuentran conectadas únicamente por un cable Ethernet, obteniendo así una configuración punto a punto; en la cual se sabe no habría pérdida de paquetes adicionales a las emuladas.

Para generar un conjunto de trazas de tráfico VoIP, se estableció un conjunto de llamadas de prueba mediante la aplicación VoIPAS y para lograr emular el escenario de la Figura 1 b), bajo diferentes condiciones de pérdida de paquetes, se aplicó la metodología presentada en la sección 5.1.

A continuación se describen las características de los equipos usados en el escenario de medición:

- La computadora A tiene instalado Windows XP Professional, con una IP 192.168.1.141. En esta PC se introducen los vectores de pérdida mediante la aplicación VoIPAS, tal y como se mencionó anteriormente, dicha aplicación va a leer cada elemento del vector de pérdida a la misma taza que transmite los paquetes de voz en la llamada de prueba y enviará el paquete de voz si encuentra un 0, o simplemente no lo enviará si encuentra un 1.
- La computadora B tiene instalado Windows XP Professional, con una IP 192.168.1.140. En esta PC, se instaló de igual manera la aplicación VoIPAS para que pueda tener comunicación con la computadora A; y para capturar

los patrones de tráfico (pérdidas) generados durante la comunicación entre A y B, se utilizaron dos analizadores, los cuales son el Wireshark (analizador de protocolos) y CommView (analizador de MOS y factor R).

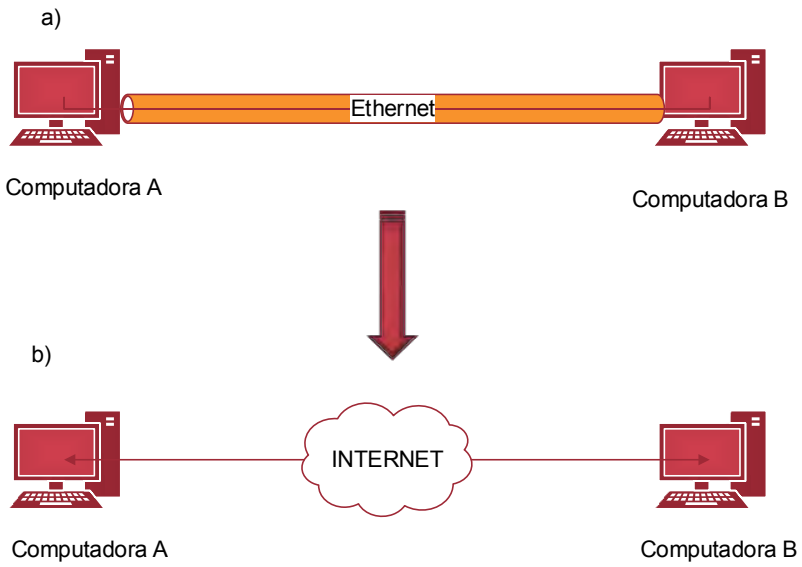


Figura 5.215 a) Escenario de medición real; b) Escenario de medición emulado

5.3 Mediciones

En este apartado, se describen los pasos que se siguieron para la realización de la medición de los patrones de tráfico, para dos tamaños de paquetes ($20ms$ y $40ms$).

En primer punto, se configura la computadora A. En esta se ejecuta la aplicación VoIPAS, luego se crea una nueva sesión dando click a la opción *New Session*, como se muestra en la Figura 5.3.

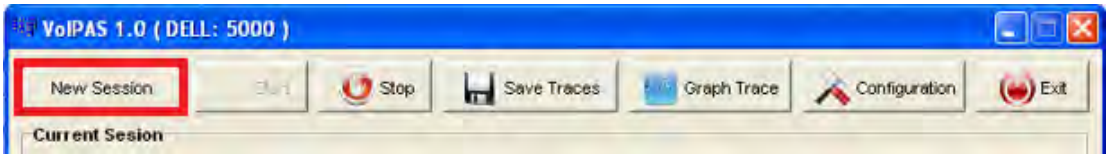


Figura 5.3 Creación de una nueva sesión

Tal y como se muestra en la Figura 5.4, se configura la IP de la computadora destino (192.168.1.140); el puerto se deja en 5001; se selecciona el tipo de códec (G711 A Law) con un tamaño de paquete ($20\ ms$); se carga el vector de pérdida con la opción *Load* y las demás opciones no se modifican; click en *ok* para iniciar la comunicación.

Figura 5.4 Configuración de una sesión

En segundo punto, se configura la computadora B. En esta se ejecuta la aplicación VoIPAS, y no se le configura absolutamente nada, ya que esta PC funcionará como receptor.

Una vez que emisor y receptor estén configurados (A y B), en la aplicación de VoIPAS de la computadora A, se le da click en *Start* para que inicie la comunicación entre ambas computadoras, como se muestra en la Figura 5.5.

Figura 5.5 Iniciar comunicación

Seguidamente, se abre Wireshark para capturar el tráfico, se selecciona la interfaz de red que se utiliza y luego click en *Start* para iniciar la captura, como se muestra en la Figura 5.6.

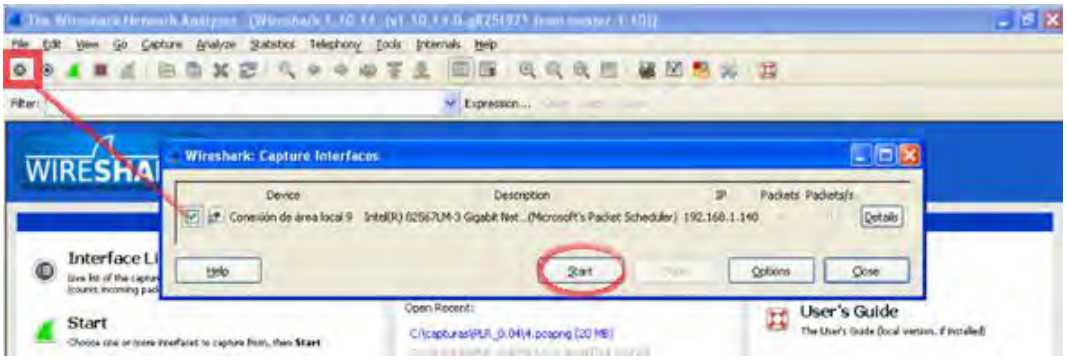


Figura 5.6 Selección de la interfaz de captura

Una vez establecida la comunicación entre A y B, se inicia de forma paralela la captura mediante el Wireshark en B, tal y como se muestra en la Figura 5.7.

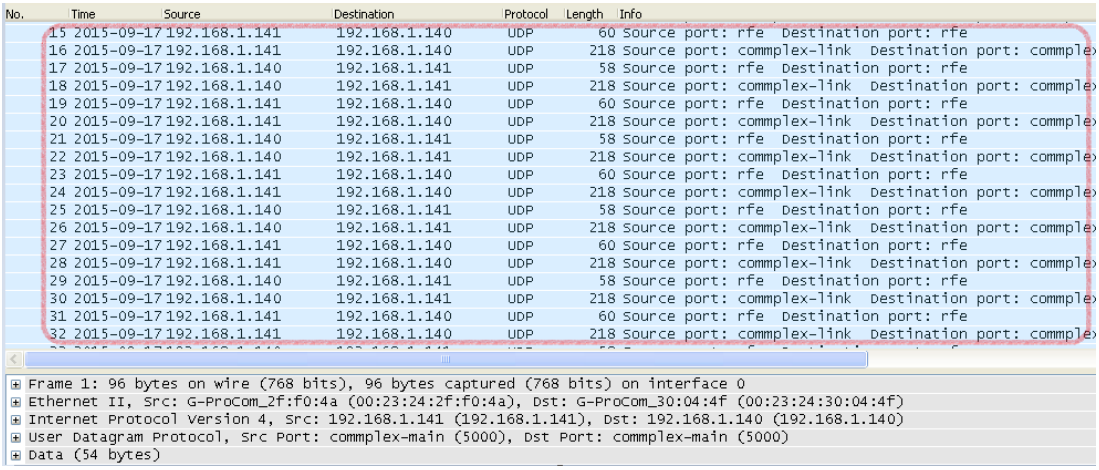


Figura 5.7 Captura del tráfico

Para este trabajo, las llamadas de pruebas tuvieron una duración de 10 minutos. Al finalizar el tiempo establecido (o recibidos los 30,000 paquetes o en el otra caso recibidos los 15,000paquetes en B), se procede a identificar los flujos RTP entre A (192.168.1.141) y B (192.168.1.140), como se observa en la Figura 5.8.

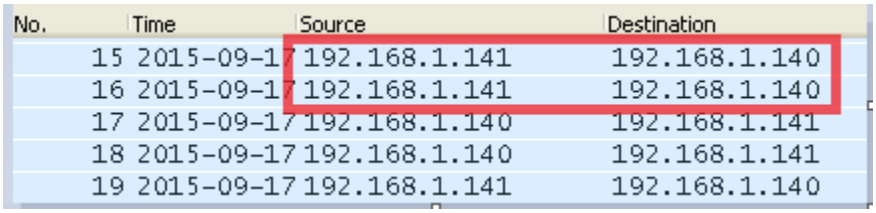


Figura 5.816 Selección de flujo

En la Figura 5.7 se puede observar que sólo se captura flujos de datagramas UDP, es decir, para poder visualizar los flujos RTP, se realiza la decodificación de dichos datagramas; para lograrlo se selecciona un flujo UDP, click derecho y seleccionamos la opción de *Decode As…* y posteriormente *RTP*, click en *ok* para finalizar la decodificación, como se muestra en la Figura 5.9.

