

UNIVERSIDAD NACIONAL DEL CALLAO

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA



PROYECTO DE TESIS

“MONITOREO Y CONTROL DE Q₀S DE LAS APLICACIONES MULTIMEDIA (VOZ DIGITALIZADA) EN TIEMPO REAL”

**SUSTENTACIÓN DE TESIS PARA OBTENER EL TÍTULO
PROFESIONAL DE INGENIERO ELECTRÓNICO**

PRESENTADO POR LOS BACHILLERES:

- BECERRA FLORES, RENZO PAVEL
- HUARANCCA AGESTO, ANGEL RONALD
- CAMPOS TARAZONA, OSCAR

Callao, 2016

PERU



**TESIS PARA OPTAR EL TITULO PROFESIONAL DE INGENIERO
ELECTRONICO**

**“MONITOREO Y CONTROL DE QoS DE LAS APLICACIONES
MULTIMEDIA (VOZ DIGITALIZADA) EN TIEMPO REAL”**

AUTOR(ES): BECERRA FLORES, RENZO PAVEL
HUARANCCA AGESTO, ANGEL RONALD
CAMPOS TARAZONA, OSCAR

ASESOR: ING. JORGE ELIAS MOSCOSO SANCHEZ

CALIFICACION: 16 (DIECISEIS)



DR. ING. SANTIAGO RUBIÑOS JIMENEZ
PRESIDENTE



ING. LUIS CRUZADO MONTAÑEZ
SECRETARIO



MG. ING. ABILIO BERNARDINO CUZCANO RIVAS
VOCAL

**CALLAO-PERU
2016**

INDICE

INTRODUCCION.....	4
I. PLANTEAMIENTO DEL PROBLEMA.....	8
1.1. Determinación del Problema.....	9
1.2. Formulación del Problema.....	11
1.3. Objetivos de la Investigación.....	12
1.4. Justificación de la Investigación.....	13
II. MARCO TEORICO	14
2.1. Antecedentes del Estudio.....	14
2.2. Marco Teórico.....	15
2.3. Definición de Términos Básicos.....	18
2.4. Abreviaturas Utilizadas	20
III. VARIABLES E HIPOTESIS.....	24
3.1. Definición de las Variables.....	25
3.2. Operacionalización de las Variables.....	25
3.3. Hipótesis General y Específica.....	26
IV. METODOLOGIA.....	28
4.1. Tipo de Investigación.....	28
4.2. Diseño de la Investigación.....	28
4.2.1. Introducción.....	29
4.2.2. Motivación.....	29
4.2.3. Objetivos y Contribuciones.....	31
4.2.4. Estructuración de la Tesis.....	34
4.2.5. Consideraciones respecto de QoS	36

4.2.6. QoS y las aplicaciones.....	38
4.2.7. Tipos de QoS y sus Parámetros/Métricas.....	40
4.2.8. Métricas de QoS y límites.....	44
4.2.9. Throughput	44
4.2.10. Retraso (delay).....	44
4.2.11 Eco y su Efecto en la Calidad de VOIP.....	45
4.2.12 Organizaciones Internacionales y sus Consideraciones sobre Calidad de Servicio.....	47
4.2.13 Los Parámetros Internacionales de QOS de la ITU-T Y TIPHON.....	54
4.2.14 Caracterización y Modelaje de los Parámetros /Métricas del Servicio de Voz.....	55
4.2.15 Los Máximos y Mínimos de los Parámetros de QOS.....	57
4.2.16 Cuantificación Lineal de la QOS.....	58
4.2.17 Cuantificación Lineal de los Codecs de este trabajo.....	59
4.2.18 Cálculo de las Métricas QOS.....	61
4.2.19 Ejemplo de Cálculo de QOS.....	63
4.2.20 QOS de solo tres Métricas.....	64
4.2.21 El Protocolo RTP.....	65
4.2.22 El protocolo RTCP.....	66
4.2.23 Las Estadísticas del Protocolo RTP/RTCP.....	67
4.2.24 Las RFCS del RTP/RTCP y sus Principales Especificaciones.....	69
4.2.25 La RFC 4585 (AVPF) y sus Principales Características.....	72
4.2.26 El Algoritmo de Generación del "EARLY PACKET"	73
4.2.27 Cálculo del "MAXIMUM DITHERING INTERVAL" (T-DITHER-MAX).....	73

4.2.28 Reglas para Envío y Recepción de Paquetes RTCP.....	74
4.2.29 Cálculo del Intervalo de Transmisión del RTCP.....	75
4.2.30 Inicialización de los Participantes de la Sesión.....	76
4.2.31 Recibiendo un Paquete RTP o RTCP que no es del Tipo “BYE”.....	78
4.2.32 Recibiendo un Paquete RTCP del tipo “BYE”.....	78
4.2.33 Time Out de un SSRC.....	79
4.2.34 Finalización del Timer de Transmisión.....	80
4.2.35 Transmitiendo un Paquete BYE.....	81
4.2.36 Actualización de WE_SENT.....	82
4.2.37 Asignación de Banda en el SDES.....	83
4.2.38 Informes SR Y RR.....	84
4.2.39 Ejemplo de Aplicación del Cálculo de Banda del RTCP.....	85
4.2.40 Formato de los Paquetes RTCP.....	86
4.2.41 Cálculo de las Métricas de QoS Cálculo del retraso fin-a-fin (RTT)....	89
4.2.42 Ejemplo de Cálculo del RTT.....	89
4.2.43 Cálculo del Jitter (IPDV).....	91
4.2.44 Cálculo de la Pérdida de Paquetes (IPLR).....	93
4.2.45 El Algoritmo de Monitoreo y Control de QoS.....	94
4.2.46 El Sistema de QoS.....	95
4.2.47 Los Módulos del Sistema de QoS.....	97
4.2.48 Los CODECS Utilizados en este Trabajo.....	101
4.2.49 Ejemplo de Cálculo de la Banda necesaria y Mínima para el CODEC G.711.....	103
4.2.50 El Algoritmo Propuesto.....	103

INTRODUCCION

El mayor problema de la utilización del protocolo UDP/IP como mecanismo de transporte para aplicaciones en tiempo real (real-time applications) es la falta de garantía de entrega de los paquetes a sus destinos, además que los paquetes dañados o retrasados con respecto al momento de su reproducción también son descartados. La alternativa de utilizar TCP/IP no es muy eficaz, debido a su gran "overhead" de encabezado ya que las múltiples solicitudes de retransmisión la convierten en no apta para aplicaciones en tiempo real debido a la generación de tráfico, a menudo inútil, si se considera que los paquetes retransmitidos pueden llegar demasiado tarde para su correcta reproducción. Con el objetivo de hacer frente a estas limitaciones, proporcionar una herramienta útil para aplicaciones multimedia y múltiples tentativas de solución, fue diseñado por el IETF (Internet Engineering Task Force), órgano de Ingeniería de la INTERNET, el protocolo RTP/RTCP.

Aunque previsto para uso exclusivo con UDP/IP, también se utiliza con otros protocolos, como TCP/IP y AAL5/IP* . El RTP, como es definido por el IETF, consta de dos partes, el RTP propiamente dicho, utilizado para la transmisión de datos, y su correspondiente protocolo de control RTCP, para envío de estadísticas periódicas de los receptores para sus fuentes (Tx) de datos. Entre otros recursos, el RTP proporciona: 1) Identificación del tipo de información; 2) Identificación de la fuente; 3) Identificación de la secuencia de los paquetes; 4) Timestamping.

El protocolo RTCP, proporciona informaciones periódicas sobre la calidad de los datos transmitidos/recibidos y estadísticas de los participantes de las sesiones. El

lado "receptor" de la aplicación que está siendo utilizada, envía periódicamente informes de recepción (RR) para su fuente de datos, estos informes incluyen datos de los parámetros que tienen que ver con las métricas de QoS, específicamente: pérdida de paquetes, retraso, variación de retraso y jitter.

De forma general, el RTCP permite implementar cuatro funciones básicas:

- Mejora de la calidad de la comunicación, utilizando las estadísticas generadas periódicamente.
- Mantener un registro de todos los participantes de la sesión a través del campo CNAME (Canonical Name).
- En el caso de conferencia todos los participantes envían paquetes RTCP a todos los demás participantes.
- Transporta alguna información de control de la sesión.

Cada paquete RTCP comienza con un cabezal fijo de 8 bytes seguido de elementos estructurados que pueden ser de longitud variable, de modo que varios paquetes RTCP pueden ser concatenados sin la necesidad de un separador, formando un paquete compuesto que luego se encapsula en un paquete UDP/IP.

I. PLANTEAMIENTO DEL PROBLEMA.

1. La INTERNET (Red mundial de computadores) inició sus actividades utilizando básicamente aplicaciones TCP altamente confiables y hoy conocidas como: Aplicaciones Standard que la convirtieron rápidamente en un grande suceso. Estas aplicaciones usan TCP como protocolo de transporte, no tienen problemas de timing o de ancho de banda ni tampoco exigencias de atraso, perdida o secuencia de paquetes. Si un paquete se pierde o atrasa, esto puede ser resuelto, por ejemplo, aplicando un mecanismo de retransmisión de paquetes.
2. La llegada de las tecnologías de banda ancha llevó a aumentos dramáticos, la capacidad de las redes conmutadas de paquetes desde algunos megabits por segundo a miles de megabits por segundo. Esta capacidad de comunicación de datos era necesaria para las nuevas aplicaciones tales como videoconferencia, voz y telefonía vía Internet conocidas genéricamente como Aplicaciones Multimedia.
3. Esas aplicaciones imponen requisitos muy estrictos a los parámetros que tienen que ver con las métricas de Calidad de Servicio (QoS) como atraso, variación, de atraso, perdida de paquetes, ancho de banda, etc.
4. Algunas requieren rígidos valores de atraso fin a fin y otras requieren un mínimo de ancho de banda, velocidad etc. Algunas requieren baja perdida de paquetes, otras pueden requerir alto "throughput". Visando la satisfacción de esos requerimientos los Investigadores mejoran la Internet actual principalmente en su sistema de QoS y un grande número de

protocolos tratan los problemas con él relacionados; sin embargo, muchos de estos problemas son complicados y difíciles de implementar, algunos son de grande costo de recursos y otros no atienden a los objetivos de garantía de QoS. Da la impresión que la Internet llegó a un impase.

5. Cada paquete AAL5 se divide en un número entero de células ATM y vuelto a montar en un paquete antes de la entrega a la recepción de acogida.
6. Este proceso se conoce como segmentación y re ensamblado (ver abajo). La última celda contiene relleno para asegurar que todo el paquete es un múltiplo de 48 octetos de largo. La célula final contiene hasta 40 octetos de datos, seguido de bytes de relleno y el remolque 8 octetos.

En otras palabras, AAL5 coloca el remolque en los últimos 8 octetos de la final de células donde se puede encontrar sin saber la longitud del paquete; el final de células se identifica mediante un bit en la cabecera ATM (ver más abajo), y el remolque está siempre en los últimos 8 octetos de esa célula.

1.1. DETERMINACION DEL PROBLEMA.

El inicio de la INTERNET se hizo con aplicaciones TCP, muy diferente de las actuales aplicaciones UDP que prácticamente acapararon el contexto de las comunicaciones con computadores y que se las conoce hoy, genéricamente como "Aplicaciones Multimedia" o también "aplicaciones que utilizan video y audio". Recurrente de esta situación, estas aplicaciones altamente sensibles a "timing" originaron la búsqueda de soluciones para los problemas resultantes de su uso, identificando como efectos negativos en las redes de conmutación de paquetes y circuitos, el atraso de las señales transportadas, la variación del atraso, el ancho

único punto) y utilizará métricas puramente “objetivas”. Estos parámetros son: 1)retraso fin a fin, 2) jitter, 3)perdida de paquetes y 4) ancho de banda que influyen en mayor o menor grado a las aplicaciones que trafican en las redes de paquetes, particularmente a las que tienen que ver con audio y video llamadas comúnmente de Aplicaciones Multimedia. Disponer de monitoreo y control de esos parámetros es lo que se conoce como QoS (Calidad de Servicio).

1.3. OBJETIVOS DE LA INVESTIGACION

Existen varios objetivos, entre ellos: el de ofrecer un mecanismo simple y fácil de implementar que permuta y evalúa la calidad de la voz en las redes de comunicación por paquetes. Al mismo tiempo, propiciar el entendimiento de las causas y efectos limitantes y restrictivos en las redes tipo IP.

1.3.1 OBJETIVO GENERAL

Proponer una metodología para monitoreo y control de calidad de las Aplicaciones Multimedia iniciando con el servicio de voz digitalizado, verificando su comportamiento dentro de los valores límites de QoS utilizados por las principales administraciones de Telecomunicaciones mundiales.

1.3.2 OBJETIVO ESPECÍFICO

1. Experimentar los efectos de los flujos TCP en el tráfico simultáneo UDP de voz y viceversa y su efecto en el ancho de banda y el comportamiento de las métricas de QoS en enlaces Wan de baja y media velocidad (acceso).

2. Contrastar los valores del ítem anterior con experiencias similares con ruteadores reales.
3. Adquirir conocimientos detallados de las redes IP, y de las Aplicaciones Multimedia verificando la ganancia tecnológica y de costo al usar las métricas desde un punto de fuente única y analizar el grado de aproximación de los valores obtenidos cuando se compara con el método actual subjetivo de la fórmula empírica de cálculo de la QoS indicada en el ítem y de las referencias máximas de las métricas.
4. Proponer un método alternativo de implementación QoS de forma objetiva para los circuitos de acceso.

1.4 JUSTIFICACION DE LA INVESTIGACION

El estudio se viabiliza y justifica por si solo en la medida que un mejor entendimiento de los conceptos de las redes IP pueden representar enormes beneficios y facilidades en la implementación de la calidad de servicio necesaria en las redes de conmutación de paquetes y en la Internet en particular; además de las economías resultantes de su implementación.

- Esta investigación se justifica también en la medida de su divulgación permitirá expandir en nuestro medio, el conocimiento y la problemática de la internet actual a sus desafíos en la gestión de redes.

II. MARCO TEORICO

2.1. ANTECEDENTES DEL ESTUDIO

Gran cantidad de investigaciones están en curso, procurando hacer las redes de conmutación de paquetes, más confiables principalmente para comunicación con aplicaciones Multimedia. Como las aplicaciones de voz (VoIP) son altamente sensibles al retraso, una buena ingeniería de red, fin-a-fin, es necesaria para tornar la **VoIP** un suceso. Hasta hace poco tiempo el principal requerimiento de las aplicaciones tradicionales tales como **FTP** y *Telnet* era la correcta entrega de los paquetes. Debido a que el protocolo **TCP** fue proyectado como confiable, retransmite los paquetes que no alcanzaron su destino o llegaron muy tarde, garantizando así la entrega de los paquetes.

Las técnicas convencionales para reducir la pérdida de paquetes como retransmisión y **FEC** (*Forward Error Correction*) podrían ser usadas.

Sin embargo, la retransmisión consume tiempo lo que las hace inadecuadas para aplicaciones *real time* y **FEC** está proyectada para mejorar los efectos de la pérdida de paquetes introduciendo datos redundantes. Otras técnicas como enrutamiento en diversidad y *adaptive rate*, tampoco son suficiente eficientes para este tipo de aplicaciones.

2.2. MARCO TEORICO

La transmisión de voz sobre redes IP (VoIP) tiene un rol cada vez más importante y un uso cada vez más difundido. Los usuarios esperan ver satisfechas sus expectativas de calidad del servicio con independencia de la tecnología utilizada. En este sentido, la "calidad de experiencia" QoE mide, "cuan bueno" un servicio de red satisface las expectativas y necesidades vistas por el usuario. Por otro lado, la "calidad de servicio" QoS se refiere a la medida del rendimiento de la red desde el punto de vista técnico, y a la posibilidad de gestionarla para cumplir con las prestaciones necesarias para las aplicaciones. La VoIP enfrenta problemáticas propias de las redes de datos, que se manifiestan como degradaciones en la calidad del servicio percibida por los usuarios (QoE). Estas degradaciones pueden deberse por ejemplo a retardos, *jitter* y pérdida de paquetes, entre otros factores. Para que la tecnología de VoIP pueda ser utilizada en forma masiva y comercial, es esencial garantizar una calidad de voz aceptable. Para ello se han desarrollado métodos para medirla. Estos métodos se dividen en dos grupos: **subjetivos y objetivos**. Los métodos **subjetivos**, se basan en conocer directamente la opinión de los usuarios. Típicamente resultan en un promedio de opiniones (por ejemplo, en un valor de MOS – Mean Opinión Score - con valores de 1 a 5). Los métodos **objetivos** miden propiedades físicas de una red para prever o estimar el rendimiento percibido por los usuarios. A su vez, estos últimos se subdividen en **Intrusivos** (se inyecta una señal de voz conocida en el canal y se estudia su degradación a la salida, (por ejemplo PESQ) y **No Intrusivos** (monitorean

ciertos parámetros en un punto de la red y en base a estos permite establecer en tiempo real la calidad que percibiría un usuario).