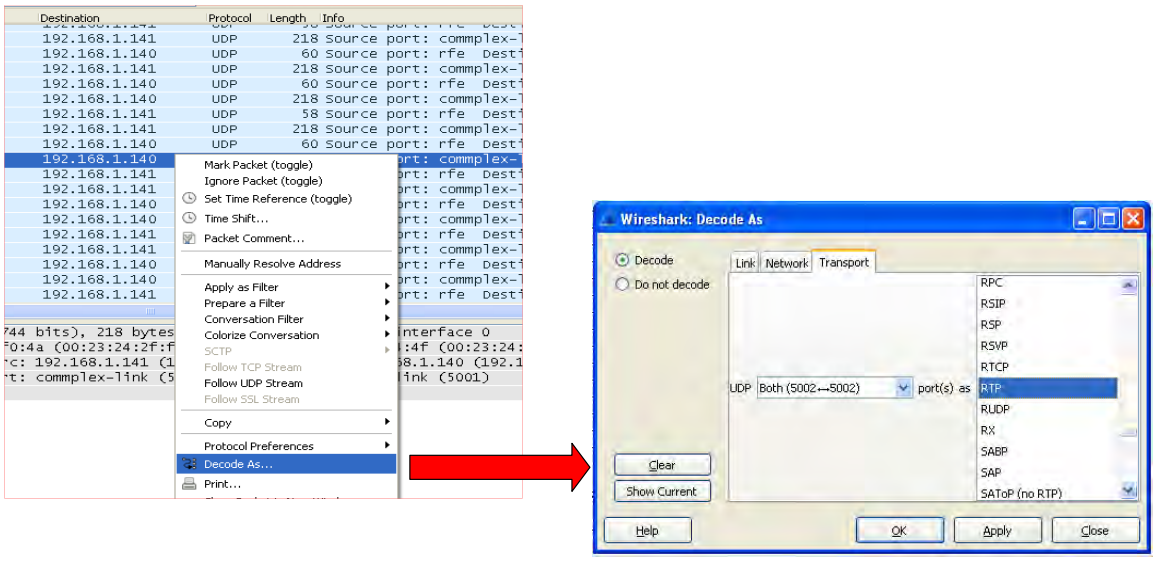


Figura 5.9 Decodificación de los flujos UDP

Posteriormente, ya que se decodificaron los datagramas UDP, seleccionamos la pestaña *Telephony*, click en la opción *RTP* y luego click en *Show All Streams* para visualizar los flujos RTP, como se muestra en la Figura 5.10.

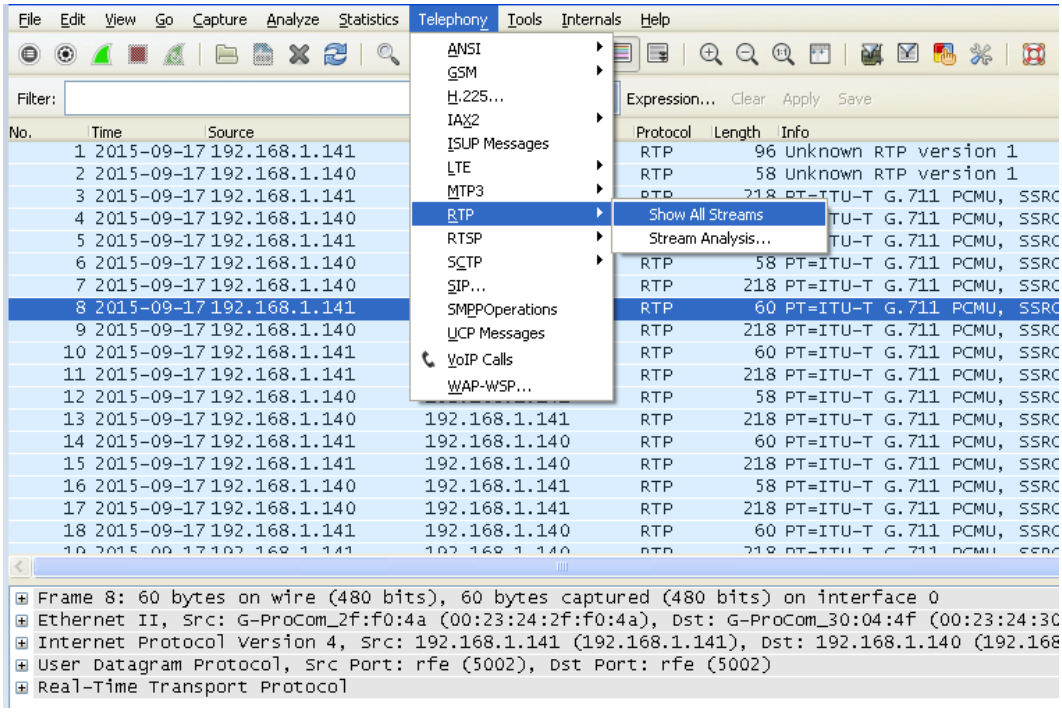


Figura 5.10 Visualización de los flujos RTP

Al finalizar el punto anterior, se verifica que el porcentaje de los paquetes perdidos medidos durante la comunicación, coincida con el porcentaje de paquetes perdidos (PLR) del vector de pérdida que se introdujo en la aplicación VoIPAS (es decir, el PLR simulado); por ejemplo, en esta ocasión se requiere que el vector pierda 296 paquetes (296 unos en el vector de pérdidas), que equivale al 1% del total de paquetes transmitidos ($30{,}000$ paquetes) de la ventana del $10\%(w = 0.1)$. En la Figura 5.11, se puede observar que se han perdido 296 paquetes con un porcentaje de 1.0%, es decir, que no hubo más paquetes perdidos de lo esperado, como se mencionó en un principio, en una configuración punto a punto no hay pérdidas; por lo tanto, con esta prueba se verifica que en dicha red solo se presentan las pérdidas emuladas.

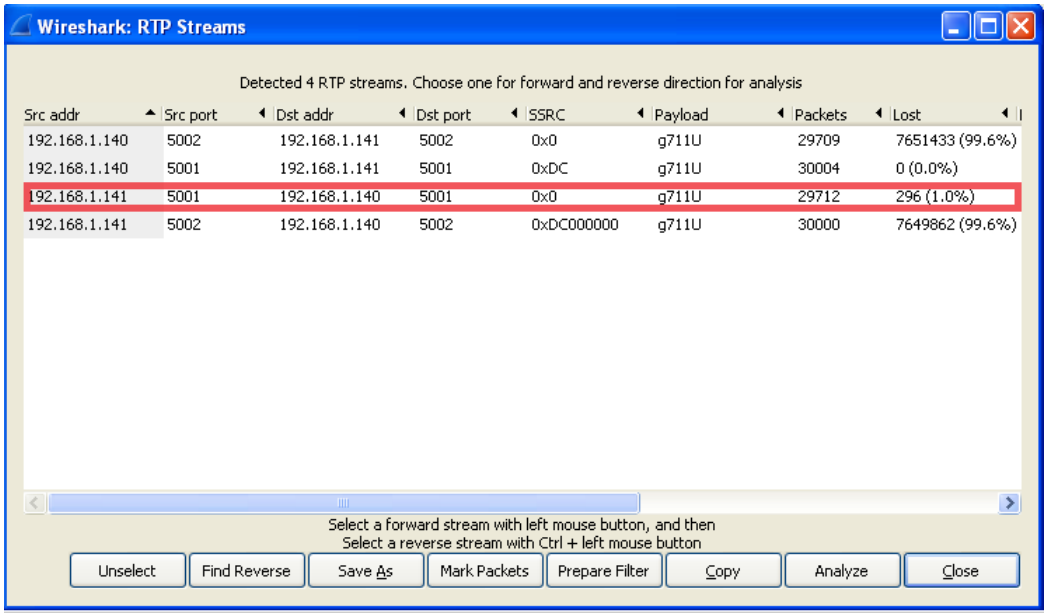


Figura 5.11 Resultados de los flujos RTP

Este procedimiento se realizó para los diferentes niveles de pérdida $PLR = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.1, 1.2, 1.3, 1.4, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5\}$ que contendrán los vectores introducidos a la aplicación VoIPAS. Así mismo, para cada tamaño de ventana, es decir, 10 ventanas por tamaño de paquete ($20ms$ y $40ms$); por lo tanto, se realiza para 440 combinaciones.

NOTA: Para el tamaño de paquete 40, se realiza el mismo procedimiento desde un principio, en lo único que cambia es cuando se configura el *Size Packet*, se selecciona $40\ ms$, como se muestra en la Figura 5.12.

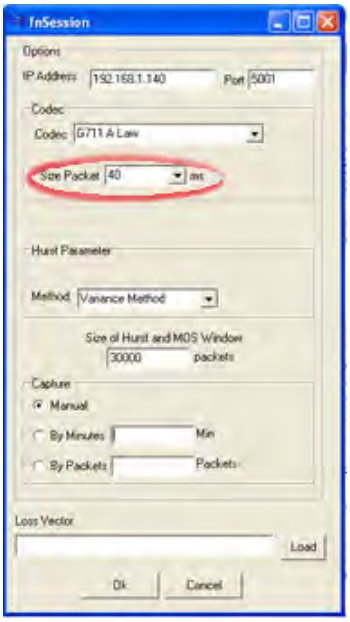


Figura 5.1217 Configuración de una sesión, para el tamaño de paquete "40 ms"

Otro software que se utilizó es el CommView, este analiza y captura cada paquete para mostrar información importante, incluye un analizador de VoIP para el análisis en profundidad. Para ello, es necesario abrir CommView, se selecciona la red a utilizar (*Conexión de área local 9*), y posteriormente se le da click en *iniciar captura*, como se observa en la Figura 5.13.

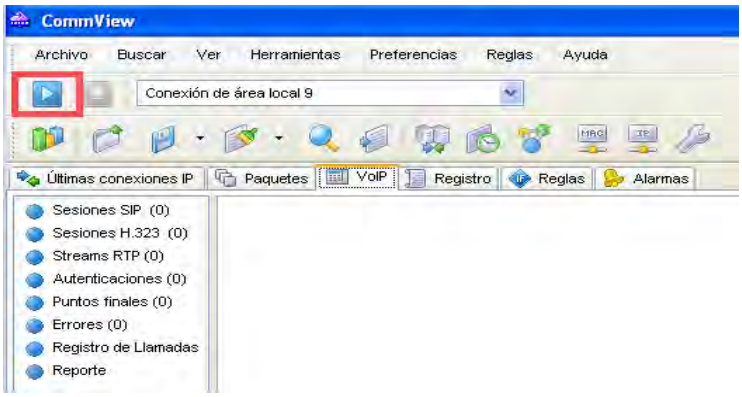


Figura 5.13 Selección de la interfaz de captura

Una vez establecida la comunicación entre A y B, seleccionamos el apartado de *VoIP*, click en *Streams RTP*, en la Figura 5.14 se observa como se capturan los paquetes durante la conexión.