Entre otros motivos, por el costo y la dependencia de los aspectos subjetivos de las personas en medidas del tipo subjetivo, (método más usado) actualmente, se está buscando métodos confiables del tipo objetivo que atiendan plenamente para este fin y dejar de depender, entre otras cosas, de la subjetividad de cada persona.

Algunas Consideraciones sobre redes

A seguir algunas consideraciones sobre redes de paquetes y los protocolos de transporte TCP y UDP. Por ejemplo el control de flujo no es adecuado para media continua pues datos perdidos pueden ser tolerados pero retransmisiones o atrasos extras no son tolerados, y cuando un segmento no llega a su destino, el control de congestión asume "congestión" en lugar de errores. Protocolos nuevos tales como RTP [7] superan la mayoría de las limitaciones arriba mencionadas mediante el uso de APPLICATION LEVEL FRAMING (ALF), la integración de su procesamiento con el procesamiento propio de la aplicación (Integrated layer processing- ILP) [5] y el uso de perfiles para los algoritmos de control que son más adecuados para los requerimientos de las aplicaciones. Además RTP tiene un protocolo de control asociado, el RTCP, para evaluar las condiciones de la red, pero no impone ningún algoritmo predefinido para trabajar en el control de los datos; en este documento, no es tan importante destacar el caso concreto de nuestra elección

del RTP pero si considerar esas características abstractas de los nuevos protocolos. Una forma natural de usar redes ATM con un servicio "best-effort" es escoger la clase de servicio ABR (Available Bit Rate).

Sin embargo hay algunas razones porque esto no es adecuado para multimedia: El ABR fue proyectado para atender a aplicaciones de datos que tienen condiciones de controlar ancho de banda de reserva. Ellos pueden adaptarse a las condiciones variantes de la red pero es importante que la perdida de células sea tan baja cuanto posible (Aplicaciones de datos son bastante sensibles a perdida de células).

Esto no es un requisito para las aplicaciones multimedia. ABR tiene su propio ciclo de control (para prevenir perdidas) el cual es basado en conceptos abstractos de multimedia, tales como células (el mismo problema indicado arriba para los protocolos de transport tradicionales). El camino más sensato es entonces usar la clase de servicio Variable Bit Rate (VBR), y usar prioridades para trabajar con valores sobre contrato. La aplicación fija un límite inferior de ancho de banda para la conexión, bajo del cual no sirve para mantener la interacción. Después intenta elevar la calidad usando CLP=1 células, tanto cuanto la red pueda manejar.

Se asume que los mecanismos UPC/NPC usa "Cell tagging" hasta que el valor del Peak cell rate (PCR).es alcanzado y descarta células arriba de esta tasa. Por consiguiente una conexión de video, por ejemplo debería fijar el PCR de acuerdo a la longitud del frame, pero puede fijar el Cell Rate sostenible (SCR)

a un valor menor que el valor real que el pretende usar (Apenas arriba del mínimo nivel de QoS, por ejemplo).

2.3. DEFINICION DE TERMINOS BASICOS

AS: Autonomus System, áreas que en su conjunto modelan una red y dentro de las cuales las rutas son determinadas por el ruteo intradominio.

ASN1: Abstract Syntax Notation 1, lenguaje de definición de objetos estándar usados por SNMP.

BGP: Border Gateway Protocol. Protocolo de ruteo Intradominio.

CBR: Constraint Based Routing, ruteo que intenta encontrar un camino que optimice cierta métrica escalar y al mismo tiempo no viole un conjunto de restricciones.

CLNS: OSI Vonnection Less Network Srvice, protocolo bajo el cual opera SNMPv1.

CR-LDP: Constraint Route LDP, protocolo de distribución de etiquetas del tipo enrutamiento expícito que ofrece características de Ingeniería de Trafico.

CSPF: Constraint Shorset Path First, algoritmo para la computación de caminos que toma en cuenta al mismo tiempo un conjunto de restricciones.

DDP: Apple Talk Datagram-Delivery Protocol, protocolo bajo el cual opera SNMP V1.

DLCI: Data Link Connection Identifier, ejemplo de etiquetas o encabezado que pueden utilizarse como etiqueta de MPLS.

FEC: Forwarding Equivalence Class, representación de un conjunto de paquetes que comparten los mismos requerimientos para su transporte en MPLS.

IETF: Internet Engineering Task Force, grupo de trabajo dedicado en su mayoría al control del tráfico en lo que la Ingeniería de Tráfico se refiere.

IPX: Novel Internet Packet Exchange, protocolo bajo el cual opera SNMPv1. 23 **IS-**

IS: Intermediate System to Intermediate system, protocolo de ruteo intradominio.

ISP: Internet Service Provider, proveedor de acceso a Internet.

LDP: Label Distribution Protocol, protocolo responsable de que el LSP sea establecido para que sea funcional mediante el intercambio de etiquetas entre los nodos de la red.

LER: Label Edge Router, router encargado de la distribución de etiquetas.

LIB: Label Information Base, tabla de conectividad con la cual es examinada y comparada la etiqueta MPLS al llegar del LER al LSR, determinando la acción a ser seguida.

LSP: Label Switched Paths, ruta que sigue un paquete entre dos nodos de la red MPLS.

LSR: Label Switch router, router encargado de dirigir el tráfico dentro de la red MPLS.

MIB: Management Information Base, colección de información organizada jerárquicamente donde los objetos son accedidos usando SNMP y la cual reside en el elemento de red.

MIRA: MInimum Interface Routing Algorithm, algoritmo de ruteo de caminos que intenta minimizar la “interferencia” que provoca el establecimiento de un nuevo camino a potenciales nuevos caminos que son desconocidos.

MMF: Max-Min Fairness, principio de asignación usado para formular el esquema de utilización de recursos en donde se intenta asignar la mayor cantidad de recursos a cada demanda, al mismo tiempo que se intenta mantenerlos lo más similares posible.

MNF: Maximum Network Flow, máximo ancho de banda que puede traficar la red entre determinado par de nodos ya sea por un único camino o varios.

2.4 ABREVIATURAS UTILIZADAS

ACELP - Algebraic Code Excited Linear Prediction

ADSL - Asymmetric Digital Subscriber Line

APP - Application Defined Packet

ATM - Asynchronous Transfer Mode

ATT - American Telephone and Telegraph

AVPF - Audio-Visual Profile with Feedback

BW – Bandwidth

CNAME - Canonical Name

CPU - Central Processing Unit

CSRC - Computer Security Resource Center

DiffServ - Differentiated Service

DLSR - Delay since Last Sender Report

DSP - Digital Signal Processor

EMAIL - Electronic Mail

ERL - Echo Return Loss

ETSI - European Telecommunications Standards Institute

GMPLS - Generalized Multi-Protocol Label Switching

GPL - General Public License

GSM - Global System for Mobile

HTTP - Hypertext Transfer Protocol

IETF - Internet Engineering Task Force

ILBC - Internet Low Bit Rate Codec

IntServ - Integrated Service

IP - Internet Protocol

IPDT - Internet Packet Delay Transfer

IPDV - Internet Packet Delay Variation

IPLR - Internet Packet Loss Ratio

ISDN - Integrated Services Digital Network

ISP - Internet Service Provider

ITU-T - International Telecommunications Union - Telecommunications

LSR - Last Sender Report

Mbone - Multicast Backbone

MOS - Mean Opinion Score

MPLS - Multi Protocol Label Switching

MTU - Maximum Transmission Unit

PLC - Packet Loss Concealment

PPP - Point-to-Point Protocol

QoS - Quality of Service

RFC - Request for Comments

RR - Receiver Report

RTCP - Real-Time Transport Control Protocol

RTP - Real-Time Transport Protocol

RTT - Round Trip Time

SDES - Source Description

SLA - Service Level Agreement

SLS - Service Level Specification

SR - Sender Report

SSRC - Synchronization Source

TCP - Transmission Control Protocol

TIPHON - Telecommunications and Internet Protocol Harmonization Over Network

UDP - User Datagram Protocol

UTC - Universal Time Coordinated

VAD - Voice Activity Detection

III. VARIABLES E HIPOTESIS

Se asumió como hipótesis de este trabajo:

“La calidad del servicio de voz (telefonía), depende exclusivamente de los parámetros físicos que influyen sus características presentes en la red física en la cual trafican. La magnitud de la interferencia o degradación depende de la magnitud de estos factores los cuales son indicados a seguir y fueron retirados de los documento publicados por la UIT (Unión Internacional de Telecomunicaciones) organismo especializado de las Naciones Unidas en el campo de las telecomunicaciones que estudia los aspectos técnicos, de explotación y tarifarios y publica recomendaciones sobre los mismos, con miras a la normalización de las Telecomunicaciones en el plano mundial. La Asamblea Mundial de Normalización de las Telecomunicaciones (AMNT), que se celebra cada cuatro años, establece los temas que han de estudiar las Comisiones de Estudio del UIT-T, que a su vez producen recomendaciones sobre dichos temas. La aprobación de recomendaciones por los miembros de la UIT-T es el objeto del procedimiento establecido en la Resolución 1 de la AMNT. En ciertos sectores de la tecnología de la información que corresponden a la esfera de competencia del UIT-T, se preparan las normas necesarias en colaboración con la ISO y la CEI (Comisión electrónica Internacional). La norma vigente es la Rec. UIT-T G.1020 (07/2006).

3.1. DEFINICIÓN DE LAS VARIABLES EN LA INVESTIGACIÓN

Para los fines de este trabajo hemos considerado los parámetros a seguir como los influyentes en la calidad del servicio de voz: retardo, variación de retardo, perdida de paquetes y ancho de banda.

a.- Variables Independientes.

- **IPDT:** métrica del retardo fin a fin.
- **IPDV:** métrica de la variación del retardo (jitter).
- **IPLR:** métricas de la perdida de paquetes.

b.- Variables Dependientes.

- Calidad de servicio en condiciones normales de operación:
- $QoS = f(IPDT; IPDV; IPLR, BANDA)$
- QoS = Calidad de Servicio.

3.2 OPERACIONALIZACIÓN DE VARIABLES.

a.- Calidad de Servicio Normal

- $QoS = f(IPDT; IPDV; IPLR, Banda):$
- Donde: IPDT = 150ms
- IPDV = 50 ms
- IPLR = 3%

b.- Calidad de servicio durante congestión:

- $QoS_c = f(IPDT_c, IPDV_c, IPLR_c, BANDA):$
- Dónde: $IPDT_c > 150ms$, $IPDV_c > 50ms$, $IPLR_c > 3%$, $BANDA = f(CODEC)$

3.3.- HIPÓTESIS

3.3.1 Hipótesis General

Manteniendo los valores de los parámetros de red:

- IPDT = Atraso = de X ms.
- IPDV = Variación del atraso = Y ms.
- BANDA = (depende del Codec utilizado)

Se garante calidad de los servicios Multimedia. X,Y, Z y BANDA a ser determinados.

3.3.2 Hipótesis Específica

Manteniendo los parámetros de QoS por debajo de los valores indicados, la QoS de voz será garantizada.

IPDV < 150 ms, IPDV = 50 ms, IPLR = 1%, BANDA = dependiente del códec de voz.

3.3.3 Definición de Variables

A.- Variables independientes

- **IPDT = Atraso:** El intervalo de tiempo entre la salida del paquete de la fuente y la llegada al destino.
- **IPDV = "Jitter":** Variación del atraso: cada paquete demora un cierto tiempo variable en llegar a su destino.
- **IPLR = Pérdida de paquetes.** El porcentaje de unidades de datos que no llegaron a su destino en un intervalo dado.

B.- Variable Dependientes

- **QoS = Calidad de Servicio**

3.4. OPERACIONALIZACIÓN DE VARIABLES

- $QoS = f(IPDT; IPDV; IPLR, BANDA)$
- $QoS_{serviço-voz} = [a \cdot f(IPDT) + \beta \cdot g(IPDV) + \gamma \cdot t(THRU) + d \cdot w(IPLR)] \cdot f(IPDT) \cdot g(IPDV) \cdot t(THRU) \cdot w(IPLR)$

Donde: $0 = \{a, \beta, \gamma, d\} = 1, y$

$a + \beta + \gamma + d = 1$

- $QoS_{serviço-voz} = [a \cdot f(IPDT) + \beta \cdot g(IPDV) + d \cdot w(IPLR)] \cdot f(IPDT) \cdot g(IPDV) \cdot w(IPLR)$
- $QoS_{serviço-voz} = [a \cdot f(IPDT) + \beta \cdot g(IPDV)] \cdot f(IPDT) \cdot g(IPDV)$

IV. METODOLOGIA

4.1. Tipo de Investigación

Documental / Experimental vía simulación y "real". De una manera general, las actividades fueron: Identificación del problema, verificación de la documentación existente, selección de los instrumentos de recolección de datos, definición del procedimiento y método de obtención de los datos, resultados e informe final. Los datos resultantes fueron obtenidos por simulación con el software OPNET MODELER, versión académica. (Esta herramienta permite modelar, el tráfico través de un potente y amigable Interface gráfico), contrastando con resultados obtenidos en pruebas con equipos (ruteadores) reales.

4.2 Diseño de la investigación

La estrategia seguida fue la siguiente:

- 1) Verificación** del grado de influencia de flujos UDP-UDP Y TCP - UDP, traficando en el mismo enlace con un flujo de voz, G711, G 729, GSM y G723.1A, simulando situaciones de congestionamiento y calculando los valores de QoS correspondientes.
- 2) Repetición** de las experiencia indicada arriba, pero esta vez con equipos (routers) "reales".
- 3) Contrastación** de los valores obtenidos en las 2 experiencias, simulación y real.
- 4) Ejemplo** de implementación de QoS para tráfico sensible a pérdida de paquetes.

4.2.1 Introducción

En este Capítulo presentamos aspectos relativos a la motivación, objetivos y principales contribuciones de esta tesis.

4.2.2 Motivación:

El control de la Calidad de Servicio (**QoS**) es una de las tecnologías clave para las redes de la próxima generación (**NGN**), Actualmente, sólo la calidad de la red "troncal" está totalmente gestionada, pero la **QoS** fin-a-fin, para los usuarios finales necesita ser también garantizada.

La calidad de una red pública conmutada de telefonía (**PSTN**) es totalmente garantizada, y la calidad del terminal telefónico es bastante estable. De esta forma, el gerenciamiento de la calidad fin-a-fin de los servicios de telefonía fue conseguido. De otro lado, ninguna de las redes **IP** o servicios **VoIP** tiene su calidad garantizada. Existen numerosos tipos de terminales de **VoIP** con diferencias en sus implementaciones que afectan fuertemente la calidad de voz de los usuarios finales, indicando la necesidad urgente de tecnologías de gerenciamiento de calidad embarcadas en los terminales **VoIP** para conseguir el equivalente **QoS**, fin-a-fin, que el conseguido por la **PSTN**. Estas tecnologías son recomendadas en la **RFC 4585 "Extended RTP Profile for RTCP"** del **IETF** y en la recomendación **P.564** del **ITU-T**, como la estructura indicada para el gerenciamiento fin-a-fin de la calidad, pero la forma de su uso no es indicada. En este trabajo se está proponiendo un método de gerenciamiento fin-a-fin de la calidad de las señales de voz usando "**Extended RTP Profile for RTCP**".