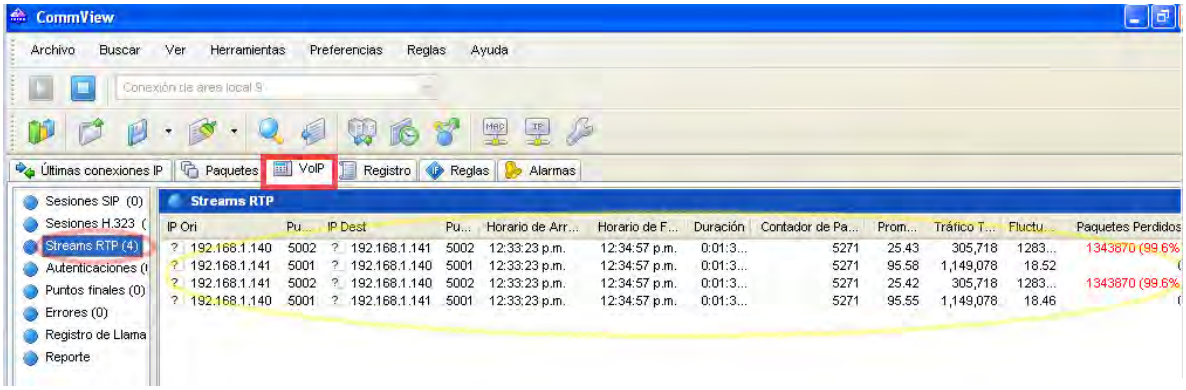


Figura 5.14 Captura del tráfico

Como se mencionó anteriormente, las llamadas de pruebas tuvieron una duración de 10 minutos. Con CommView no es necesario decodificar el flujo, ya que el mismo software se encarga de ello. Al finalizar el tiempo establecido (o recibido los $30{,}000$ o los $15{,}000$ paquetes en B), se procede a identificar los flujos RTP entre A (192.168.1.141) y B (192.168.1.140). Se verifica que el porcentaje de los paquetes perdidos medidos durante la comunicación, corresponda al porcentaje de paquetes perdidos del vector de pérdida (PLR) introducido en VoIPAS. Por ejemplo, en este caso se requiere que el vector pierda 32 paquetes (32 *unos* en el vector de pérdida), que equivale al 0.1% del total de paquetes transmitidos de la ventana del 40%. En la figura 5.15, se observa que se han perdido 32 paquetes con un porcentaje de 0.1%, es decir, que una vez más se comprueba que en una configuración punto a punto no se pierden paquetes, y sólo se pierden los requeridos por el vector mediante la aplicación VoIPAS.

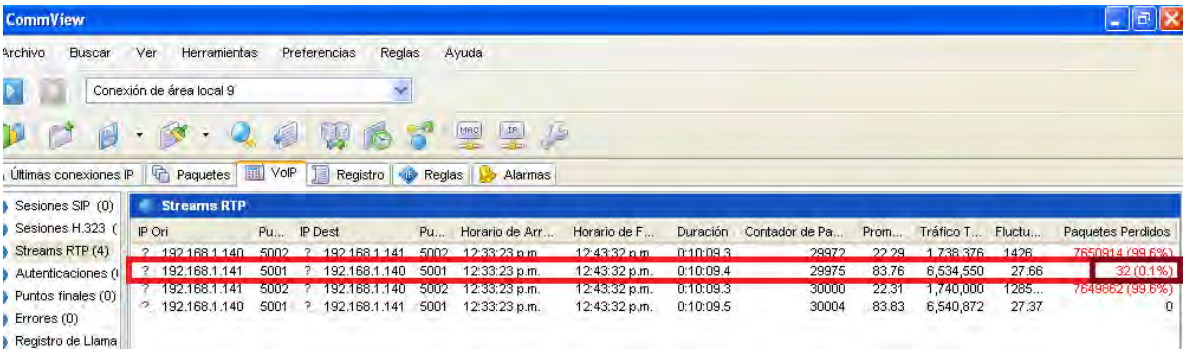


Figura 5.1518 Resultados de los flujos RTP

Este procedimiento se efectúa para todos los niveles de pérdida $PLR = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.1, 1.2, 1.3, 1.4, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5\}$ que contienen los vectores que se introducen a la aplicación VoIPAS. Del mismo modo, para cada tamaño de ventana, es decir, 440 combinaciones en total.

NOTA: Para el tamaño de paquete 40, se realiza el mismo procedimiento desde un principio, la única diferencia es cuando se configura el *Size Packet*, se selecciona $40\ ms$, como se muestra en la Figura 5.12.

5.4 Análisis de PLR

En la Figura 5.16, se muestran de manera gráfica los vectores de pérdida introducidos a la aplicación VoIPAS para los diferentes tamaños de ventana $W = wN$ (donde $w = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$) y para dos elementos del vector PLR (elementos $PLR = 0.1\%$ y $PLR = 5.0\%$). En la figura se observa la distribución de los niveles de pérdida de paquetes sobre diferentes ventanas de tiempo, los cuales muestran diferentes niveles de rafagosidad presentados en diversos escenarios de red.

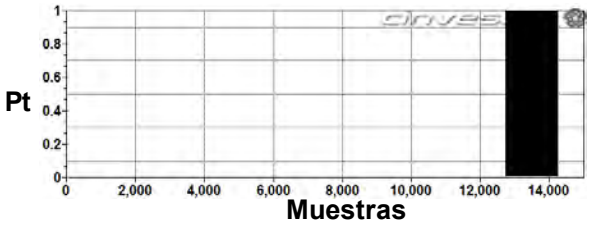
La Tabla 5.1, muestra los diferentes tamaños de ventana con sus respectivos porcetajes de pérdidas que corresponden a la Figura 5.16.

	W	*PLR*		*W*	*PLR*
a)	10 %	0.1 %	***k)***	10 %	5.0 %
b)	20 %	0.1 %	***l)***	20 %	5.0 %
c)	30 %	0.1 %	***m)***	30 %	5.0 %
d)	40 %	0.1 %	***n)***	40 %	5.0 %
e)	50 %	0.1 %	***o)***	50 %	5.0 %
f)	60 %	0.1 %	***p)***	60 %	5.0 %
g)	70 %	0.1 %	***q)***	70 %	5.0 %
h)	80 %	0.1 %	***r)***	80 %	5.0 %
i)	90 %	0.1 %	***s)***	90 %	5.0 %
j)	100 %	0.1 %	***t)***	100 %	5.0 %

Tabla 5.1 Tamaños de ventana y porcentajes de pérdida de la Figura 5.16

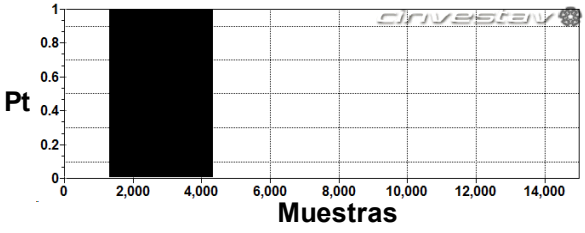
En la Figura 5.16 a), se puede observar que el nivel de rafagosidad es mucho mayor que en la figuras 5.26 b), c),…j), bajo el mismo nivel de pérdida de paquetes (0.1%). Lo mismo ocurre con el porcentaje más alto, el del 5%. El nivel de rafagosidad se distribuye en función del tamaño de ventana, es decir, que en la ventana del 10%, el nivel será mayor que en la ventana del 100%, usando el mismo porcentaje de pérdida.

a)

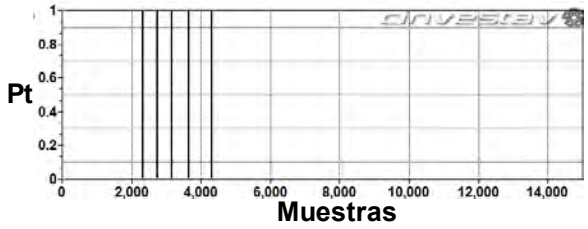


k)

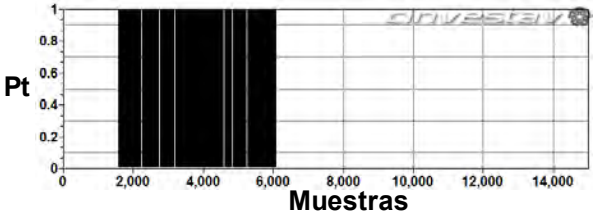
b)



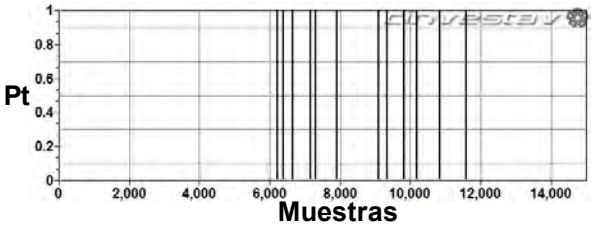
l)



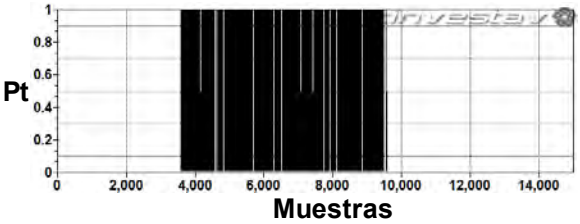
c)



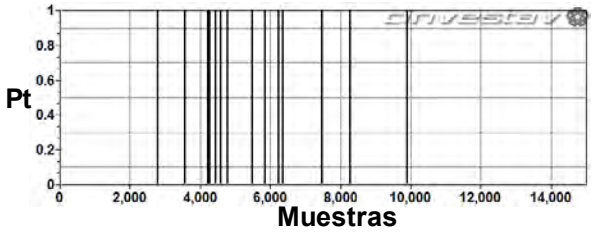
m)



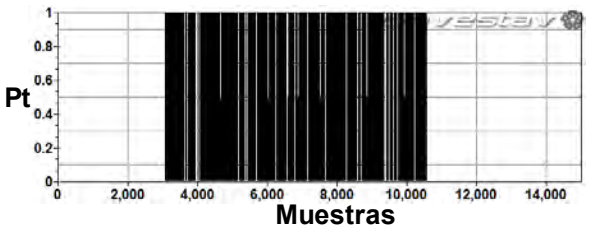
d)



n)

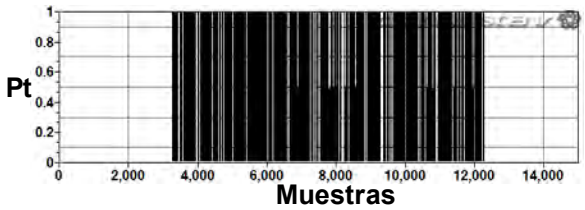


e)

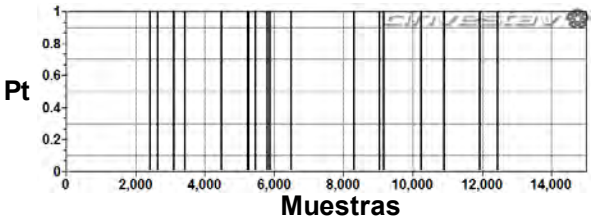


o)

f)

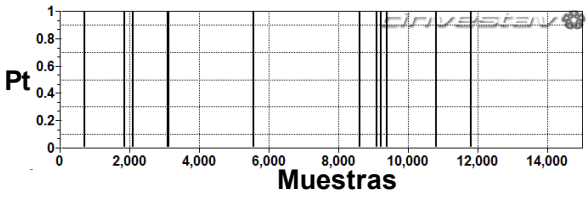


p)

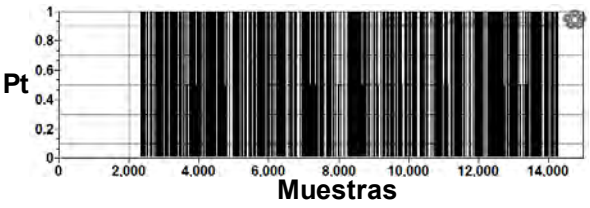


g)

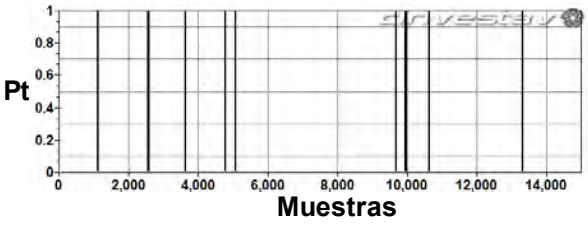
q)



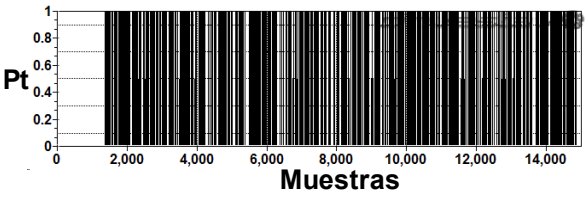
h)



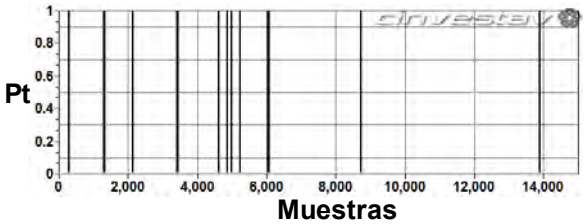
r)



i)



s)



j)

t)

Figura 5.16 Resultados de análisis de PRL

5.5 Análisis de MOS

La Tabla 5.2 muestra los valores de MOS obtenidos para cada una de las 220 mediciones realizadas correspondientes al tamaño de paquete de $20ms$. Así mismo, en la Tabla 5.3 se muestran los valores del tamaño de paquete de $40ms$. En total, se realizaron 440 mediciones.

En las Figuras 5.17 y 5.18 se pueden observar la superposición de las diferentes familias de curva de MOS vs PLR para diferentes tamaños de ventanas. Por otra parte, se analiza que a medida que aumenta el PLR $(0.1\% \leq PLR \leq 5\%)$, decrementa la calidad de servicio en la comunicación $(3.800 \leq MOS \leq 4.364)$ para el tamaño de paquete $20ms$ y para el tamaño de paquete $40ms$ $(3.840 \leq MOS \leq 4.406)$. Sin embargo, también se observa que el nivel de rafagosidad tiene efecto nulo en la QoS que percibe el usuario final. Por lo tanto, se llega a la conclusión que el MOS no considera los efectos de las pérdidas a ráfagas.

w/PLR	0.1%	0.2%	0.3%	0.4%	0.5%	0.6%	0.7%	0.8%	0.9%	1.0%	1.1%	1.2%	1.3%	1.4%	1.5%	2.0%	2.5%	3.0%	3.5%	4.0%	4.5%	5.0%
0.1	4.361	4.350	4.341	4.330	4.324	4.307	4.301	4.281	4.276	4.269	4.256	4.238	4.225	4.222	4.203	4.158	4.095	4.048	3.970	3.916	3.905	3.805
0.2	4.364	4.352	4.341	4.322	4.319	4.312	4.299	4.288	4.271	4.258	4.254	4.244	4.231	4.213	4.203	4.141	4.094	4.026	3.967	3.930	3.928	3.800
0.3	4.359	4.347	4.336	4.334	4.320	4.304	4.292	4.288	4.270	4.266	4.260	4.243	4.232	4.221	4.215	4.158	4.0992	4.032	3.978	3.922	3.915	3.801
0.4	4.360	4.349	4.342	4.332	4.317	4.304	4.294	4.284	4.281	4.265	4.252	4.247	4.227	4.220	4.203	4.147	4.096	4.038	3.975	3.923	3.906	3.808
0.5	4.359	4.350	4.344	4.326	4.312	4.303	4.299	4.290	4.276	4.257	4.260	4.244	4.234	4.215	4.208	4.155	4.092	4.036	3.960	3.914	3.908	3.801
0.6	4.357	4.354	4.346	4.331	4.312	4.311	4.299	4.282	4.272	4.266	4.253	4.237	4.235	4.223	4.212	4.148	4.091	4.038	3.976	3.919	3.902	3.812
0.7	4.360	4.349	4.344	4.324	4.324	4.307	4.297	4.288	4.274	4.256	4.251	4.237	4.227	4.213	4.206	4.151	4.097	4.032	3.975	3.914	3.914	3.806
0.8	4.363	4.350	4.335	4.330	4.314	4.315	4.299	4.290	4.279	4.265	4.256	4.246	4.226	4.226	4.202	4.151	4.097	4.030	3.980	3.923	3.916	3.810
0.9	4.357	4.350	4.340	4.324	4.316	4.312	4.291	4.278	4.273	4.257	4.259	4.242	4.232	4.224	4.203	4.145	4.093	4.023	3.972	3.906	3.900	3.814
1.0	4.363	4.353	4.343	4.330	4.315	4.309	4.301	4.288	4.282	4.267	4.252	4.243	4.231	4.214	4.203	4.150	4.090	4.041	3.972	3.916	3.912	3.803

Tabla 5.2 Resultados de prueba MOS (tamaño de paquete 20ms)

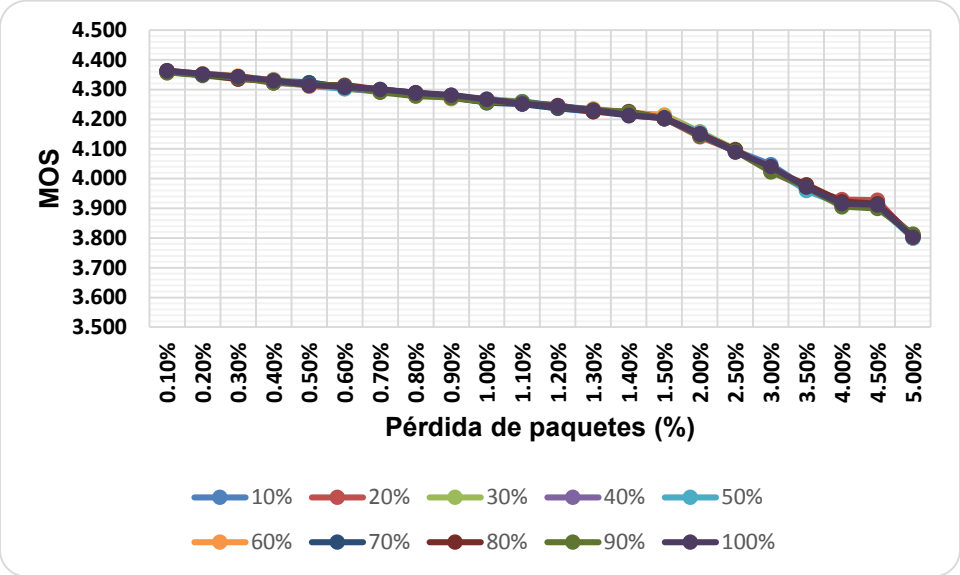


Figura 5.17 Resultados de prueba de MOS (tamaño de paquete 20ms)