Las normas **E.800** de la **ITU-T** (*International Telecommunications Union - Telecommunications*) [1] y **ETSI** (*European Telecommunications Standards*

recomendación G.107 [5] permite evaluar como la calidad de la conversación telefónica es afectada por los parámetros de los sistemas de transmisión y, como consecuencia, permite garantizar el grado de satisfacción de los usuarios limitando las características de transmisión fin-a-fin. Sin embargo, en el caso de los servicios de acceso a *Internet* existen aún grandes dificultades para establecer la relación entre la calidad percibida y la calidad de funcionamiento de la red, debido principalmente al carácter flexible y multiservicio de la *Internet*, que posibilita el uso de múltiples aplicaciones simples y complejas, pasando por navegación *web* o correo electrónico, hasta las actuales complejas aplicaciones multimedia interactivas.[6] [7].

4.2.3 OBJETIVO Y CONTRIBUCIONES

El objetivo general de este trabajo es proponer un método para seguimiento, evaluación y restauración de la calidad de servicio (QoS) para aplicaciones Multimedia sobre IP (VoIP), a partir de las características más relevantes de las redes y que influyen en la calidad de los servicios de voz y video partiendo de las siguientes premisas/referencias e incluyendo las actividades siguientes:

1. Como datos de entrada para evaluación: las estadísticas generadas por el protocolo RTCP (Real-Time Transport Control Protocol) de acuerdo con la recomendación RFC 4585 "Extended RTP Profile for RTCP" de Julio de 2006.
2. Como límites de las métricas de QoS, los valores "objetivos" máximos siguientes (compatibles con las recomendaciones ITU-T - Y - 1540/1541):
 - **IPDT** (*Internet Packet Delay Transfer*) (retraso) = 150 ms.
 - **IPDV** (*Internet Packet Delay Variation*) (variación del retraso) = 50 ms.

- **IPLR** (*Internet Packet Loss Ratio*) (tasa máxima de pérdida de paquetes) = 1%.
3. Como medio de restauración: un algoritmo a ser implementado en la capa de aplicación que cambia el *codec* por uno de menor banda cuando los límites son excedidos.
 4. *Codecs* utilizados: G.711, G.729, **GSM** (*Global System for Mobile*) y G.723.1 (**ACELP** [*Algebraic Code Excited Linear Prediction*]).
 5. Utilización de la Ecuación empírica 4.1, desarrollada para el cálculo de la Calidad de Servicio (**QoS**) indicada a continuación:

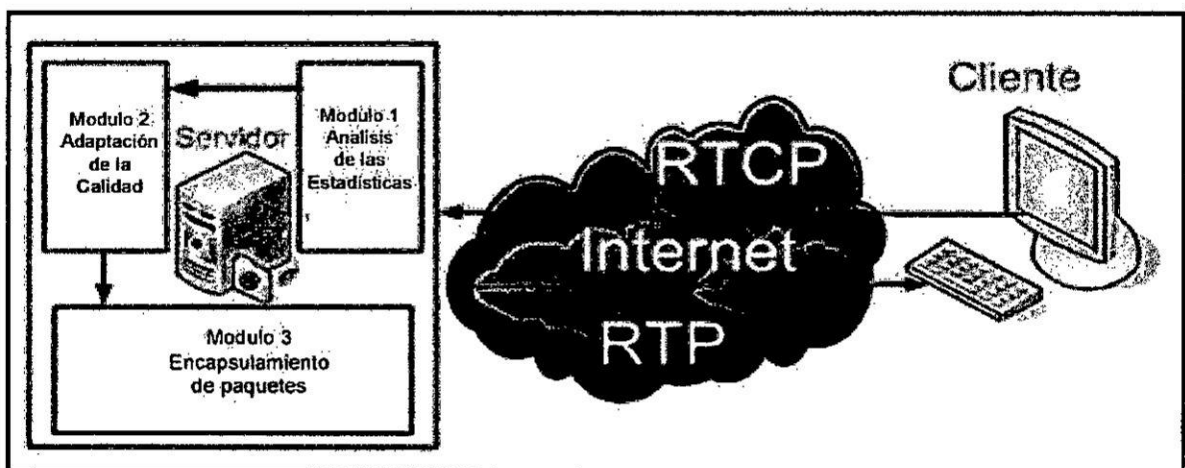
$$QoS = \frac{[\alpha x f (IPDT) + \beta x g (IPDV) + \delta x w (IPLR) + \gamma x t (THRU)]}{x f (IPDT) x g (IPDV) x w (IPLR) x t (THRU)}$$

en la cual: $0 < (\alpha, \beta, \delta, \gamma) < 1$ y $\alpha + \beta + \delta + \gamma = 1$

- 6.- Modelamiento de los parámetros de QoS
- 7.- Validación del algoritmo en un ambiente de simulación y real.
- 8.- El esquema general de la propuesta es presentado en la Fig. 4.1

FIG. 4.1

ESQUEMA GENERAL DE LA PROPUESTA



Las posibles contribuciones visualizadas de esta investigación son:

- La implementación de un algoritmo de seguimiento, evaluación y restauración de **QoS** para telefonía VoIP a partir de parámetros puramente objetivos obtenidos a partir de las estadísticas fornecidas por el protocolo **RTCP** de acuerdo con la especificación de la **RFC 4585** del **IETF**.
- Aplicación práctica de la Ecuación 4.1 para el cálculo de la **QoS**.

Las estadísticas del protocolo **RTP** (*Real-Time Transport Protocol*)/**RTCP** son generadas automáticamente de tiempo en tiempo incluyendo los parámetros relacionados con el cálculo de las métricas de **QoS**. Restricciones cuanto a los instantes de *timing*, no permitieron hasta hoy una utilización en gran escala, de estas estadísticas, pero la introducción, en 2006, de la **RFC 4585**, tiende a cambiar esta situación debido a retirada parcial de las restricciones iniciales y al hecho de la tendencia creciente por parte de los organismos internacionales para su utilización preferencial en la evaluación objetiva de **QoS** en los *packet-layer objective models*. Los límites máximos de las métricas están indicados en el ítem 2 de la pag. 31. Los valores considerados ya fueran ampliamente utilizados y probados en la red pública de telefonía. La escala de calidad aceptable tiene valores de **QoS** de 0 a 1 en incrementos de 0,1 siendo "1" el valor de la calidad estándar y "0" el valor mínimo permitido. La validación fué hecha vía simulación y con equipos reales (roteadores).

4.2.4 ESTRUCTURACIÓN DE LA TESIS

El trabajo de tesis propiamente dicho se inicia en el ítem 4.2.1 del Capítulo IV con el título de Introducción. El trabajo completo comprende los capítulos del I a X. A seguir una breve descripción del contenido de cada capítulo.

En la "Introducción", se hace una breve descripción de la motivación, objetivos y contribuciones visualizadas además de la estructuración completa del trabajo. En el capítulo I, se trata el Planteamiento inicial de la investigación, en el capítulo II se trata el Marco Teórico de la investigación considerando entre otros los antecedentes principales en la área de QoS y algunas abreviaciones y términos utilizados. En el capítulo III están incluidas las variables e Hipótesis hechas para este trabajo. En el capítulo IV entre otros conceptos (calidad de servicio) se presentan algunos conceptos fundamentales sobre QoS analizando entre otros los parámetros que tienen que ver con las métricas, sus valores, límites y los tipos utilizados por las principales organismos internacionales de estandarización, se abordan las estadísticas generadas por el protocolo **RTCP** y se explica la terminología utilizada en las diversas **RFCs** aplicables al asunto. Es detallado el algoritmo de monitoreo y control de **QoS** propuesto para reducir los efectos de congestión en las redes **IP** y describe cada uno de los módulos que lo componen.

Se detalla las experiencias efectuadas con flujos individuales de voz viajando solos y cuando comparten enlaces con flujos interferentes **TCP** y **UDP**. Trata también de la validación de la propuesta presentada.

valores de sus métricas, especialmente en ausencia de cualquier mecanismo para establecer prioridades o equivalente, por lo tanto, una de las ventajas del uso de mecanismos de **QoS** puede ser las prioridades y la protección a flujos de tráfico seleccionado.

Últimamente la discusión de estos puntos se está dando en un enfoque más equilibrado que mezcla “ancho de banda” y la instalación de características de **QoS** específicas para aplicaciones pre seleccionadas.

Las aplicaciones multimedia utilizan **UDP** (*User Datagram Protocol*) y son particularmente sensibles a retraso fin-a-fin y la variación de retraso, aunque muchas toleren una buena cantidad de pérdida de paquetes. Además, **TCP**, en el cual la mayoría de las aplicaciones convencionales se basan, fue diseñado para dar la máxima seguridad para la entrega fiable de datos, pero sin ningún tipo de control en caso de retraso, *jitter* o cualquier otro parámetro de **QoS**.

4.2.6 QoS Y LAS APLICACIONES

A continuación un resumen de las principales aplicaciones de acuerdo con sus características básicas y requisitos de **QoS**:

1. Aplicaciones Interactivas: Normalmente usadas para la relación entre personas o entre personas y máquinas. Implica en una secuencia de operaciones y transferencia de información entre los puntos finales de las aplicaciones. Estos tipos de aplicaciones pueden depender de uno o más parámetros de **QoS** al mismo tiempo, como **BW** (*Bandwidth*), retraso (*delay*), variación de retraso (*jitter*) y pérdida de paquetes (*Packet Loss*). En el caso de **VoIP**, las llamadas

telefónicas pueden requerir un retraso fin-a-fin máximo de 300 ms, aunque la degradación se nota después de 150 ms. *Jitter* es aceptable hasta 50 ms, por encima del cual es muy difícil adaptarse a sus efectos sobre la conversación.

2. Aplicaciones no interactivas: No interactúan con los criterios de valoración. Por ejemplo, se puede citar los datos de copia de seguridad y transferencia de archivos.

3. Aplicaciones elásticas: Pueden trabajar en una multitud de condiciones de la red sin una degradación significativa del rendimiento. Por lo general, el servicio *best effort* abastece a este tipo de aplicaciones que pueden ser "*hard*" y "*soft*". El tipo "*hard*" no puede funcionar si sus requisitos de **QoS** no se cumplen plenamente todo el tiempo, y los "*soft*", como "multimedia", toleran cierta degradación de los parámetros para un corto período de tiempo con calidad reducida.

4. Aplicaciones inelásticas: O *real-time* como se les conoce también, imponen controles estrictos *delay/jitter* y *bandwidth*, que son muy sensibles, pero pueden tolerar una cierta pérdida de paquetes a costa de implementación de las funciones adicionales, por ejemplo, el tamaño de los *buffers*.

5. Aplicaciones tolerantes: no deben confundirse con elásticas. Mientras que las elásticas no imponen ninguna exigencia, las tolerantes imponen restricciones de **QoS**, pero no muy rígidas, ya que permiten a las aplicaciones ejecutarse aún si los parámetros de **QoS** no son óptimos. Aplicaciones tolerantes son generalmente inelásticas con respecto a sus requisitos de **QoS**, pero si se cambian estos

parámetros, pueden no ejecutarse correctamente. Un ejemplo de este tipo de aplicación es **VoIP**.

6. Aplicaciones multimedia, datos y computación a gran escala: No todas las aplicaciones que dependen de **QoS** implican la transmisión de vídeo y audio. Hay muchas otras aplicaciones que requieren versiones fiables y precisas o muestras de datos periódicos, por ejemplo, las aplicaciones de red de ordenadores, las transacciones de comercio electrónico, control remoto de los sistemas distribuidos y herramientas de colaboración, que también requieren grandes volúmenes de datos (*terabytes*) que son transmitidos y procesados entre puntos *alejados geográficamente y dentro de un cierto intervalo de tiempo*.

4.2.7 Tipos de QoS y sus Parámetros/Métricas

Se identifica tres tipos de QoS: Intrínseca, percibida y evaluada.

Intrínseca: Es suministrada directamente por la propia red y se puede describir en términos de parámetros objetivos, tales como retraso, pérdida de paquetes, *jitter*, etc.

Percibida: Es la calidad "sentida" por los usuarios y depende en gran medida de rendimiento de la red, pero se mide por la "opinión media" de los usuarios, normalmente a través del método de MOS (*Mean Opinion Score*), cuyos valores significan: 1 = pobre, 2 = deficiente, 3 = regular, 4 = buena y 5 = excelente, siendo el MOS la media de todos los valores individuales.

Evaluada: Se define como la QoS tolerada por un usuario que está utilizando un servicio particular. Depende entre otras cosas, del mecanismo de precios, el nivel de apoyo de su ISP (*Internet Service Provider*) y otros aspectos comerciales y de *marketing*. En este tipo, una posible reducción en la calidad sin duda va ser tolerada por el usuario si el servicio no tiene costo, pero seguramente habrá quejas si el usuario está pagando por él.

En resumen, se puede suponer que la **QoS** intrínseca representa lo que podría hacer atractivo un servicio al cliente en particular. La calidad percibida es lo que hará que el servicio sea aceptable cuando el cliente lo use. El servicio evaluado es el que determina si el cliente mantendrá el servicio o lo devolverá a la mayor brevedad.

En la actualidad, la prestación de calidad de servicio que se ofrece como intrínseca se llama **SLS** (*Service Level Specification*), que es un conjunto de parámetros y sus valores que definen conjuntamente el servicio ofrecido a un determinado tipo de tráfico. **SLS** es el detalle técnico en el **SLA** (*Service Level Agreement*), que representa el acuerdo negociado entre el cliente y el proveedor de *Internet* y que incluye características de los niveles de servicio y sus indicadores asociados. La tabla 4.1 muestra un ejemplo de un **SLS** posible.

TABLA 4.1
EJEMPLO DE UN SLS

Service Level Specification	Range
Connection type	Constante Bit Rate (CBR)
Scope	End-to-end
Connection identificatio	Sequence of identifiers
Traffic description and conformance testing	Peak rate = 64 Kbps/bucket size for peak rate = 512 bytes/maximym burst size not applicable
Performance guarantees	Packet loss rate = 1% packet transfer delay = 250 ms/packet delay jitter = 30 ms
Multi level precedence and preemption (MLPP)	Priority

Puesto que hay muchos tipos de aplicaciones, por lo general cada una requiere un grado diferente de servicio. Varias organizaciones internacionales de normalización han estado tratando de definir las categorías de servicios (también llamadas "clases de QoS") implementadas en la capa de aplicación. Entre ellas podemos mencionar: la recomendación Y-1541 del ITU-T que presenta una definición de los tipos de QoS para el mundo IP, que se resumen en la tabla 4.2.

El ETSI en el proyecto TIPHON (Telecommunications and Internet Protocol Harmonization Over Network), propone una definición alternativa de QoS, que se resumen en la tabla 4.3.

TABLA 4.2
CLASES DE QOS DEL ITU-T

QoS	Characteristics
0	Real-time, jitter sensitive, highly interactive
1	Real-time, jitter sensitive, interactive
2	Transaction data, highly interactive
3	Transaction data, interactive
4	Low loss only (short transaction, bulk data, video)
5	Traditional applications of default IP networks

TABLA 4.3
CLASES DE QOS DEL TIPHON

QoS class	Components	QoS characteristics
Real-time conversational (telephone, videoconference, videophone and	Speech, audio, video, multimedia	Delay and delay variation sensitive limited tolerance to loss and errors, constant and
Real-time streaming (audio and video broadcast, surveillance, graphics)	Audio, video, multimedia	Tolerant to delay, delay variation sensitive, limited tolerance to loss and errors,

Near real-time interactive (web browsing)	Data	Delay sensitive, tolerant to delay variation, error sensitive,
Non-real-time tackground (e- mail and file transfer)	Data	Not delay and delay variation sensitive, error sensitive, best

Derivado de las categorías enumeradas en el servicio ISDN (Integrated Services Digital Network), el Foro ATM (Asynchronous Transfer Mode) define cinco categorías de servicios que figuran en la tabla 4.4.

TABLA 4.4
CATEGORÍAS DE SERVICIO DEL FORO ATM

ATM service categories	Representative applications	QoS characteristics
CBR (Constant Bit Rate)	Circuit emulation	Low cell delay variation, low loss
rt-VRB (real-time Variable Bite Rate)	Video on demand	Moderate cell delay variation, low loss
nrt-VRB (non-real-time Variable Bite Rate)	Packet traffic	Moderate loss
ABR (avariable Bit Rate)	Adaptable rate sources	Low loss
UBR (Unspecified Bit Rate)	Best-effort traffic	No requeriments

4.2.8 MÉTRICAS DE QOS Y LÍMITES

La especificación de los parámetros de QoS por lo general depende del contexto de las aplicaciones involucradas, pero los parámetros que se enumeran a continuación se consideran básicos y se discuten brevemente en lo que se refiere a su efecto sobre la comunicación.