w/PLR	0.1%	0.2%	0.3%	0.4%	0.5%	0.6%	0.7%	0.8%	0.9%	1.0%	1.1%	1.2%	1.3%	1.4%	1.5%	2.0%	2.5%	3.0%	3.5%	4.0%	4.5%	5.0%
0.1	4.394	4.393	4.390	4.377	4.369	4.367	4.335	4.329	4.308	4.307	4.303	4.300	4.292	4.268	4.266	4.204	4.156	4.099	4.047	3.982	3.936	3.881
0.2	4.396	4.385	4.385	4.378	4.367	4.356	4.343	4.336	4.326	4.323	4.314	4.275	4.272	4.264	4.250	4.216	4.160	4.101	4.029	3.971	3.940	3.890
0.3	4.406	4.389	4.387	4.375	4.373	4.341	4.338	4.332	4.317	4.316	4.306	4.294	4.285	4.266	4.262	4.198	4.134	4.114	4.045	3.987	3.915	3.871
0.4	4.398	4.389	4.387	4.369	4.361	4.356	4.345	4.333	4.327	4.322	4.304	4.292	4.276	4.260	4.242	4.212	4.129	4.092	4.030	3.965	3.941	3.851
0.5	4.398	4.394	4.379	4.376	4.352	4.352	4.332	4.332	4.321	4.312	4.300	4.289	4.271	4.253	4.240	4.192	4.143	4.121	4.059	4.001	3.931	3.840
0.6	4.397	4.386	4.375	4.368	4.347	4.346	4.336	4.336	4.334	4.306	4.300	4.277	4.268	4.262	4.262	4.200	4.188	4.108	4.037	3.975	3.894	3.887
0.7	4.395	4.388	4.384	4.376	4.367	4.367	4.347	4.326	4.317	4.309	4.307	4.293	4.274	4.263	4.230	4.218	4.151	4.105	4.036	3.979	3.916	3.891
0.8	4.400	4.391	4.382	4.376	4.367	4.366	4.341	4.339	4.334	4.301	4.301	4.295	4.276	4.264	4.250	4.197	4.144	4.113	4.023	3.975	3.923	3.902
0.9	4.403	4.381	4.379	4.367	4.362	4.344	4.342	4.332	4.316	4.314	4.293	4.285	4.282	4.265	4.250	4.200	4.150	4.051	4.016	3.997	3.936	3.853
1.0	4.399	4.386	4.385	4.376	4.369	4.361	4.335	4.334	4.325	4.321	4.307	4.304	4.289	4.263	4.223	4.204	4.157	4.074	4.052	4.004	3.941	3.863

Tabla 5.3 Resultados de prueba MOS (tamaño de paquete 40ms)

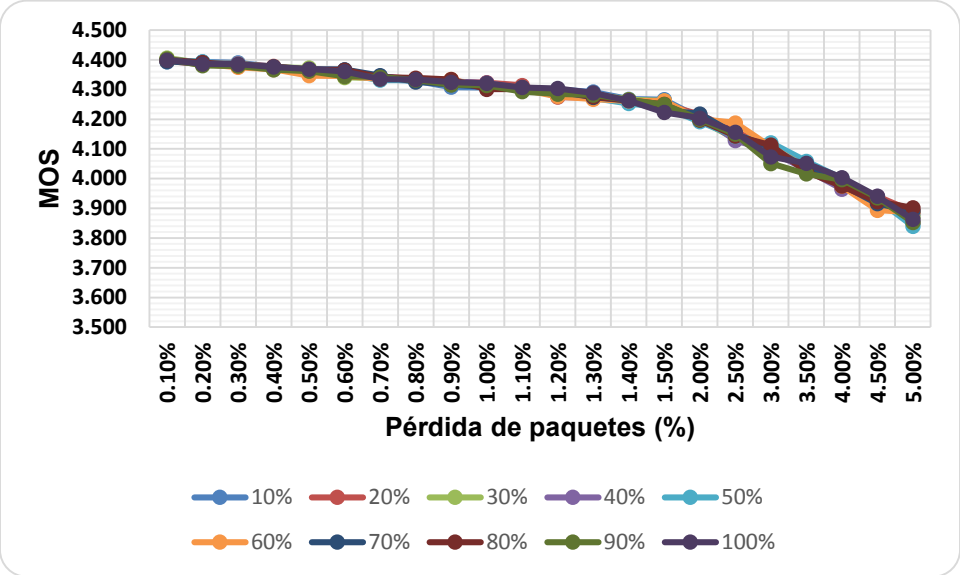
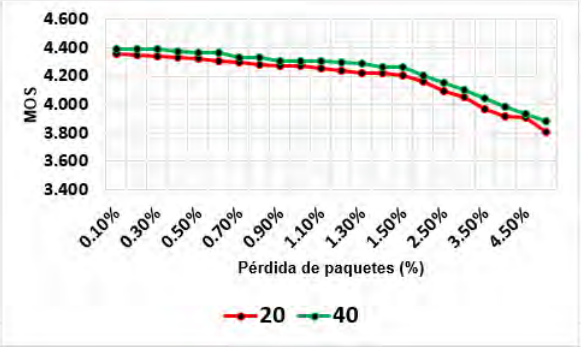


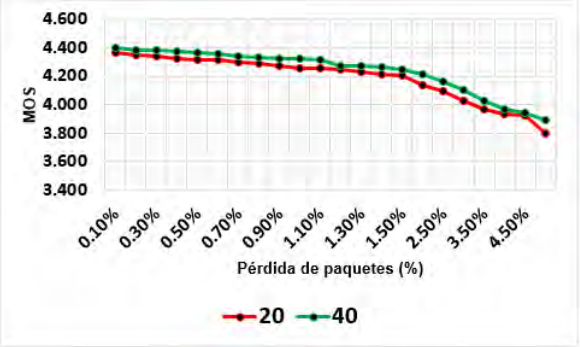
Figura 5.18 Resultado de prueba de MOS (tamaño de paquete 40ms)

5.5.1 Comparación de MOS entre tamaño de paquetes (20ms y 40ms) por ventana

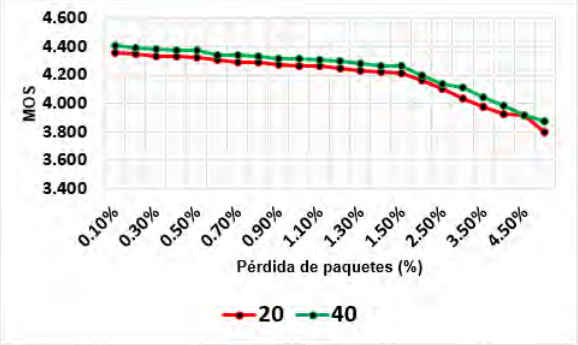
En la Figura 5.19, se ilustra una comparación de los valores de MOS entre los dos tamaños de paquetes ($20ms$ y $40ms$) por ventana. Como se puede observar en la Figura 5.19, identificamos que los valores de MOS correspondientes al tamaño de paquete de $40ms$ en todas las ventanas está por encima que los valores de MOS correspondiente al del tamaño de paquete de $20ms$, esto significa que la calidad de servicio es ligeramente mayor en las comunicaciones con tamaño de paquete de $40ms$ que con las de $20ms$.



a) W=10%

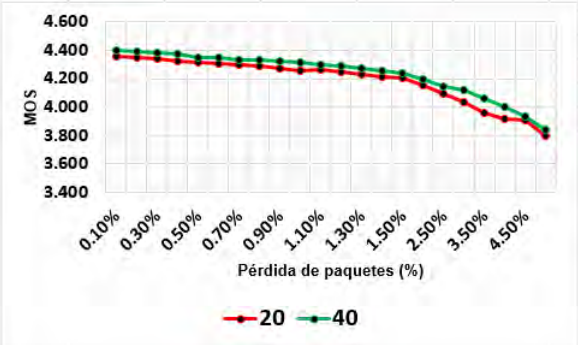


b) W=20%

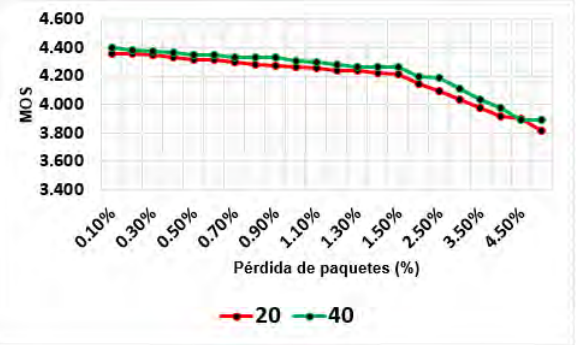


c) W=30%

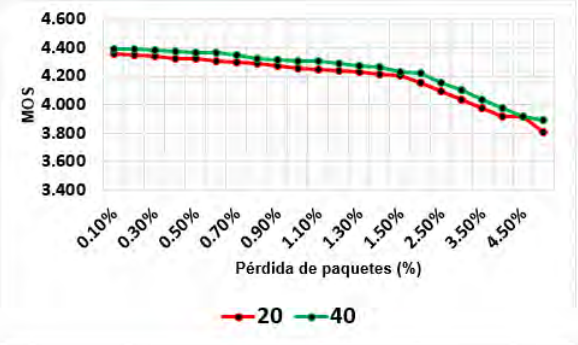
d) W=40%



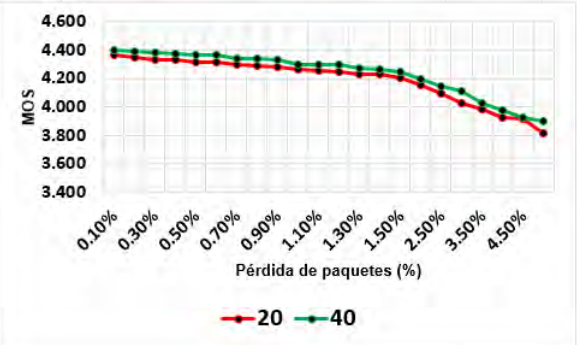
e) W=50%



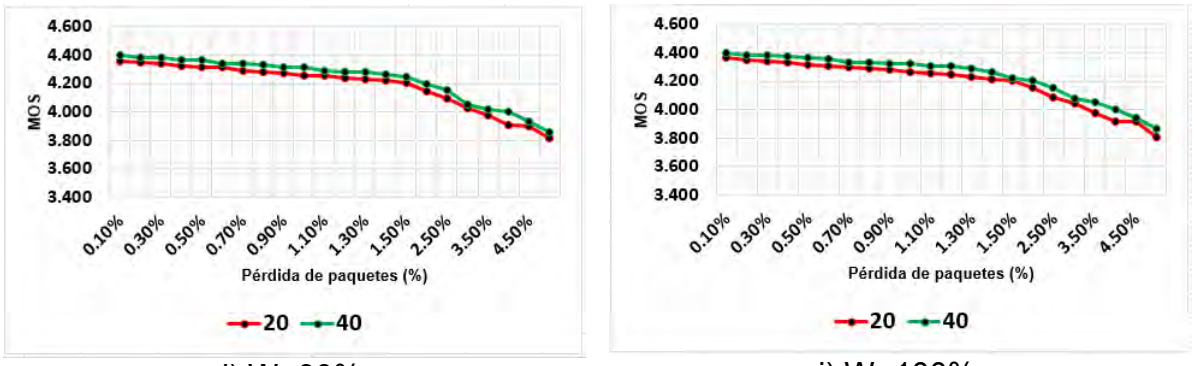
f) W=60%



g) W=70%



h) W= 80%



i) W=90% j) W=100%

Figura 5.19 Comparación de MOS por ventana

5.6 Análisis del Parámetro H

Otro análisis realizado en el presente trabajo fue evaluar el parámetro de Hurst a las trazas de jitter de las 440 mediciones. Para dicho análisis fue necesario procesar las trazas de jitter de los archivos .pcap que resultaron de las mediciones. Como primer paso, se extrajeron los archivos *CSV* correspondientes a cada medición. Esto se realiza con los pasos descritos en la sección 5.3, al obtener los resultados de los flujos RTP, se selecciona el flujo (192.168.1.141 a la 192.168.1.140, puerto 5001) y le damos click en la opción *Analize*, como se muestra en la Figura 5.20.

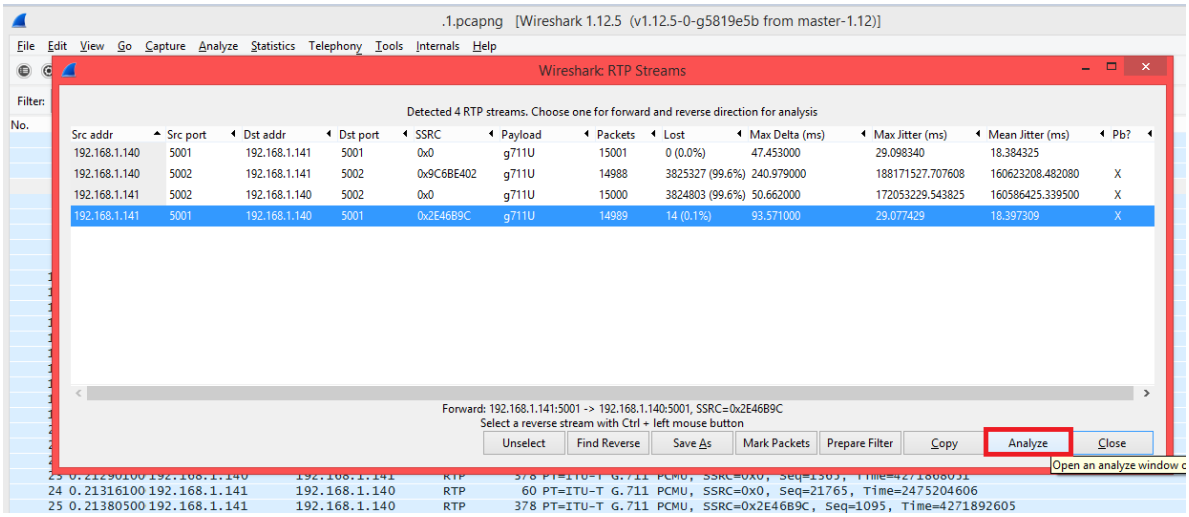


Figura 5.2019 Analizador del flujo

Seguidamente aparece otro cuadro, el cual se trata de un análisis completo sobre el flujo seleccionado; en este le damos click en *Save as CSV…* como se muestra en la Figura 5.21.

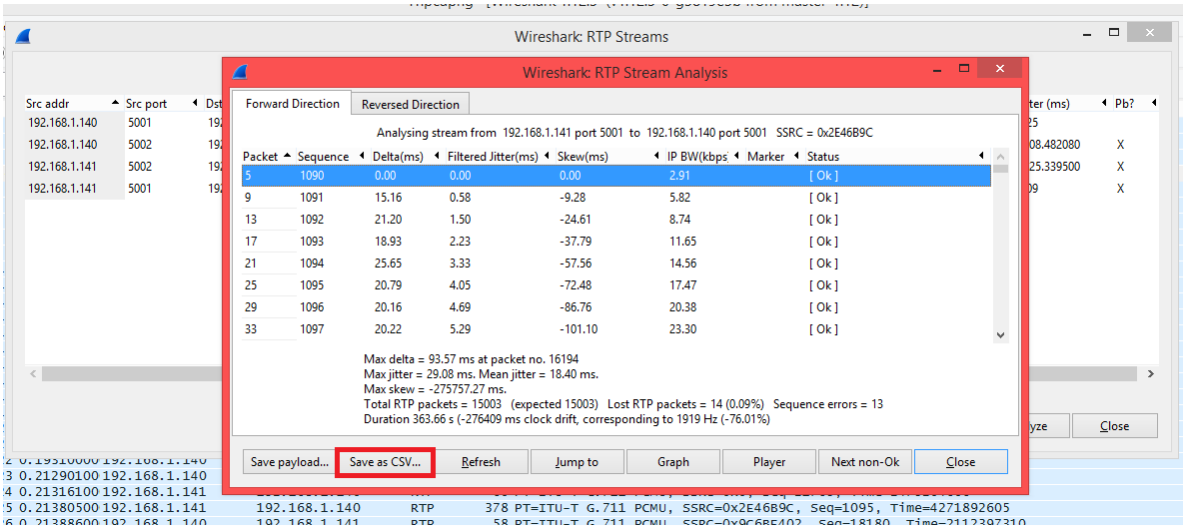


Figura 5.2120 Guardar en CSV

Posteriormente se elige la carpeta en donde se desea guardar el archivo; en este caso se guardó con el nombre de *PLR-0.1.csv* (es necesario colocar la extensión *.csv*) en el escritorio dentro de la carpeta *Análisis de mediciones*, luego click en *OK* para guardar el archivo, como se muestra en la Figura 5.22.

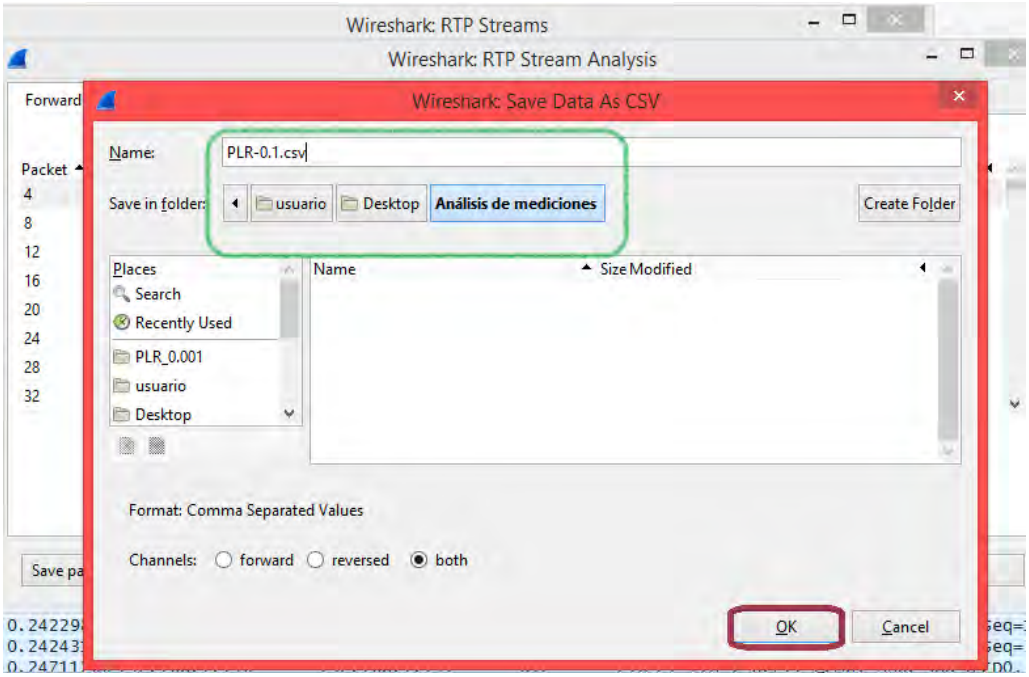


Figura 21.22 Guardar

Este procedimiento se realizó para las 440 mediciones para obtener el formato *CSV* de cada porcentaje de pérdida aplicado a cada tamaño de ventana. Al finalizar con todas las mediciones, se procede a utilizar un software desarrollado en C++ (*Ethereal CSV Processing)*, para obtener las series de tiempo de jitter en archivos *txt*; para realizar las estimaciones del parámetro H de cada serie de tiempo procesada.

En la Figura 5.23 se observa el software utilizado, como se puede notar hay seis parámetros a procesar que generaran seis series de tiempo (Sequency, Timestamp, Delta, Jitter Skew y IP BW), para seleccionar el archivo *CSV* le damos click en *Select a directory* para procesar de un conjunto de archivos CSV que se encuentren en un directorio determinado, y de esta manera se generen las series de tiempo correspondientes a cada medición.