4.2.9 THROUGHPUT

Es el número efectivo de unidades de datos transportados por unidad de tiempo (bits/segundo). En el caso de la voz, la banda necesaria depende del

codec usado, de la compresión del encabezado RTP, VAD (Voice Activity Detection), etc.

4.2.10. Retraso (delay)

Es el intervalo de tiempo entre la salida del paquete de la fuente y la llegada a su destino. Es comúnmente conocido como end-to-end delay. Sus efectos sobre el rendimiento de la transmisión generalmente están separados en dos áreas: 1) el eco 2) el problema de romper el flujo normal de la conversación. Los debates sobre VoIP generalmente tratan de esta cuestión sobre los efectos del retraso en cuanto a su posible interferencia con la cadencia normal de las conversaciones de voz. La recomendación G.114 del ITU, trata de esto y especifica que el retraso "one way" se puede acumular hasta 150 ms, sin efecto sobre la conversación. Para verificar esta recomendación, la ATT (American Telephone and Telegraph) realizó en 2004 en su laboratorio VQA (Voice Quality Assessment) nuevos estudios de retraso que se resumen a continuación. Se hicieron llamadas telefónicas a 24 miembros del grupo del laboratorio de apoyo de MOS, ATT, VQA y mantenidas conversaciones de un promedio de cinco minutos de duración. El retraso "one way" de estas conexiones se fijó en 200 ms. Las conversaciones fueron concluidas preguntando a los participantes si sentían alguna diferencia en esta conversación en relación a sus conversaciones normales. Nadie se ha quejó o dijo algo relacionado con el retraso o la superposición de las comunicaciones. La mayoría indicó que se trataba de una conversación normal, lo que llevó a la conclusión de que hasta 200 mseg. no se observó ninguna alteración.

4.2.11 ECO Y SU EFECTO EN LA CALIDAD DE VOIP

Las instalaciones iniciales de los enlaces **VoIP** en la red básica, mostraron una anomalía en el comienzo de una conexión, por lo general después de la primera palabra o dos palabras, llamada "eco inicial", cuyo origen se puede explicar brevemente de la siguiente manera:

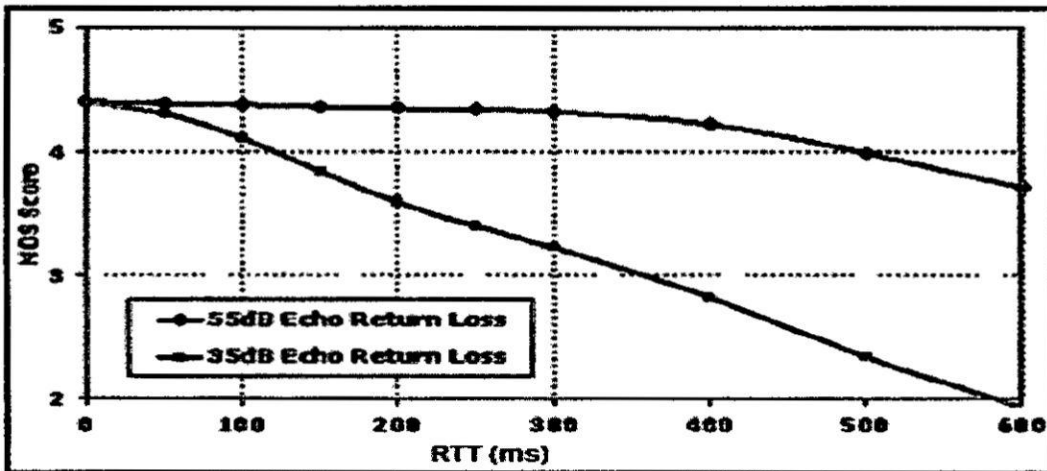
El procesador del filtro de adaptación del "supresor de eco" necesita un poco de tiempo para procesar los datos mientras está convergiendo, y por lo tanto una cierta cantidad de energía de eco residual "se escapa" del supresor de eco y se refleja de vuelta al usuario (que habla). La duración del eco inicial es una función compleja del **ERL** (Echo Return Loss) de la conexión, del proyecto del filtro adaptativo y del límite del supresor de eco (que bloquea la energía residual prácticamente "abriendo" el camino de regreso). La sensibilidad del usuario a este eco inicial aumenta directamente con el retraso de la conexión. El aumento de un enlace **VoIP** no hace que el supresor de eco de la red trabaje más lento, generalmente existe antes y después de la introducción del enlace **VoIP**, lo que pasa es que sin el retraso adicional de la conexión de **VoIP**, el eco inicial está por debajo del límite de la percepción del usuario y con este retraso adicional el eco comienza a ser "sentido" (oído). Este problema está relacionado a los usuarios cuyas líneas fueron indebidamente equilibradas en la red y, por lo tanto, caracterizadas por **ERL** bajos (en el rango de 8 a 12 dB).

Como ya se mencionó, el eco se expresa en términos de **ERL**, que es la relación entre la señal original y el nivel de eco expresado en decibeles (**dB**). Una relación alta significa un eco bajo, ejemplo, 55 **dB**, mientras que un eco de 15 **dB**,

representa un alto nivel de eco. El tipo más común de eco es conocido como "talker echo" (eco del usuario que habla) y representa la proporción de la voz de la persona que habla que se refleja de nuevo para sí mismo. La Fig. 4.3, muestra la relación entre delay y la calidad de la conversación para dos condiciones: 1) para ERL de 55 dB; 2) para un nivel moderado de eco (35 dB de ERL).

Si el RTT (*Round Trip Time*) es muy bajo, digamos 30 ms, el usuario "que habla" no puede distinguir entre el eco y el tono deliberadamente **introducido**.

FIG. 4.2 CALIDAD EN FUNCIÓN DEL ECO



Si el retraso es un poco más alto, digamos de 50 ms, el usuario que está hablando no puede escuchar la copia retrasada de su discurso como una copia distinta, pero impacta la calidad de la voz, dando como resultado una calidad de sonido generalmente descrita como "hollow" (agujero), "cave-like" (cavernoso), "tunnel-like" (túnel) o similar. A medida que aumenta el retraso, el eco se hace más evidente, y el efecto combinado de la intensidad del sonido (loudness) del eco y su retraso causa considerables molestias.

4.2.12 ORGANIZACIONES INTERNACIONALES Y SUS CONSIDERACIONES

SOBRE CALIDAD DE SERVICIO

La evaluación de la calidad de la comunicación implica la asociación de "necesidades objetivas de calidad de servicio" con los distintos tipos de tráfico. En relación con el entorno IP, los sistemas de medición objetivos más utilizados son los siguientes (usando la terminología de la ITU-T):

1. **IPDT:** *Internet Packet Delay Transfer*
2. **IPDV:** *Internet Packet Delay Variation (jitter)*
3. **IPLR:** *Internet Packet Loss Ratio*
4. **IPER:** *Internet Packet Error Rate*

Los parámetros antes mencionados direccionan colectivamente los aspectos más importantes de la calidad de los usuarios de las redes **IP**.

El **IPDT** se refiere al tiempo medio que requiere una red para transferir paquetes del origen al destino. Su limitación es de vital importancia para las implementaciones con éxito de la **VoIP**, videoconferencias y aplicaciones de datos en tiempo real. El **IPDV** caracteriza el *jitter* y debe ser controlado para evitar *underflow* u *overflow* en los *routers IP* o *buffers* de los terminales. El **IPLR** debe limitarse para garantizar la inteligibilidad de la voz.

$$\text{IPLR} = \frac{\text{paquetes perdidos}}{\text{paquetes enviados}}$$

Es importante señalar que la ITU-T 1540/1541 no hace referencia al parámetro de "rendimiento" o "ancho de banda". Lo mismo ocurre con la mayoría de los trabajos y publicaciones en este ámbito. Se utilizan sólo los parámetros de retraso, jitter y pérdida de paquetes como métricas para **QoS**. Hay que tener en cuenta que debe ser ofrecida la misma banda calculada por codec, cantidad de paquetes, etc. La tabla 4.5 muestra un resumen de las principales características de los codecs utilizados para **VoIP**.

TABLA 4.5

CARACTERÍSTICAS PRINCIPALES DE LOS CODECS PARA VOIP

	Codec	Codec Bite Rate (Kb)	Sample Period (msg) (Packets/seg)	Payload o Frame Sizes (bytes)	Frames por Paquete (Ethernet)	Overhead FR/ETH (bytes)	Banda Frame Relay (Kb)	Banda Ethernet (Kb)
1	G.711 (PCM)	64.000	20 (50)	160	1	47/58	82800	87.200
2	G.728 (LD-ELP)	16.000	10 (100)	20	1	47/58	26800	62.400
3	G.726 (ADPCM)	32.000	20 (50)	80	1	47/58	50.800	55.200
4	GSM	13.000	20 (50)	33	1	47/58	31800	36200
5	G.729 (CS-CELP)	8.000	10 (100)	10	1	47/58	45600	54.400
6	G.723-1 (MP-MLQ)	6.400	30 (33)	24	1	47/58	19.210	22202
7	G.723-1 A (ACELP)	5.300	30 (33)	20	1	47/58	18.190	21182
8	AMR (ACELP)	12.200	20 (50)	31	1	47/58	31.000	35400

9	AMR (ACELP)	7.400	20 (50)	19	1	47/58	26.200	30600
10	AMR WB/G.722-2 (ACELP)	6.600	20 (50)	17	1	47/58	26.000	29800
11	AMR (ACELP)	4.750	20 (50)	12	1	47/58	25400	27950

Nota: El simulador Opnet MODELER version académica usa overhead de 47 bytes em el protocolo PPP/Frame Relay y 58 bytes para el protocolo Ethernet.

Ejemplo de Cálculo de Ancho de banda (Para codec G723.1 A con protocolo relay).

Banda: $5300 \times 0.03 = 159/8 = 20$ bytes (Payload en bytes).

$20 + 47 = 67 \times 8 = 536 \times 34 = 18190$ bps (Payload + cabezal). Como ya se mencionó, es necesario usar los valores adecuados para cada aplicación específica. Por lo tanto, debe tener una serie de requisitos de **QoS** por tipo de tráfico. Los valores posibles máximos se muestran en la tabla de métricas de rendimiento, fin-a- fin en la tabla 4.6 que se tomaron de la recomendación ITU-T-Y-1541.

TABLA 4.6

MEDIDAS DE "PERFORMANCE" DE LA ITU-T

QoS CLASS	CHARACTERISTICS	IPDT	IPDV	IPLR	IPER
0	Real-time, jitter sensitive, highly interactive	100 ms	50 ms	1×10^{-3}	1×10^{-4}
1	Real-time, jitter sensitive, interactive	400 ms	50 ms	1×10^{-3}	1×10^{-4}
2	Transaction data, interactive	100 ms	U	1×10^{-3}	1×10^{-4}
3	Transaction data, interactive	400 ms	U	1×10^{-3}	1×10^{-4}

4	Low loss only (Short transactions, buçk data, video streaming)	1 s	U	1×10^{-3}	1×10^{-4}
5	Tradtional applications of default IP networks	U	U	U	U

* U: sin especificar.

Los valores indicados en la tabla se aplican a las redes públicas **IP** y se limitan a los valores promedios referenciales y cada empresa o **ISP** puede proporcionar sus propios valores de rendimiento fin-a-fin.

La clase 0, por ejemplo, significa que para una aplicación genérica, en tiempo real, sensible al *jitter* y altamente interactiva, los siguientes límites deberían garantizarse: retraso de transferencia fin-a-fin promedio por debajo de 0,1 segundos, *jitter* promedio por debajo de 0,05 segundos, *packet loss rate* por debajo de 10^{-3} y *packet error rate* debajo de 10^{-4}

La recomendación no se refiere a aplicaciones específicas, establece apenas los límites para tipos de tráfico. En la tabla 4.7 están indicados valores de referencia para algunas aplicaciones.

TABLA 4.7

EJEMPLOS DE REQUISITOS DE QOS DE ALGUNAS APLICACIONES

Application	IPDT	IPDV	IPLR
Data acquisition from sensors	100 ms	50 ms	0
Radar traces	20 msg	1-3 ms	10^{-3}
Weapon control	20 msg	10 ms	0
Sensor control	20 msg	10 ms	0
Voice	250 s	30 ms	10^{-2}
Video streaming	5-10 s	U	2×10^{-2}

En la tabla 4.8 están indicados los valores para voz que se utilizarán en este trabajo. Se definieron considerando los valores de las tablas anteriores y también los valores utilizados por las administraciones de telefonía convencional para conmutación de circuitos y serán las referencias (límites máximos) para todo este trabajo.

TABLA 4.8

MÉTRICAS DE ESTE TRABAJO

Aplicación	IPDT (msg)	IPDV (msg)	IPLR (seg.(%))
Voz sobre IP	≤ 150	< 50	$\leq 1\%$

Comentarios:

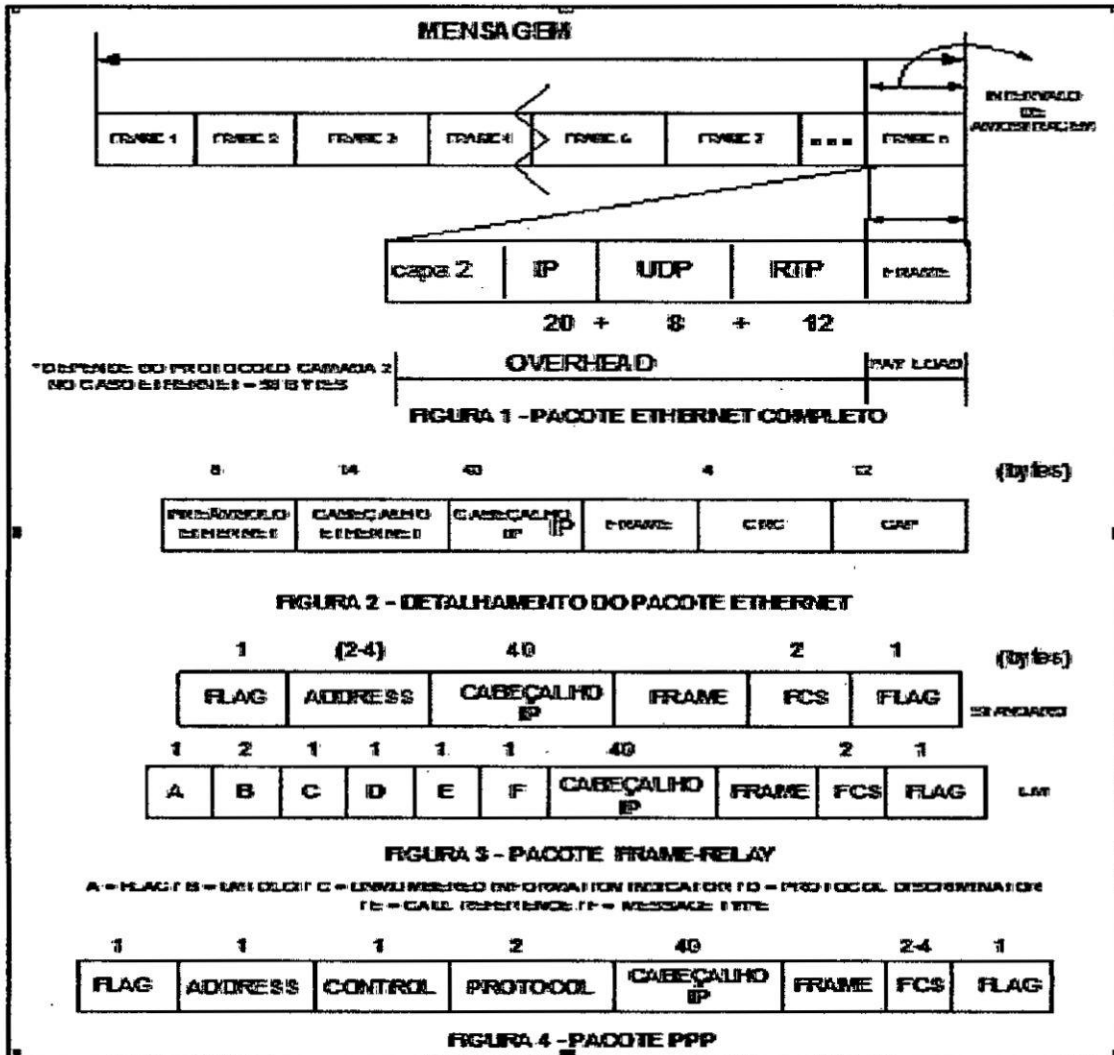
Hasta poco tiempo atrás la evaluación de calidad de la voz era hecha exclusivamente de forma subjetiva, a través del llamado "QoS percibido" (P QoS)

medido a través de la opinión media de los usuarios, usando el llamado MOS (mean opinión score) asignando un valor a la aplicación en evaluación, con los siguientes valores de MOS: 1=malo, 2=pobre, 3=médio, 4= bueno y 5=excelente. Aunque el objetivo principal sea la calidad percibida por el usuario, la calidad de servicio depende fuertemente de la performance de la red.