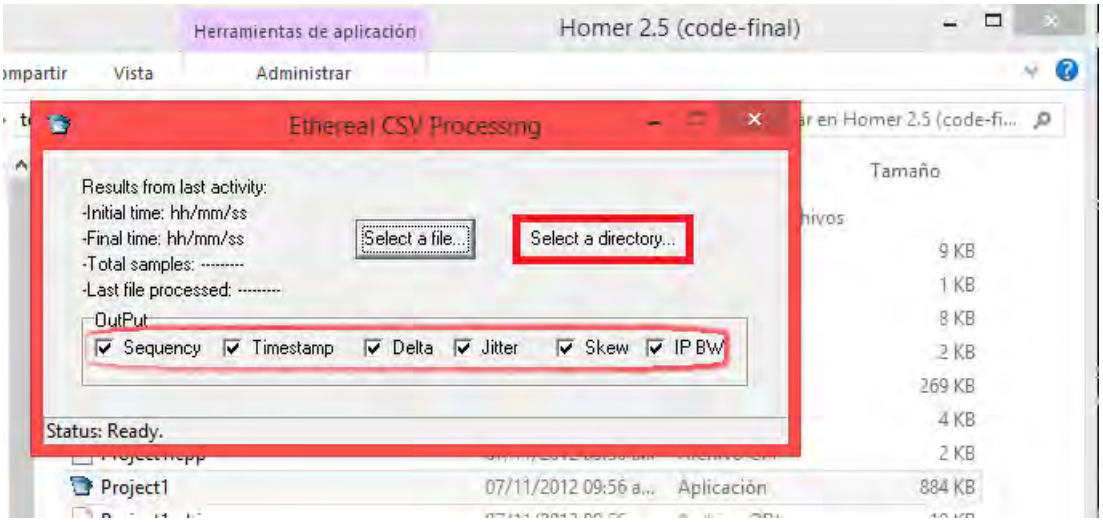


Figura 5.23 Seleccionar un directorio

Este proceso se realizó para las 20 ventanas obteniendo así los resultados requeridos, como se muestra en la Figura 5.24.

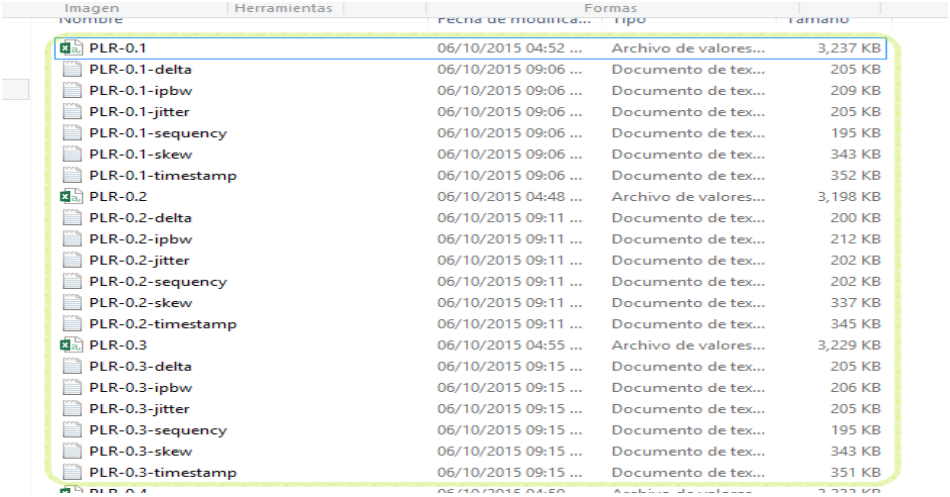


Figura 5.24 Resultados obtenidos por el software

Al finalizar el proceso anterior, se procede a obtener el valor del $Parámetro\ H$ a las trazas de jitter por tamaño de ventana para cada comunicación con diferente porcentaje de pérdidad emulado. Esto es posible creando una carpeta con el nombre de jitter dentro de la carpeta de cada ventana; posteriormente, se adjuntan las trazas de *jitter correspondientes*, como se muestra en la Figura 5.25.

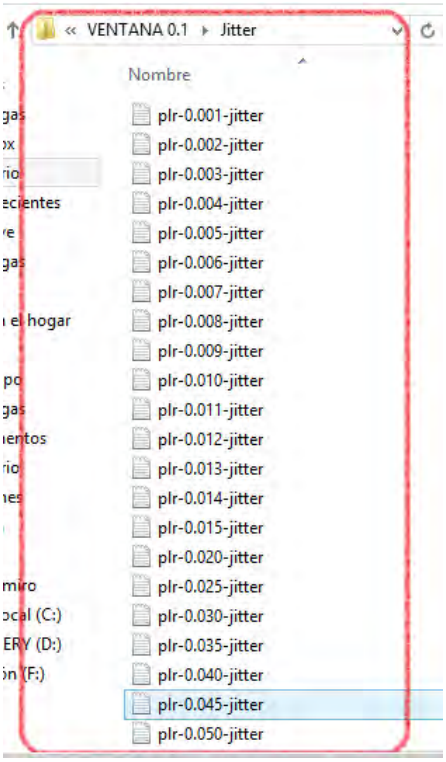


Figura 5.25 Archivos txt-jitter

Posteriormente se ejecuta un script en MATLAB que realiza la estimación del parámetro de Hurst mediante el método de la varianza de cada unas de las trazas de jitter procesadas.

En la Tabla 5.4 muestra los valores de H obtenidos para cada una de las 220 mediciones realizadas correspondientes al tamaño de paquete de $20ms$. Así mismo, en la Tabla 5.5 muestra los valores de H para las trazas de jitter correspondientes al tamaño de paquete de $40ms$, respectivamente. En total, se realizaron 440 mediciones.

En las Figuras 5.26 y 5.27, se muestran de manera gráfica los diversos valores de H estimados para cada comunicación con diferente valor de PLR emulado por tamaño de ventana. Como resultado se obtienen las diferentes familias de curvas en función del tamaño de ventana que relacionan el $Parámetro\ H$ vs PLR. En las figuras se puede observar que a medida que aumenta el PLR $(0.1\% \leq PLR \leq 5\%)$ aumenta ligeramente el $Parámetro\ H$. Sin

embargo, se puede observar que el parámetro de Hurst es capaz de detectar el nivel de rafagosidad, es decir, el $Parámetro\ H$ si considera los efectos de las pérdidas a ráfagas.

w/PLR	0.1%	0.2%	0.3%	0.4%	0.5%	0.6%	0.7%	0.8%	0.9%	1.0%	1.1%	1.2%	1.3%	1.4%	1.5%	2.0%	2.5%	3.0%	3.5%	4.0%	4.5%	5.0%
0.1	0.8604	0.8861	0.8960	0.9034	0.9131	0.9333	0.9334	0.9480	0.9489	0.9587	0.9605	0.9655	0.9674	0.9611	0.9700	0.9817	0.9860	0.9904	0.9920	0.9944	0.9963	0.9958
0.2	0.8583	0.8733	0.8700	0.8897	0.8902	0.8994	0.9095	0.9093	0.9173	0.9326	0.9355	0.9439	0.9429	0.9483	0.9482	0.9587	0.9683	0.9750	0.9786	0.9837	0.9848	0.9871
0.3	0.8472	0.8742	0.8655	0.8713	0.8895	0.8918	0.8982	0.9022	0.9022	0.9076	0.9098	0.9136	0.9193	0.9274	0.9263	0.9420	0.9521	0.9655	0.9655	0.9675	0.9766	0.9764
0.4	0.8601	0.8733	0.8767	0.8732	0.8809	0.8823	0.8883	0.8903	0.8953	0.8954	0.8979	0.9050	0.9099	0.9110	0.9133	0.9298	0.9381	0.9445	0.9549	0.9552	0.9610	0.9657
0.5	0.8545	0.8692	0.8602	0.8686	0.8676	0.8777	0.8709	0.8780	0.8839	0.8885	0.8925	0.8942	0.8955	0.9045	0.9009	0.9155	0.9233	0.9305	0.9356	0.9479	0.9505	0.9557
0.6	0.8531	0.8571	0.8731	0.8632	0.8651	0.8647	0.8748	0.8665	0.8781	0.8760	0.8823	0.8889	0.8793	0.8900	0.8897	0.9047	0.9070	0.9184	0.9209	0.9286	0.9509	0.9383
0.7	0.8546	0.8640	0.8719	0.8640	0.8668	0.8614	0.8691	0.8649	0.8660	0.8770	0.8709	0.8723	0.8747	0.8815	0.8821	0.8915	0.8986	0.9048	0.9111	0.9143	0.9142	0.9228
0.8	0.8623	0.8586	0.8680	0.8538	0.8567	0.8639	0.8622	0.8672	0.8718	0.8675	0.8651	0.8752	0.8753	0.8787	0.8757	0.8773	0.8841	0.8885	0.8953	0.8934	0.9038	0.9057
0.9	0.8674	0.8648	0.8634	0.8580	0.8594	0.8610	0.8615	0.8655	0.8573	0.8568	0.8555	0.8564	0.8629	0.8622	0.8669	0.8683	0.8702	0.8798	0.8776	0.8811	0.8824	0.8835
1.0	0.8580	0.8654	0.8635	0.8552	0.8510	0.8619	0.8638	0.8577	0.8535	0.8580	0.8608	0.8599	0.8558	0.8567	0.8459	0.8451	0.8499	0.8357	0.8550	0.8547	0.8584	0.8480

Tabla 5.4 Resultados de prueba H (tamaño de paquete 20ms)