Esta situación há mudado um poco y actualmente, la mayoría de las implementaciones de QoS son ofrecidas en términos de QoS intrínseco (parâmetros objetivos) usando el llamado "Service Level Specification" (SLS), que son los parametros con sus valores que en conjunto definen el servicio ofrecido al trafico. SLS es la parte técnica separada del "acuerdo negociado entre el cliente y el proveedor del servicio, en lo relacionado a las características de los niveles de servicio y su conjunto de métricas asociadas" [ITU 62 t-Y-1241], el cual es comunmente conocido como Service Level Agreement (SLA). En resumen podemos decir que el SLS especifica el servicio desde el punto de vista técnico y el SLA incluye tambien aspectos no técnicos como precios y recursos de los dispositivos de la red.

FIG. 4.3

CABEZALES DE LOS PRINCIPALES PROTOCOLOS USADOS EN VOIP



4.2.13 LOS PARÁMETROS INTERNACIONALES DE QOS DE LA ITU-T Y TIPHON

Las tablas, 4.9 y 4.10 muestran los valores de los parámetros de QoS y sus límites utilizados por el TIPHON, (Telecommunications and Internet Protocol Harmonization over Network), Órgano Europeo y el ITU-T, respectivamente.

TABLA 4.9 TIPHON

Categoría de degradación	Pérdida de paquetes	Pico de jitter
Muy buenos	0%	0 ms
Bueno	3%	75 ms
Medio	15%	125 ms
Malo	25%	225 ms

TABLA 4.10**ITU-T- RECOMENDACIÓN G:114**

Atraso	Valor (mseg)
0 a 150 ms	aceptable para la mayoría de las aplicaciones del usuario
150 a 400 ms	aceptable desde que se conozca la influencia de ese atraso en la calidad de las aplicaciones del usuario
Más de 400 ms	inaceptable para efectos de planificación de la red

4.2.14 CARACTERIZACIÓN Y MODELAJE DE LOS PARÁMETROS /MÉTRICAS DEL SERVICIO DE VOZ

Conociendo el comportamiento de los parámetros de las métricas de QoS podemos caracterizarlos de acuerdo a lo indicado en la fig. 4.5; abajo, recordando que la QoS de los servicios de voz dependen de la THRU, IPDV, IDPV, e IPLR y aumentando con el aumento de THRU (fig A), y disminuyendo con el aumento del

IPDT, IPDV e IPLR (fig, B) y todas la afectan simultáneamente. Esta última afirmación puede ser representada por el grafico indicado en la fig. 4.6.

FIG.4.4

MODELAJE DE LA QOS

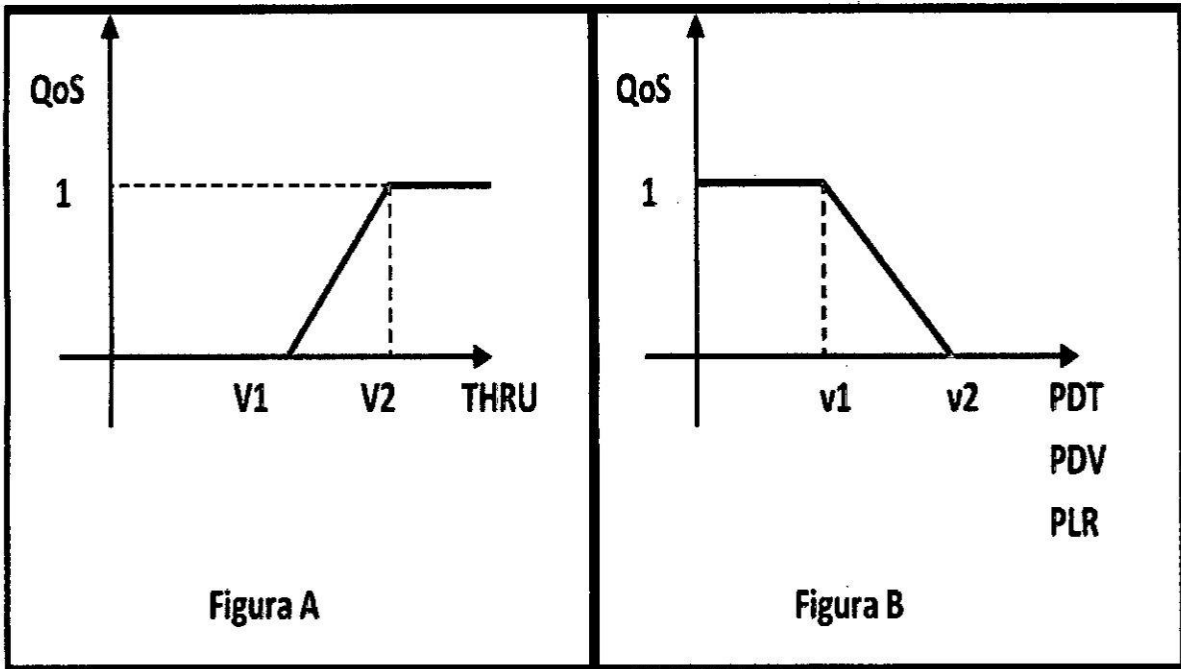
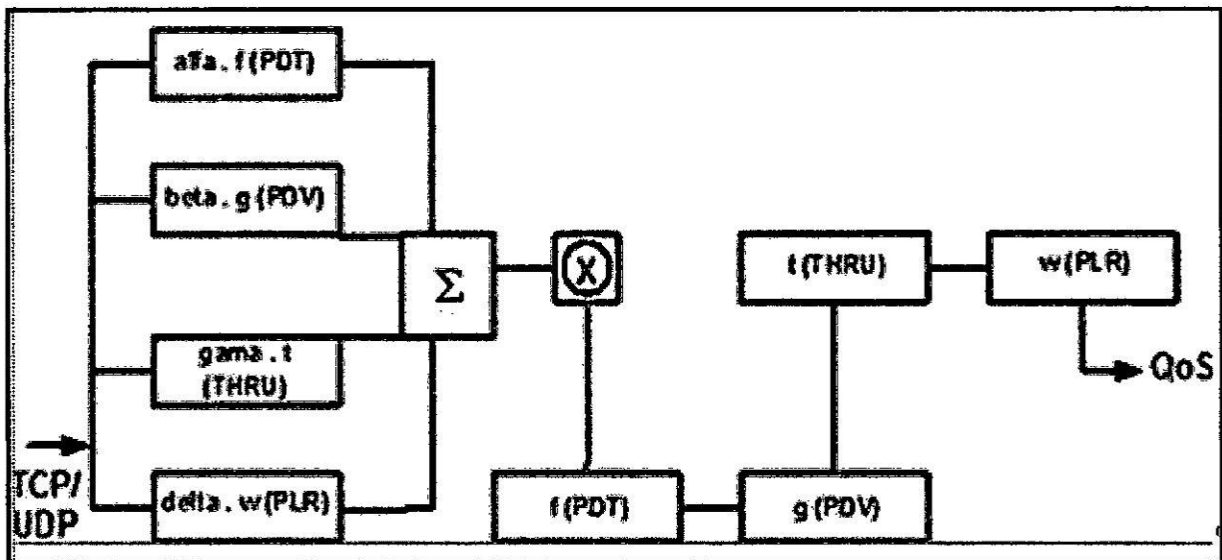


FIG.4.5

EFFECTO SIMULTÁNEO DE LAS MÉTRICAS DE QOS



De acuerdo con la fig. 4.6 [1], la expresión para la **QoS** de voz, puede ser escrita como: Por consiguiente la fórmula para el cálculo de este modelo puede ser escrita como indicada en [1] y la expresión para la QoS de voz puede ser considerada:

$$QoS \text{ servicio-voz} = [a \times f(\text{PDT}) + \beta \times g(\text{PDV}) + \gamma \times t(\text{THRU}) + d \times w(\text{PLR})] \times f(\text{PDT}) \times g(\text{PDV}) \times t(\text{THRU}) \times w(\text{PLR}) \dots$$

$$\text{Donde: } 0 \leq \{a, \beta, \gamma, d\} \leq 1, \text{ y} \tag{Ecuación (4.1)}$$

$$a + \beta + \gamma + d = 1 \tag{Ecuación (4.2)}$$

$$\text{Donde: } 0 \leq \{a, \beta, \gamma, d\} \leq 1, \text{ y} \tag{Ecuación (4.3)}$$

Las funciones $f(\cdot)$, $g(\cdot)$, $w(\cdot)$ y $t(\cdot)$, son del tipo representado en la Fig.4.5 y deben ser determinadas para cada uno de los servicios. *Alpha, beta, gamma y delta* son parámetros de ponderación.

4.2.15- LOS MÁXIMOS Y MÍNIMOS DE LOS PARÁMETROS DE QOS

FIG. 4.6

LOS LÍMITES DE LAS MÉTRICAS DE QOS

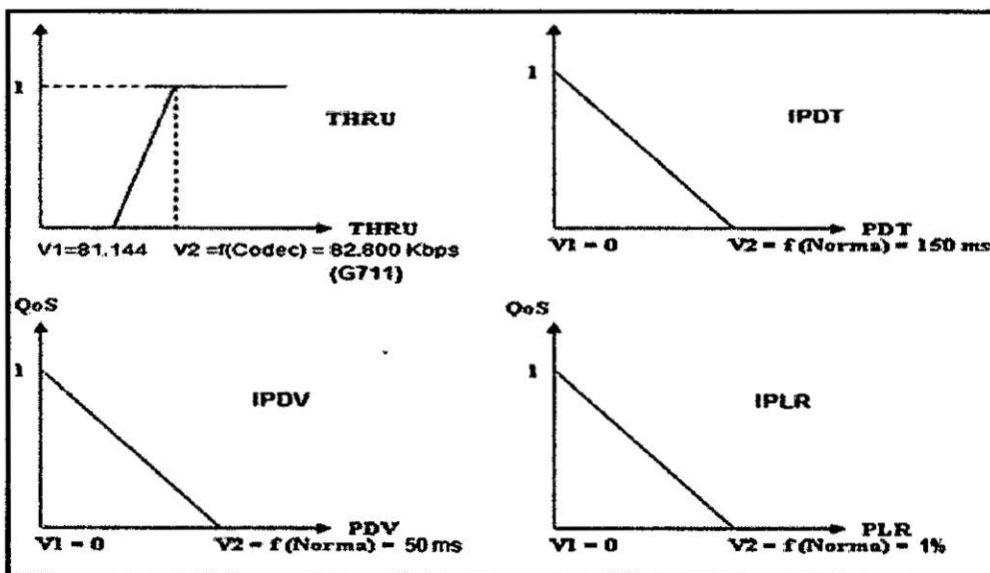


TABLA 4.11

LÍMITES DE LOS PARÁMETROS DE QOS

Codec	Banda V1 bps	Banda V2 bps	PDT V1 ms	PDT V2 ms	PDV V1 ms	PDV V2 ms	PLR V1 %	PLR V2 %
G.711	*81144	82800	0	150	0	50	0	1
G.729	*45144	45600	0	150	0	50	0	1

GSM	*31164	31800	0	150	0	50	0	1
G.723.1	*17654	18190	0	150	0	50	0	1

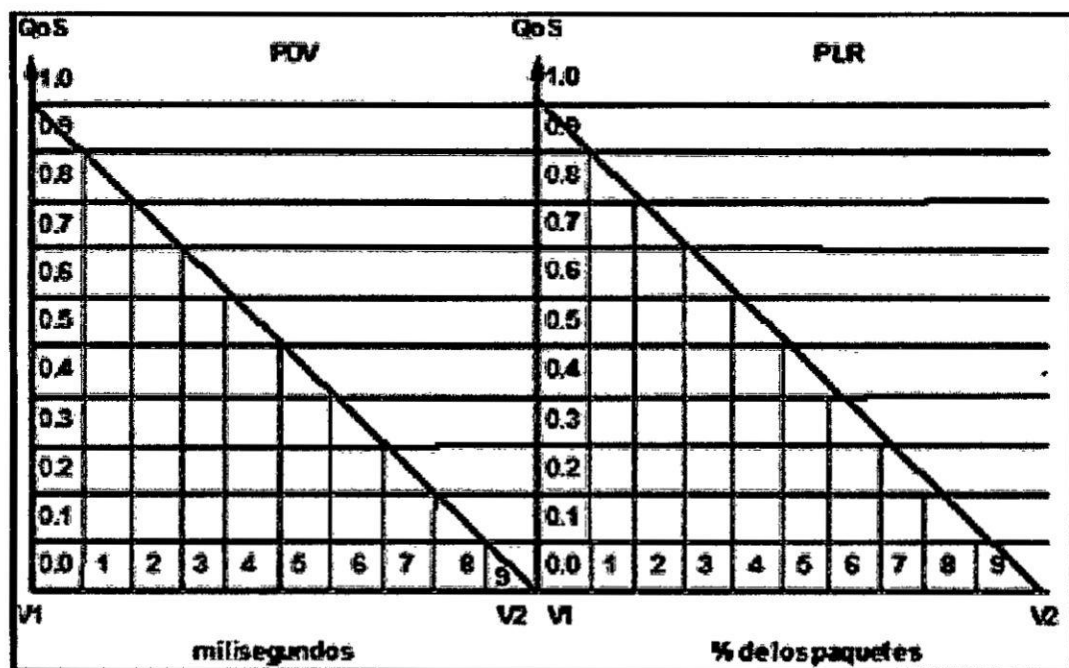
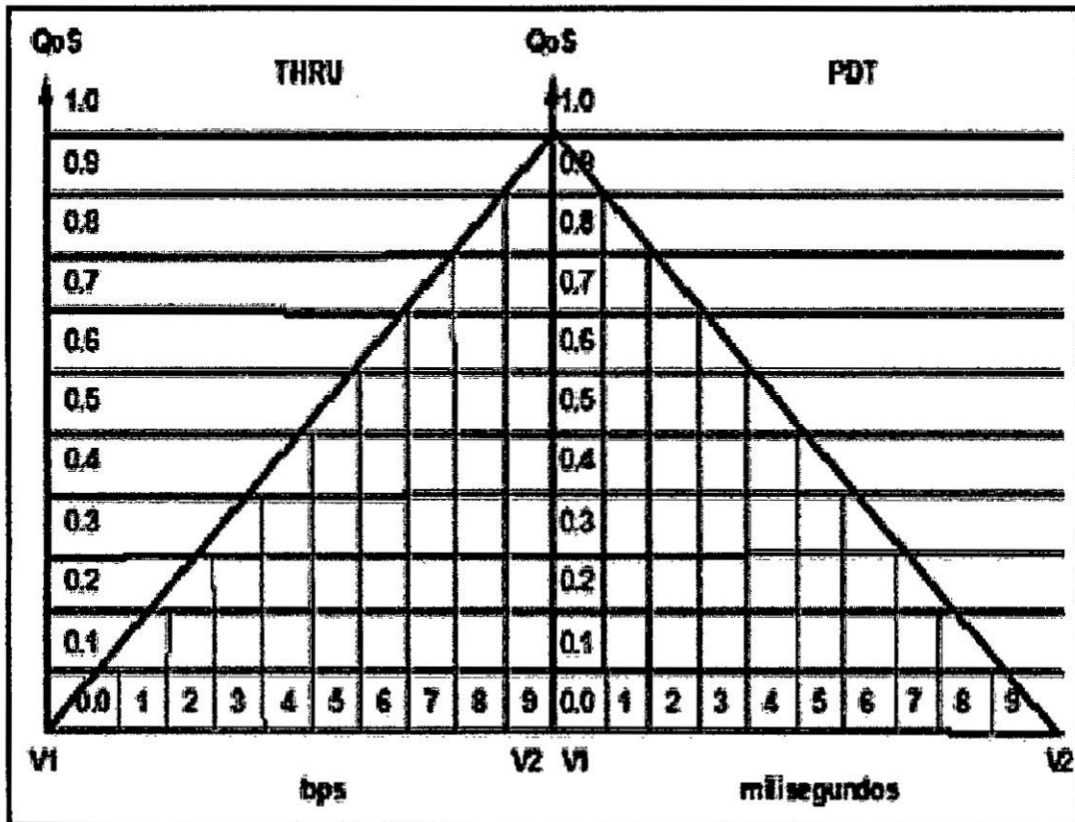
* Determinada en función de la IPLR

4.2.16 CUANTIFICACIÓN LINEAL DE LA QOS

La Fig. 4.8 abajo muestra el procedimiento de aproximación lineal de la participación de los parámetros **THRU**, **IPDT**, **IPDV** y **IPLR** en la **QoS**. Se asume la variación de **QoS** en incrementos iguales a 0.1.

FIG. 4.7

CUANTIFICACION LINEAL DE LA QoS



4.2.17 CUANTIFICACION LINEAL DE LOS CODECS DE ESTE TRABAJO

A seguir los resultados de la cuantificación aproximada para cada uno de los 4 Codecs anualizados.(para protocolo frame relay)

TABLA 4.12
CUANTIFICACIÓN LINEAL PARA EL CODEC G.711

QoS	THRU-bps	IPDT-mseg	IPDV-mseg	IPLR 3 x 10 ⁻²	IPLR Paquetes
1	82800	0.0	0.0	0.0	0.0
0.9	82634	15	.5	0.003	1
0.8	82468	30	10	0.006	1
0.7	82303	45	15	0.009	1
0.6	81137	60	20	0.012	1
0.5	81972	75	25	0.015	1
0.4	81806	90	30	0.018	1
0.3	81640	105	35	0.021	1
0.2	81475	120	4060	0.024	1
0.1	81309	135	45	0.027	1
0.0	81144	150	50	0.030	1

TABLA 4.13

CUANTIFICACION LINEAL PARA EL CODEC G.729 (100 PAQUETES POR SEGUNDO)

Qos	THRU-bps	IPDT-mseg	IPDV-mseg	IPLR 3x10 ⁻²	IPLR Paquetes
1	45600	0	0	0.00	0
0.9	45554	15	5	0.003	1
0.8	46508	30	10	0.006	1
0.7	45463	45	15	0.009	1
0.6	45417	60	20	0.012	1
0.5	45372	75	25	0.015	1
0.4	45326	90	30	0.018	1
0.3	45280	105	35	0.021	1
0.2	45235	120	40	0.024	1
0.1	45189	135	45	0.027	1
0.0	45144	150	50	0.03	1

TABLA 4.14
CUANTIFICACION LINEAL DE LA QOS PARA CODEC GSM
50 PAQUETES POR SEGUNDO

Qos	THRu-bps	IPDT mseg	IPDV mseg	IPLR Paquetes
1	31800	0	0	0
0.9	31736	15	5	1
0.8	31672	30	10	1
0.7	31609	45	15	1
0.6	31545	60	20	1
0.5	31482	75	25	1
0.4	31418	90	30	1
0.3	31354	105	35	1
0.2	31291	120	40	1
0.1	31227	135	45	1
0.0	31164	150	50	1

TABLA 4.15
CUANTIFICACION LINEAL DE LA QOS PARA CODEC G.723.1 ACELP
(34 PAQUETES POR SEGUNDO)

Qos	THRU-bps	IPDT-mseg	IPDV-mseg	IPLR Pacotes
1	18190	0	0	0
0.9	18136	15	5	1
0.8	18082	30	10	1
0.7	18029	45	15	1
0.6	17975	60	20	1
0.5	17922	75	25	1
0.4	17868	90	30	1
0.3	17814	105	35	1
0.2	17761	120	40	1
0.1	17707	135	45	1
0.0	17654	150	50	1

4.2.18 CALCULO DE LAS METRICAS QoS

Aplicando la formula (4.1) a los valores del Codec G.711, obtendremos:

- $QoS_{servicio-voz} = [a \cdot f(IPDT) + \beta \cdot g(IPDV) + \gamma \cdot t(THRU) + d \cdot w(IPLR)] \cdot f(IPDT) \cdot g(IPDV) \cdot t(ITHRU) \cdot w(IPLR)$

Ecuación (4.1)

Donde: $0 = \{a, \beta, \gamma, d\} = 1,$

Ecuación (4.2)

$a + \beta + \gamma + d = 1$

Ecuación (4.3)

Aproximadamente las métricas de QoS a la función "Rampa" tenemos: Si una función rampa continua en el tiempo es denotada por $R(t)$ y es expresada matemáticamente como:

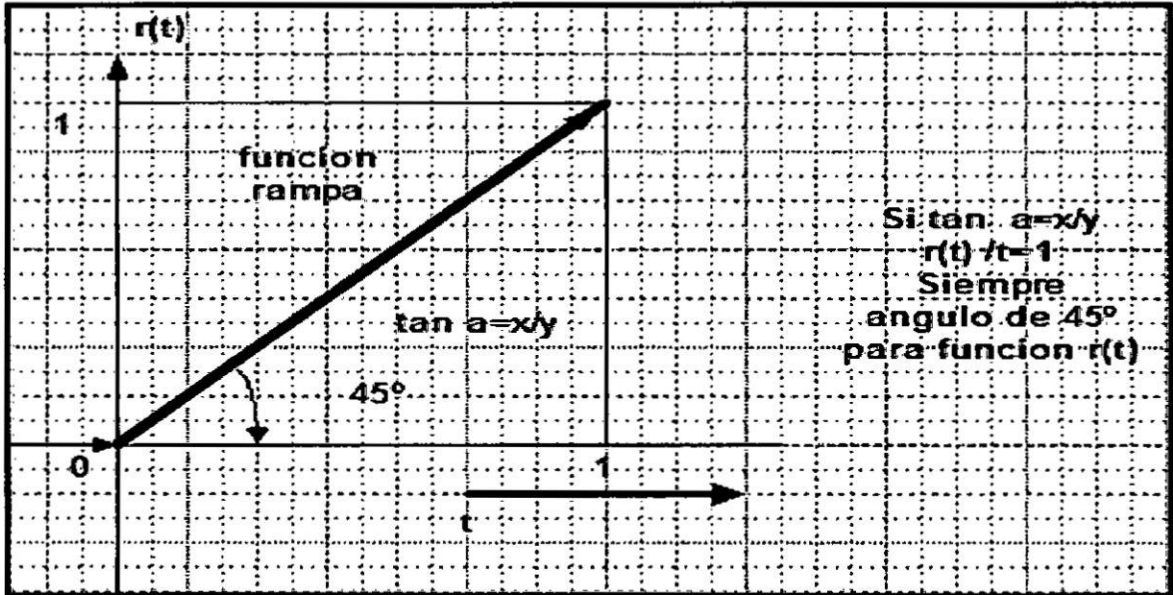
$$r(t) = t, \text{ si } t \geq 0 = 0,$$

caso contrario (esto es si $t < 0$)

La función rampa, $r(t)$ establece que la señal iniciara en el tiempo cero e instantáneamente tomara una inclinación dependiendo de las características del tiempo, para el lado positivo o negativo de un eje de coordenadas. Así la función rampa $r(t)$ es un tipo de función elemental la cual existe solamente para el lado negativo.

Fig.4.8

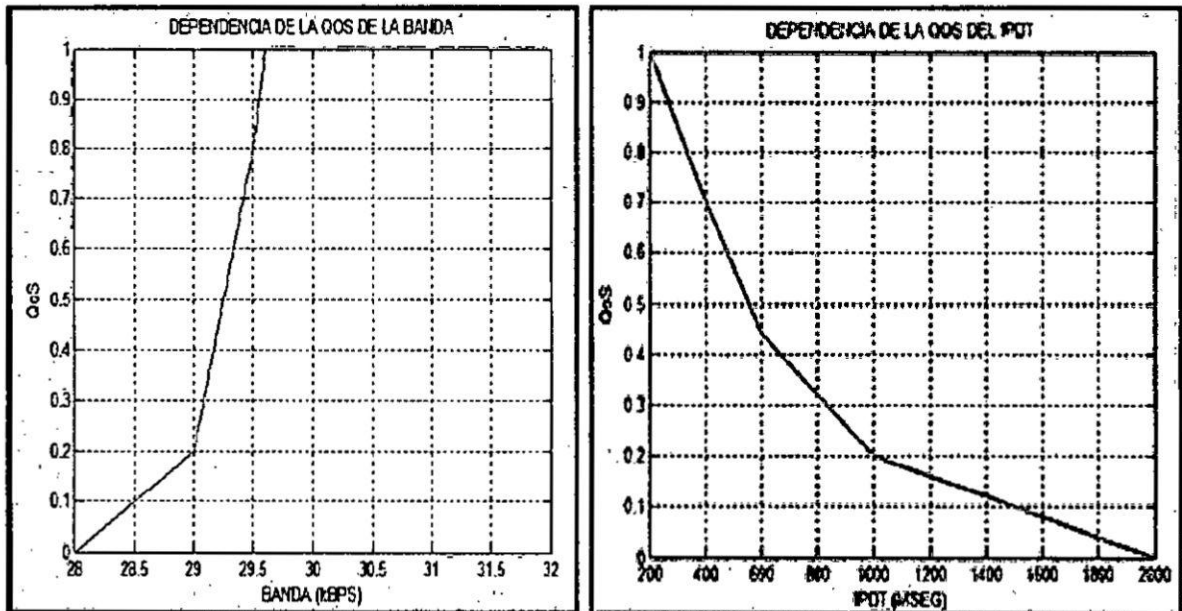
FUNCION RAMPA



Utilizando la función rampa de la fig.4.9 y también las fig.4.10 a,y b, tomadas de [1] podremos calcular las funciones $f(\text{IPDT})$; $g(\text{IPDV})$; $t(\text{THRU})$; y $w(\text{IPLR})$.

FIG 4.9

DEPENDENCIA DE LA QOS DEL IPDT Y THRU



4.2.19 EJEMPLO DE CÁLCULO DE QOS

- Para $\alpha = 0.25$, $\beta = 0.25$, $\gamma = 0.25$ y $\delta = 0.25$, G.711:
- Aplicando la ecuación 4.1 y función rampa a las funciones $f(\cdot)$:
- $f(\text{IPDT}) = (V2\text{IPDT} - \text{IPDT}) / (V2\text{IPDT} - V1\text{IPDT}) = 150 - 90 / 150 = 0.4$ mseg.
- $g(\text{IPDV}) = (V2\text{IPDV} - \text{IPDV}) / (V2\text{IPDV} - V1\text{IPDV}) = 50 - 30 / 50 = 0.4$
- $t(\text{THRU}) = (V2\text{THRU} - \text{THRU}) / (V2\text{THRU} - V1\text{THRU}) = 82800 - 81904 / 82800 - 81520 = 0.7$
- $W(\text{IPLR}) = (V2 \text{ IPLR} - \text{IPLR}) / (V2 \text{ IPLR} - V1 \text{ IPLR}) = (\text{NOTA})$

NOTA: De acuerdo con la tabla 3.6, la ITU-T recomienda 1×10^{-3} paquetes/seg. O 1×10^{-4} bits por seg.

Somos del parecer de usar esta métrica para utilizar esos límites como alerta de una anomalía en curso tipo inicio de congestión. Por tanto $QoS = (0.4 + 0.4 + 0.7) \times 0.25 = 0.35$

4.2.20 QOS DE SOLO TRES MÉTRICAS

Es nuestra opinión, usar la métrica IPLR como alerta de una situación de congestión en curso, de forma que la mudanza de ancho de banda pueda resolver cuando conmutar para un códec de menor banda. De esta forma se recomienda dejar la fórmula con 3 métricas hasta efectuar pruebas de sistema que confirmen la validez de esta decisión. Después de algunas pruebas analizando el comportamiento sería decidido si continuar como métrica o como una alarma del sistema.

FIG 4.10

QoS FUNCIÓN DEL IPDT E IPDV

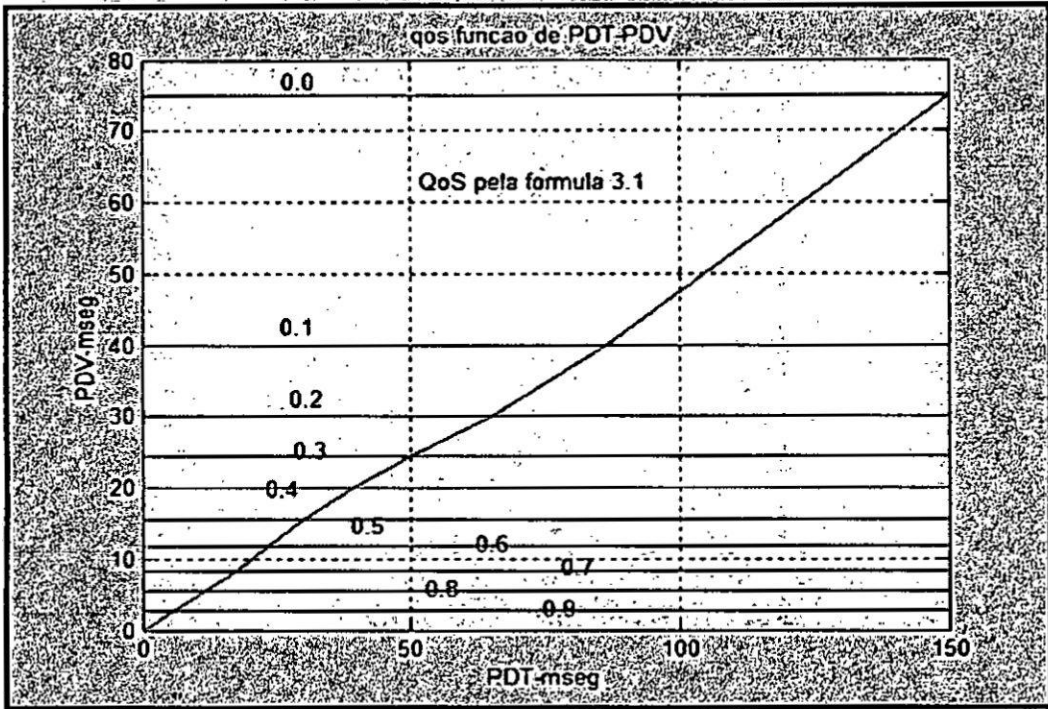
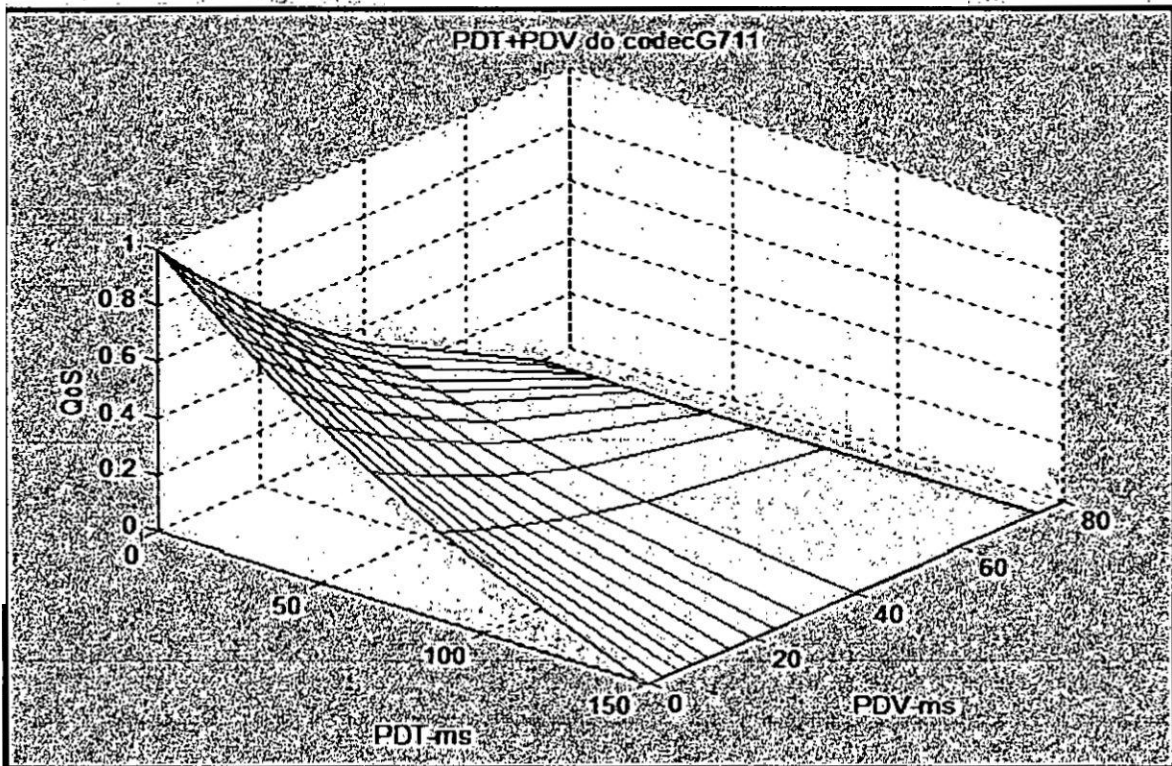


FIG. 4.11

QoS POR LA FÓRMULA 4.2 (PDT + PDV)



4.2.21 EL PROTOCOLO RTP

El **TCP** es actualmente, el protocolo de transporte estándar utilizado en Internet (aproximadamente 60%). Sin embargo, varios hechos lo hacen inadecuado para su uso con el tráfico real-time. Entre ellos:

1. **TCP** es un protocolo punto a punto sin un apoyo directo para la transmisión multicast.
2. Utiliza un mecanismo de retransmisión interno inútil para aplicaciones en real-time.
3. No transporta ningún tipo de información de "timing" (requerido por la mayoría de las aplicaciones real-time).