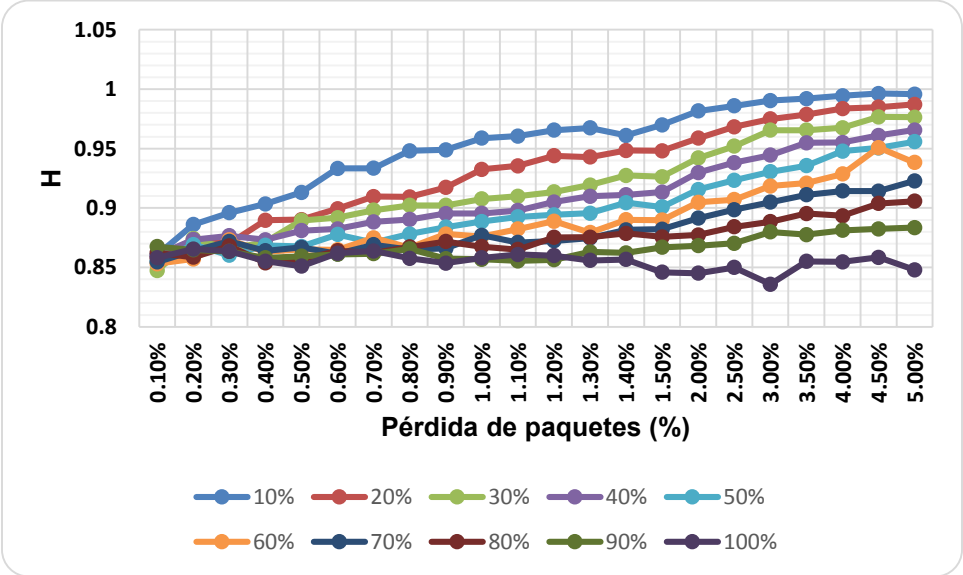


Figura 5.26 Resultados de prueba del Parámetro H (tamaño de paquete 20ms)

w/PLR	0.1%	0.2%	0.3%	0.4%	0.5%	0.6%	0.7%	0.8%	0.9%	1.0%	1.1%	1.2%	1.3%	1.4%	1.5%	2.0%	2.5%	3.0%	3.5%	4.0%	4.5%	5.0%
0.1	0.9255	0.9264	0.9300	0.9320	0.9361	0.9362	0.9555	0.9572	0.9649	0.9655	0.9662	0.9620	0.9678	0.9741	0.9738	0.9847	0.9865	0.9909	0.9924	0.9943	0.9963	0.9971
0.2	0.9222	0.9222	0.9234	0.9264	0.9277	0.9301	0.9351	0.9382	0.9417	0.9406	0.9425	0.9519	0.9556	0.9554	0.9594	0.9637	0.9720	0.9758	0.9834	0.9846	0.9874	0.9890
0.3	0.9243	0.9214	0.9208	0.9233	0.9259	0.9277	0.9251	0.9253	0.9345	0.9302	0.9312	0.9406	0.9379	0.9414	0.9436	0.9536	0.9571	0.9649	0.9738	0.9713	0.9771	0.9801
0.4	0.9226	0.9225	0.9235	0.9215	0.9223	0.9246	0.9252	0.9219	0.9248	0.9249	0.9315	0.9255	0.9317	0.9317	0.9353	0.9448	0.9494	0.9540	0.9593	0.9619	0.9673	0.9723
0.5	0.9225	0.9213	0.9227	0.9233	0.9227	0.9215	0.9244	0.9240	0.9182	0.9266	0.9260	0.9239	0.9221	0.9314	0.9280	0.9320	0.9387	0.9463	0.9505	0.9533	0.9561	0.9590
0.6	0.9228	0.9185	0.9182	0.9184	0.9126	0.9166	0.9197	0.9171	0.9194	0.9168	0.9187	0.9226	0.9186	0.9230	0.9208	0.9299	0.9311	0.9365	0.9361	0.9429	0.9514	0.9508
0.7	0.9225	0.9214	0.9195	0.9211	0.9172	0.9159	0.9179	0.9135	0.9178	0.9176	0.9198	0.9203	0.9204	0.9204	0.9165	0.9203	0.9259	0.9244	0.9288	0.9356	0.9383	0.9398
0.8	0.9216	0.9192	0.9171	0.9203	0.9154	0.9164	0.9164	0.9192	0.9210	0.9112	0.9110	0.9206	0.9147	0.9138	0.9199	0.9199	0.9120	0.9222	0.9224	0.9262	0.9208	0.9304
0.9	0.9234	0.9171	0.9204	0.9155	0.9183	0.9168	0.9170	0.9150	0.9073	0.9151	0.9113	0.9126	0.9112	0.9130	0.9098	0.9116	0.9108	0.9105	0.9110	0.9173	0.9174	0.9105
1.0	0.9233	0.9187	0.9205	0.9185	0.9192	0.9181	0.9072	0.9098	0.9096	0.9102	0.9088	0.9034	0.9100	0.9080	0.9047	0.9026	0.9051	0.8896	0.8929	0.8989	0.8979	0.8986

Tabla 5.5 Resultados de prueba H (tamaño de paquete 40ms)

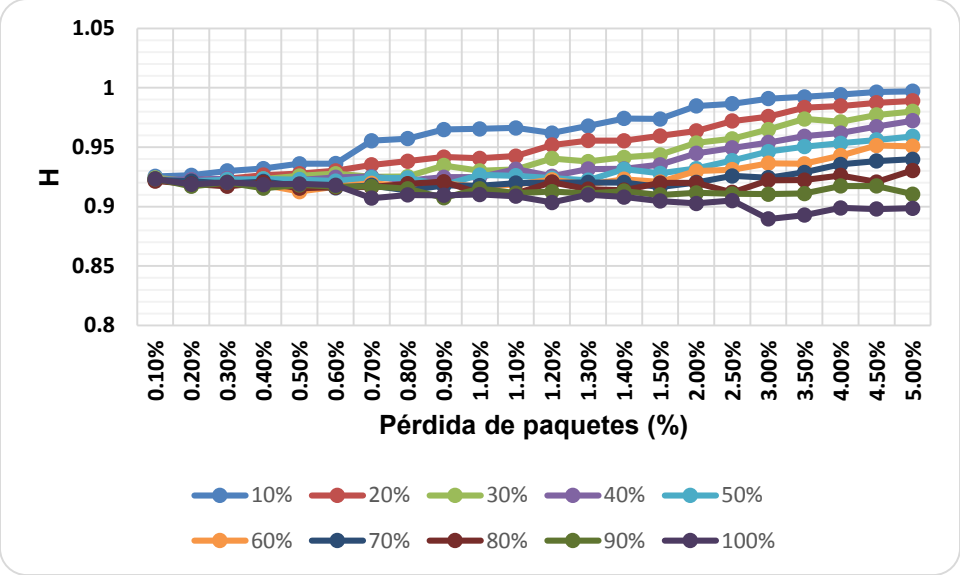


Figura 5.27 Resultados de prueba del Parámetro H (tamaño de paquete 40ms)

CAPÍTULO

6

6 Conclusiones

Las redes IP no garantizan calidad de servicio a aplicaciones multimedia tales como VoIP, esto debido a que proporcionan un servicio de mejor esfuerzo. Por tal motivo, retardos, jitter y pérdidad de paquetes son los deterioros más comúnes en la transmisión de voz sobre las redes IP. Derivado de estos puntos, muchos trabajos han sido orientados al análisis de las principales métricas de desempeño.

Por otra parte, existen diversos estudios que han demostrado que la pérdida de paquetes presenta naturaleza rafagoza y muestra una dependencia temporal. Dicho en otras palabras, si el paquete n se pierde, hay una probabilidad de que el paquete $n+1$ también se pierda. Con relación a lo anterior, existe una fuerte correlación entre la pérdida de paquetes consecutivos, lo que resulta en un comportamiento de pérdida de paquetes a ráfagas. Un modelo generalizado para capturar la dependencia temporal es una cadena de Markov de estados finitos.

En este trabajo se propuso una metodología práctica para emular pérdidas de paquetes mediante una aplicación VoIP en software (VoIPAS), la cual está basada en la generación de vectores de pérdida mediante cadenas de Markov de dos estados y ventanas de tiempo; estos vectores serán introducidos a la aplicación VoIPAS, donde cada elemento del vector corresponderá a un paquete de voz generado por la misma, cuando se presenta un 1 el paquete no se envía (paquete perdido) y cuando se presenta un 0 el paquete se envía (paquete recibido). Con esta metodología, se puede emular varios escenarios de red, estudiar el comportamiento de las principales métricas de desempeño y evaluar la calidad de servicio en una comunicación de voz sobre IP.

La finalidad de este trabajo fue emular pérdidas de paquetes mediante una aplicación VoIP; para ello se realizaron 440 vectores de PLR, con un tiempo total de 4400 minutos. Para evaluar la calidad de servicio (QoS) se utilizó el MOS. En

las Figuras 5.17 y 5.18 se pueden observar las diferentes familias de curvas de MOS vs PLR para diferentes tamaños de ventanas. Por otra parte, se observa que a medida que aumenta el PLR ($0.1\% \leq PLR \leq 5\%$), decrementa la calidad de servicio en la comunicación ($3.800 \leq MOS \leq 4.364$) para el tamaño de paquete 20 y para el tamaño de paquete 40 ($3.840 \leq MOS \leq 4.406$). También se pudo observar que el nivel de rafagosidad tiene efecto nulo en la QoS que percibe el usuario final. En conclusión, el MOS no considera los efectos de las pérdidas a ráfagas.

Además se realizó un estudio referente a las estructuras de correlación a las trazas de jitter. Esta estructura de correlación puede ser cuantificada mediante el parámetro de Hurst (H). Para estimar este parámetro, se utilizó método de la varianza.

Las Figuras 5.26 y 5.27, muestran las diferentes familias de curvas de Parámetro H vs PLR para diferentes tamaños de ventana. Se pudo observar que a medida que aumenta el PLR ($0.1\% \leq PLR \leq 5\%$) aumenta ligeramente el *Parámetro H*. Sin embargo, se puede observar tambien que el nivel de rafagosidad puede ser capturado mediante el parámetro de Hurst, es decir, el *Parámetro H* si considera los efectos de las pérdidas a ráfagas.

Por tal motivo, una posible mejora al Modelo E sería incorporar el parámetro H dentro de las ecuaciones del factor R, con el objetivo de proveer la habilidad de considerar diferentes niveles de rafgosidad dentro del Modelo E.