El otro protocolo de transporte **UDP**, tampoco incluye ninguna información de "timing". Por estos motivos, entre otros, un nuevo protocolo de transporte, llamado **RTP**, fue especificado por el **IETF** y rápidamente adoptado como protocolo estándar para el transporte de sesiones multimedia por la **ITU-T**. El **RTP** es considerado como un protocolo de transporte real-time fin-a-fin, recordando solamente que llamarlo así (como un protocolo de transporte) puede llevar a confusión, ya que se utiliza normalmente con **UDP** o **TCP**, que son, de hecho, los protocolos de transporte. Como actúa muy próximo de la aplicación transportada, es visto más como un marco que las aplicaciones pueden utilizar para implementar un nuevo protocolo (individual). El **RTP** no garantiza la entrega sincronizada ni ordenada de paquetes y deja la responsabilidad de recuperación de los segmentos perdidos y el ordenamiento de los paquetes por cuenta de la capa de aplicación. Funciona junto

con el **RTCP**, su protocolo de control y retroalimentación de la calidad de la comunicación.

4.2.22 El protocolo RTCP

Utilizando el **RTCP**, el lado "receptor" de la aplicación envía "informes de recepción" (**RR**) para su fuente de datos. Estos informes incluyen informaciones de los parámetros que permiten calcular las métricas de **QoS** específicamente la pérdida de paquetes, variación de retraso (*jitter*) y retraso (*delay*). En general, el **RTCP** realiza cuatro funciones básicas:

1. Proporciona mejora de la calidad de la comunicación;
2. Mantiene un registro de todos los participantes de la reunión a través del campo

CNAME (*Canonical Name*).

3. En el caso de la conferencia, todos los participantes envían paquetes **RTCP** a todos los demás participantes.
4. Transporta alguna información de control de la sesión.

Cada paquete **RTCP** comienza con un encabezado fijo de 8 *bytes* seguido de elementos estructurados que pueden ser de longitud variable, de modo que varios paquetes **RTCP** pueden ser concatenados sin la necesidad de un separador, formando un paquete compuesto que luego se encapsula en un paquete **UDP/IP**.

4.2.23 LAS ESTADÍSTICAS DEL PROTOCOLO RTP/RTCP

El RTP consiste de 2 protocolos: RTP para aplicaciones transmitiendo datos real time y RTCP para monitoramiento de QoS y transmitir informaciones de los participantes durante una sesión.

RTP es normalmente integrado dentro de la aplicación, sus paquetes son normalmente encapsulados dentro de paquetes UDP a los cuales proporciona checksum y multiplexación. Los paquetes RTP consisten normalmente de un header y datos de 75 payload los cuales pueden ser frames de video o varias muestras de audio; algunos campos en el header pueden ser:

- **Payload type:** Identifica el formato del payload RTP, ejemplo H.261 para video.
- **Sequence number:** El cual es incrementado por uno cada vez que un paquete es enviado. Normalmente usado para detectar pérdida de paquetes o paquetes fuera de secuencia.
- **Timestamp:** Es el instante en que el paquete de datos es generado. Usado en sincronismo y cálculo de jitter. RTCP es usado para transmisión periódica de paquetes de control a todos los participantes de la sesión. Una idea de su funcionamiento es indicada a seguir:
- **Monitoramiento de QoS:** Los informes de RTCP incluyen ciertas estadísticas de control, las cuales son enviadas como realimentación de la calidad de los datos distribuidos. Un tipo de informe de realimentación es llamado "receiver report" (RR), este es transmitido del receptor de vuelta al transmisor. El

contiene estadísticas de performance sobre número de paquetes perdidos, secuencia de paquetes máxima recibida, jitter y otras medidas de atraso para calcular el RTT.

- **Identificación:** RTCP transporta información de identificación acerca de los participantes de la sesión en otro tipo de informe llamado "Source description items" (SDES)

4.2.24 LAS RFCS DEL RTP/RTCP Y SUS PRINCIPALES ESPECIFICACIONES

Los estándares del IETF participantes en el tema de esta investigación fueron:

1. **RFC 1889**, Enero de 1996, "RTP: *A transport protocol for real-time applications*".
2. **RFC 1890**, Enero de 1996, "RTP: *Profile for audio and videoconferences with minimal control*".
3. **RFC 3550**, Julio de 2003, "RTP: *A transport protocol for real-time applications*"
4. **RFC 3551**, Julio de 2003, "RTP: *Profile for audio and videoconference with minimal control*".
5. **RFC 3611**, noviembre de 2003, "RTP: *Control protocol extended reports (RTCP- XR)*".
6. **RFC 4585**, Julio de 2006, "Extended **RTP** *profile for RTCP - based feedback (RTP/AVPF [Audio-Visual Profile with Feedback])*".
7. **RFC 4586**, Julio de 2006, "Real-time transport control protocol (**RTCP**) *based feedback: result of the timing rule simulation*".

Antes de la creación de la **RFC 4585**, las principales normas de la **RTP (RFC 1889 y 3550)** restringían el envío de información estadística, como se muestra a continuación:

1. Los mensajes se envían en paquetes **RTCP** compuestos que incluyen al menos un **SR** (*Sender Report*) o **RR** y un mensaje de **SDES** (*Source Description*)

Estos paquetes **RTCP** son enviados en intervalos de tiempo (**T-rr**) calculados en función del tamaño medio del paquete, número de transmisores y receptores en el grupo y del ancho de banda de la sesión (un porcentaje del ancho de banda de la sesión es asignada para los paquetes **RTCP**). Esta banda es dividida entre todos los miembros de la sesión y los transmisores pueden conseguir más banda que los receptores.

2. **RFC 3550**, Julio de 2003, "RTP: *A transport protocol for real-time applications*".
3. **RFC 3551**, Julio de 2003, "RTP: *Profile for audio and videoconference with minimal control*"
4. **RFC 3611**, noviembre de 2003, "RTP: *Control protocol extended reports (RTCP- XR)*".
5. **RFC 4585**, Julio de 2006, "Extended **RTP profile for RTCP - based feedback (RTP/AVPF [Audio-Visual Profile with Feedback])**".
6. **RFC 4586**, Julio de 2006, "Real-time transport control protocol (**RTCP**) based feedback: *result of the timing rule simulation*".

Antes de la creación de la **RFC 4585**, las principales normas de la **RTP (RFC 1889 y 3550)** restringían el envío de información estadística, como se muestra a continuación:

1. Los mensajes se envían en paquetes **RTCP** compuestos que incluyen al menos un **SR (Sender Report)** o **RR** y un mensaje de **SDES (Source Description)**
2. Estos paquetes **RTCP** son enviados en intervalos de tiempo (**T-rr**) calculados en función del tamaño medio del paquete, número de transmisores y receptores en el grupo y del ancho de banda de la sesión (un porcentaje del ancho de banda de la sesión es asignada para los paquetes **RTCP**). Esta banda es dividida entre todos los miembros de la sesión y los transmisores pueden conseguir más banda que los receptores.
3. El intervalo mínimo entre 2 paquetes **RTCP** de la misma fuente era de 5 segundos.

El objetivo de estas reglas fue prevenir una explosión de realimentación de estadísticas y escalar grupos *multicast* muy grandes. Todavía, se observa que principalmente la regla del intervalo mínimo puede impedir a los receptores de enviar realimentación en tiempo hábil, es decir, enviar fuera del instante en que es realmente necesario (es preciso esperar 5 segundos).

La **RFC** más reciente, la 4585, de julio de 2006, introduce cambios en el perfil **RTP** para audio y vídeo. Los principales cambios son:

1. Fue introducido el concepto de “early **RTCP** mensaje” que, actuando en conjunto con algoritmos adecuados, propicia realimentación con bajos retrasos.
2. Introducción de pequeños mensajes de realimentación que serán transmitidas en los “payloads” del **RTCP**.

En la verdad, esta **RFC** visa resolver las restricciones principalmente de “timing” especificadas en la **RFC 3550/3551**, pero manteniendo la mayoría de los algoritmos del **RTP** y **RTCP** originales, que han probado trabajar adecuadamente en la *Internet*. Los detalles pueden ser encontrados en la propia **RFC 4585**. A seguir, un resumen de los principales ítems de estas **RFCs** que permanecen válidos.

4.2.25 LA RFC 4585 (AVPF) Y SUS PRINCIPALES CARACTERÍSTICAS

Esta **RFC** prácticamente corta las limitaciones existentes hasta antes de su creación. Los cambios introducidos en el nuevo perfil del **RTP-AVPF** pueden ser resumidos básicamente en dos:

1. El intervalo mínimo de 5 segundos es abolido.
2. Los receptores pueden, durante el intervalo, enviar una única vez un paquete **RTCP** llamado “early *packet*”, que no necesita esperar que el “intervalo calculado” transcurra para enviar estadísticas, si no que puede transmitir prácticamente de inmediato. Es importante notar que el intervalo de envío de las estadísticas **RTCP** se mantiene calculado de acuerdo con su forma original detallada en la **RFC 1889**.

Durante la operación normal, todos los miembros del grupo, calculan el intervalo $T-r$ de acuerdo con las reglas especificadas del **RTP** original, sin que exista el intervalo mínimo de 5 segundos. Esto genera condiciones para que los algoritmos de control de congestión trabajen adecuadamente, mismo cuando se trata de pequeños grupos o *unicast*. Las estadísticas son muy frecuentes y no es necesario esperar por los 5 segundos inicialmente definidos. Esto significa, por ejemplo, que se un receptor detecta la presencia de pérdida de paquetes, puede enviar muy rápido cuanto posible, paquetes de "realimentación" por medio de un "early packet" sin tener que esperar el tiempo regularmente programado.

4.2.26 EL ALGORITMO DE GENERACIÓN DEL "EARLY PACKET"

De modo general, el algoritmo para generación del "early packet" funciona de la siguiente forma: Si el receptor necesita enviar la "realimentación" en el tiempo $T-0$ primero verifica si el paquete **RTCP** anterior ya fue enviado como "early packet" o si otro "early packet" ya está programado para transmisión. En caso afirmativo, no puede enviar otro paquete de este tipo, pero puede adjuntar la "realimentación" en el próximo paquete **RTCP** normalmente programado. Si aún no fue enviado ningún **RR**, programa un "early packet", calculando el llamado "maximum dithering interval".

4.2.27 CÁLCULO DEL “MAXIMUM DITHERING INTERVAL” (T-DITHER-MAX)

1. Si es una sesión *unicast* (tamaño del grupo = 2), **T-dither-max** = 0. Esto debido a que no tiene sentido un receptor *unicast* esperar por otros receptores si ellos tienen la misma información para enviar.
2. Si el receptor tiene un valor de **RTT** (conocido) hasta el transmisor de la media, entonces el **T-dithermax** = $K \times T\text{-rtt}/2 \times \text{Members}$ (con $K = 1$, **T-rtt** indicando el *round trip time*).
3. Si el receptor no tiene un valor de **RTT** (conocido) hasta el transmisor, entonces:

Si el próximo paquete **RTCP** regularmente programado está dentro del intervalo **T-0** a **(T-0 + Tdither-max)**, no puede ser programado un “*early packet*”, pero las estadísticas son adjuntadas en el próximo paquete **RTCP**. Otro “*dithering interval*”, **T-dither** es escogido al azar del intervalo $[0; \text{T-dither-max}]$. El “*early packet*” es programado para $T\text{-e} = T\text{-0} + \text{T-dither}$.

De modo general, los “*early packets*” también son paquetes **RTCP** compuestos, pero pueden no incluir mensajes **SDES**, por tanto, consistirían sólo en mensajes **SR** o **RR** y el mensaje de “realimentación”.

Después del envío del “*early packet*”, un *flag* “*allow_early*” es hecho igual a “**FALSE**”, para prevenir que la misma fuente envíe dos “*early packets*” consecutivos. Después que el próximo paquete **RTCP** normalmente programado es

segundo (*bytes*). Deberá ser especificada como una fracción del parámetro “banda de la sesión”.

- *We_sent* = *flag* que es “TRUE” si la aplicación envió datos desde que el segundo informe anterior fue transmitido.
- **AVG-RTCP-size** = el tamaño medio del paquete **RTCP** compuesto, en octetos, por todos los paquetes **RTCP** enviados y recibidos por este participante. El tamaño incluye encabezados de la capa baja de los protocolos de transporte y de red (**UDP** y **IP**). 82
- *Initial* = *flag* que es “TRUE” si la aplicación aún no envió ningún paquete **RTCP**. La mayoría de estos parámetros utiliza el “intervalo calculado” **T** para la transmisión de paquetes y será detallado en el ítem siguiente.
- **4.2.29 CÁLCULO DEL INTERVALO DE TRANSMISIÓN DEL RTCP (RFC 1889/3550)**

Para mantener escalabilidad, el intervalo medio entre paquetes de un participante de la sesión deberá escalar con el tamaño del grupo. Este intervalo es llamado de “intervalo calculado” **T**, y es determinado de la siguiente forma:

$$T (Tr) = \frac{n(C)}{0.25(\text{banda RTCP})}$$

$$T (rx) = \frac{n(L)}{0.75(\text{banda RTCP})}$$

Donde:

T (Tr): T de los transmisores.

T (rx): T de los receptores.

1. Si el número de transmisores es menor o igual que 25% de los miembros del grupo, el intervalo va depender si el participante es transmisor o no. Si es transmisor, la constante "C" es considerada como el tamaño medio del paquete **RTCP (AVG-RTCP- size)** dividido por 25% del ancho de banda **RTCP (RTCP_BW)**, y la constante "n" representa el número de transmisores. Si el participante es receptor, la constante "C" es considerada como el tamaño medio del paquete **RTCP** dividido por 75% del ancho de banda del **RTCP**. La constante "n" es considerada como el número de receptores (miembros - transmisores). Si el número de transmisores es mucho más grande de que 25%, transmisores y receptores son considerados conjuntamente.
2. La constante "C" es calculada como el tamaño medio del paquete **RTCP** dividido por la banda total y "n" es el número total de miembros. Y, de acuerdo con el indicado 3.5 83 en el ítem 1, un perfil **RTP** puede especificar que una banda **RTCP** puede ser explícitamente definida por 2 parámetros separados llamados "S" y "R", tanto para aquellos participantes que son transmisores cuanto para los que no lo son. En este caso, la fracción de 25% se convierte en $S/(S + R)$ y la fracción de 75% se convierte en $R/(S + R)$. Mire que si $R = 0$, la porcentaje de transmisores nunca será más grande que $S/(S + R)$ y la implementación debe evitar la división por cero;
3. Si el participante aún no envió un paquete **RTCP** (la variable *Initial* es "TRUE") la constante **Tmin** es configurada en 2,5 segundos; de otra manera, será configurada en 5 segundos.

4. El intervalo determinístico calculado T_d es configurado para $T_d = \max(T_{min}, n \times C)$.
5. El intervalo calculado, T , es fijado en un número uniformemente distribuido, entre 0,5 y 1,5 veces el intervalo calculado determinísticamente T_d .
6. El valor resultante de T es dividido por $(e - 1,5) = 1,21828$ para compensar la convergencia del algoritmo "timer *reconsideration*" para un valor de la banda **RTCP abajo** del valor medio.

Este procedimiento resulta en un intervalo al azar, pero que, en la media, destina en el mínimo 25% de la banda **RTCP** para los transmisores y el restante para los receptores. Si los transmisores constituyen más de un cuarto de los miembros, este procedimiento divide la banda igualmente entre todos los participantes.

4.2.30 INICIALIZACIÓN DE LOS PARTICIPANTES DE LA SESIÓN

Al unirse a sesión, el participante inicializa sus parámetros de la siguiente forma: $T_p = 0$, $T_c = 0$, $Senders = 0$, $Pmembers = 1$, $Members = 1$, $We_sent = "FALSE"$, $RTCP_BW = a$ fracción especificada de la banda de la sesión, $Initial = "TRUE"$, $AVG\ 84\ RTCP\text{-}size =$ al tamaño probable del primero paquete **RTCP** que la aplicación construirá después.

Con estos datos es calculado el intervalo T , después de esto, el primero paquete es programado para el tiempo $T_n = T$. Esto significa que un *timer* de transmisión es fijado y expira en el tiempo T (mire que la aplicación puede usar el foco que desear para implementar este *timer*).

El participante agrega su propio **SSRC** (*Synchronization Source*) en el cuadro de miembros.

4.2.31 RECIBIENDO UN PAQUETE RTP O RTCP QUE NO ES DEL TIPO “BYE”

Cuando un paquete **RTP** o **RTCP** es recibido de un participante cuyo **SSRC**, no está registrado en el cuadro de miembros, su **SSRC** es agregado al cuadro y el valor de los miembros es actualizado después del participante haber sido validado. El mismo procedimiento es seguido para cada **CSRC** (*Computer Security Resource Center*) cuando un paquete RTP es validado. Cuando un paquete RTP es recibido de un participante cuyo SSRC no está en el cuadro de transmisores, el SSRC es agregado al cuadro y el valor de los transmisores es actualizado.

Para cada paquete RTCP compuesto recibido, el valor del AVG-RTCP-size es actualizado de la siguiente forma:

$$AVG-RTCP-size(i) = \left(\frac{1}{16}\right) \times packet-size + \left(\frac{15}{16}\right) \times AVG-RTCP-size(i-1)$$

Donde: *Packet-size* = tamaño del paquete RTCP recientemente recibido.

4.2.32 RECIBIENDO UN PAQUETE RTCP DEL TIPO “BYE”

Si fuera recibido un paquete RTCP del tipo “BYE”, el SSRC es verificado en el cuadro de miembros. Si está presente, la entrada es removida del cuadro y el valor de 85 los miembros es actualizado. Después, el SSRC es verificado en el cuadro de transmisores. Si está presente, la entrada es removida del cuadro y el valor de los transmisores es actualizado.

4.2.33 TIME OUT DE UN SSRC

A cada cierto tiempo, los participantes deben verificar si algún de los otros participantes temporizó "time out". Para hacer esto, el participante calcula el intervalo determinístico T_d , (sin el factor al azar) para un receptor que está con $We_sent = "FALSE"$. Cualquier otro miembro de la sesión que no tenga enviado un paquete RTP o RTCP desde el tiempo $T_c - MT_d$ (M es el multiplicador *time out*, con *default* igual a 5 es considerado "timed out"), esto significa que su SSRC es eliminado de la lista de miembros y el cuadro *Members* es actualizado. Una verificación similar es efectuada en la lista de transmisores.

Cualquier miembro de la lista de transmisores que no tenga enviado un paquete RTP desde el tiempo $T_c - 2T$ (dentro de los 2 últimos intervalos de informes RTCP) es eliminado de la lista de transmisores y el cuadro es actualizado.

Si cualquier miembro es considerado "timed out", el algoritmo de reconsideración reversa, descrito arriba, debería ser ejecutado. El participante debe efectuar esta verificación lo mínimo una vez por intervalo de transmisión RTCP.

4.2.34 FINALIZACIÓN DEL *TIMER* DE TRANSMISIÓN

Cuando expira el *timer* del paquete de transmisión, el participante efectúa las siguientes operaciones:

1. El intervalo de transmisión T es calculado de acuerdo con el que fue descrito en la sección 5.3.1 de la **RFC 3550**, incluido el factor al azar.

2. Si $T_p + T = T_c$, es transmitido un paquete **RTCP**. $T_p = T_c$, y otro valor para T es calculado de acuerdo con el ítem arriba, $T_n = T_c + T$.

El *timer* de transmisión expira en el tiempo T_n . Si $T_p + T$ es más grande que T_c , $T_n = T_p + T$ y ningún paquete **RTCP** es transmitido.

3. *Pmembers* es configurado igual a *Members*.

Si un paquete **RTCP** es transmitido, el valor *Initial* = "FALSE". Consecuentemente, el valor de **AVGR-TCP-size** es actualizado de la siguiente forma:

$$\text{AVG-RTCP-size } (i) = \left(\frac{1}{16}\right) \times \text{packet-size} + \left(\frac{15}{16}\right) + \text{AVG-RTCP-size } (i-1)$$

Donde: *packet-size* es el tamaño del paquete **RTCP** recientemente transmitido.

4.2.35 TRANSMITIENDO UN PAQUETE *BYE*

Cuando un participante desea terminar una sesión, transmite un paquete *BYE* para informar a los otros participantes de este hecho con el objetivo de evitar una avalancha de paquetes *BYE* cuando muchos participantes desean dejar el sistema, un participante debe ejecutar el siguiente algoritmo se desea dejar el sistema cuando el número de miembros fuera más grande que 50. Cuando el participante decide dejar el sistema, $T_p = T_c$, el tiempo en ese instante, *Members* y *Pmembers* son inicializados para 1, "Initial" es configurado en *We_sent* = "FALSE", *senders* es configurado para 0 y **AVG-RTCP-size** = tamaño del paquete compuesto *BYE*. El intervalo calculado T es calculado. El paquete *BYE* es programado para el tiempo $T_n = T_c + T$.

Cada vez que un paquete *BYE* de otro participante es recibido, el campo *Members* es incrementado de un, sin importar si el participante existe en el cuadro de miembros o no, y cuando el muestreo de SSRC está siendo utilizado, independiente del *BYE* SSRC ser incluido en la muestra o no. *Members* no es incrementado cuando otros paquetes RTCP o RTP son recibidos, pero apenas cuando son paquetes *BYE*. Del mismo modo, *AVG-RTCP-size* es actualizado solamente cuando son recibidos paquetes *BYE*. *Senders* no es actualizado cuando llegan paquetes RTP, permanece en 0.

A partir de este instante, la transmisión del paquete *BYE* sigue las reglas para transmitir un paquete RTCP regular, como indicado arriba.

Este procedimiento permite que los paquetes *BYE* sean transmitidos inmediatamente, controlando su uso total de banda. En el peor de los casos, esto provocaría que paquetes de control RTCP usasen el doble de la banda normal (10%), 5% para paquetes RTCP *BYE* y 5% para paquetes de otro tipo. Un participante que no desee esperar por el procedimiento indicado para la transmisión del paquete *BYE* puede dejar el grupo sin la transmisión de ningún paquete *BYE*. Probablemente, este participante será *timed out*, por los otros miembros del grupo.

Si el tamaño del grupo de miembros es estimado en menos de 50 cuando el participante decide salir, el participante puede enviar un paquete *BYE* inmediatamente o, si desear, puede ejecutar el algoritmo indicado en el ítem anterior. En cualquier de los casos, el participante que nunca envió un paquete RTP o RTCP no debe enviar un paquete *BYE* cuando fuera dejar el grupo.

4.2.36 ACTUALIZACIÓN DE *WE_SENT*

La variable *We_sent* contiene el valor "TRUE", si el participante envió un paquete RTP recientemente, o un valor "FALSE", en el caso contrario. Si el participante envía un paquete RTP cuando *We_sent* está configurado como "FALSE", el se incluye sólo en el cuadro de transmisores y configura *We_sent* para "TRUE". El algoritmo de "reconsideración reversa" debería ser ejecutado para reducir el retraso antes de enviar un paquete SR. Cada vez que otro paquete RTP es enviado, el tiempo de transmisión de ese paquete es mantenido en el cuadro y el algoritmo normal de *time out* del transmisor es aplicado.

Si un paquete RTP no fue transmitido desde el tiempo $T_c - 2T$, el participante se elimina sólo del cuadro, disminuí el valor del mínimo de transmisores y configura *We_sent* para "FALSE".

4.2.37 ASIGNACIÓN DE BANDA EN EL SDES

Esta especificación define varios ítems SDES adicionales en el ítem *CNAME*, tales como *NAME* (nombre personal) y *EMAIL* (dirección de correo electrónico). Ella también proporciona un medio de definir nuevos tipos de paquetes RTCP para aplicaciones específicas. Todavía, será preciso tener mucho cuidado en la hora de asignar control de banda para estas aplicaciones, ya que puede reducir la tasa en que los informes de recepción y *CNAME* son enviados, perjudicando, así, la performance del protocolo.

Se recomienda que no más de 20% de la banda RTCP asignada para un único participante sea usada para transportar la información adicional. Por consiguiente, no se pretende que todos los ítems SDES sean incluidos en cada

aplicación. Para aquellos que son incluidos debería ser asignada una fracción de la banda, de acuerdo con su 89 utilidad y, en vez de estimar esas fracciones dinámicamente, es recomendable que los porcentajes sean estimados estáticamente dentro de los intervalos de los informes, basados en el tamaño típico de un ítem.

Por ejemplo, una aplicación puede ser designada para enviar solamente *CNAME*; *NAME* y *EMAIL* y nada más. Al *NAME* podría ser dada mayor prioridad que en el *EMAIL*, porque el *NAME* podría ser enseñado continuamente en la *interface* de la aplicación del usuario, mientras que el *EMAIL* podría ser enseñado sólo cuando solicitado. Para cada intervalo RTCP, un paquete RR y un paquete SDES con el ítem *CNAME* sería enviado.

4.2.38 INFORMES SR Y RR

Los receptores RTP proporcionan realimentación de la cualidad de la recepción utilizando informes de paquetes RTCP que pueden ser de dos formas, dependiendo si el receptor también es un transmisor o si es solamente receptor. La única diferencia entre las formas del informe de transmisor (SR) y el informe de recepción (RR), además del código del tipo, es que el informe de transmisor incluye una sesión de información del transmisor de 20 *bytes*, a ser usada por los transmisores activos.

El SR es generado si un *site* enviar cualquier paquete de datos durante el intervalo de creación del último informe o informe previo, caso contrario, un informe RR es creado.

Ambas las formas, SR y RR, incluyen cero o más bloques en el informe de recepción, un para cada una de las fuentes de sincronismo de las cuales este receptor recibió paquetes de datos RTP desde el último informe.

Cada bloque del informe de recepción proporciona estadísticas sobre los datos recibidos de la fuente indicada en ese mismo bloque. Como 31 bloques de informe RR caben en un paquete SR o RR, los paquetes RR adicionales deberán ser apilados después de los paquetes enviados SR o RR, segundo las necesidades.

Los paquetes RR adicionales deberían ser apilados después del SR o RR inicial, de acuerdo con la necesidad de contener los informes de recepción para todas las fuentes presentes durante el intervalo desde el último informe.

Si existen muchas fuentes para completar todos los informes de paquetes RR en un paquete RTCP compuesto sin exceder al MTU (*Maximum Transmission Unit*) del enlace de la red, entonces solamente el subconjunto que cabe en una MTU deberá ser incluido en cada intervalo. Los subconjuntos deberán ser seleccionados de la forma *round-robin* por medio de intervalos múltiples de forma que todas las fuentes sean informadas.

4.2.39 EJEMPLO DE APLICACIÓN DEL CÁLCULO DE BANDA DEL RTCP

El RTCP intenta limitar su tráfico a 5% de la largura de banda de la sesión. Por ejemplo, suponga que una fuente está se comunicando con otra a una velocidad de 95200 bps (G.711). Entonces, el RTCP tiende a limitar su tráfico a $0,05 \times 95200 = 4760$ bps. De este valor, 75% (3570 bps) son asignados para los receptores y el restante es asignado para el transmisor. Los

3570 bps son igualmente distribuidos entre los receptores. Si fueran 2 receptores, cada uno recibirá $3570 \text{ bps} / 2 = 1785 \text{ bps}$. El transmisor utiliza 1190 bps para transmitir tráfico RTCP. De esta forma, los participantes determinan el período de transmisión de paquetes RTCP calculando el paquete RTCP medio, durante la sesión entera, y dividiendo por su tasa asignada.

En resumen:

- Período para transmisión de paquetes **RTCP** por un transmisor es:

$$T1 = \frac{\text{número de transmisores}}{25 \times 0,05 \text{ banda de sección}} (\text{AVG} - \text{RTCP} - \text{packet size}) \quad 4.9$$

- El período para transmisión de paquetes **RTCP** por un receptor es:

$$T2 = \frac{\text{número de receptores}}{75 \times 0,05 \text{ banda de sección}} (\text{AVG} - \text{RTCP} - \text{packet size}) \quad 4.10$$

- Para **AVG-RTCP-packet size = 238 bytes**:

$$T1 = \frac{1 \times 238 \text{ bytes} \times 8 \text{ bits/bytes}}{0,25 \times 0,05 \times 95200} = 1,6 \text{ segundos (transmisores)} \quad 4.11$$

$$T2 = \frac{1 \times 238 \text{ bytes} \times 8 \text{ bits/bytes}}{0,75 \times 0,05 \times 95200} = 0,42 \text{ segundos (receptores)} \quad 4.12$$

4.2.40 FORMATO DE LOS PAQUETES RTCP

En general, el RTP y el RTCP tienen formato parecido. En el RTCP los formatos varían, entre otras cosas, en función del objetivo de cada uno de ellos; específicamente con relación al tipo de "payload", están definidos 5 tipos: SR = 200, RR = 201, SDES = 202, BYE = 203 y APP (*Application Defined Packet*) = 204. Los formatos de mayor interés de este tema son básicamente 2, el SR y el

RR y permiten, que cuando son implementados, proporcionan las siguientes estadísticas:

- Retraso del paquete fin-a-fin.
- Número de paquetes recibidos.
- Tasa de paquetes perdidos.
- Número de paquetes esperados.
- Fracción de paquetes perdidos.
- Tasa de pérdida de paquetes por segundo.
- Validación estadística de las estimativas de pérdida de paquetes.
- *Throughput* aparente disponible para los receptores.
- *Interarrival jitter*.

En las tablas 4.11, 4.12, 4.13 y 4.14 están detallados los formatos de los paquetes RTP, SR, RR y SDES **respectivamente**.

TABLA 4.11

FORMATO DEL PAQUETE RTP

V	P	X	CC	M	PT	Sequence number
Timestamp						
Synchronization source (SSRC) identifier						
Contributing source (SSRC 1) identifier						
...						
Contributing source (SSRC n) identifier						
PAYLOAD						

TABLA 4.12 FORMATO DEL REPORTE DE TRANSMISIÓN

V	P	X	RC	PT – 200	Length
SSRC of the sender					
NTP timestamp (MSB)					
NTP timestamp (LSB)					
RTP timestamp					
Sender's packet count					
Sender's octet count					
First reception report block (SSRC 1)					
...					
Lost reception report block (SSRC n)					

TABLA 4.13 FORMATO DEL REPORTE DE RECEPCIÓN

V	P	RC	PT – 201	Length
SSRC of the sender				
SSRC of the first source				
Fract. Lost			Cum. nº of packet lost	
Ext. Highest sequence number received				
Interarrival jitter estimate				
Lost sender report timestamp (LSR)				
Delay since lost sender report (DLSR)				
...				
Lost reception report block				

TABLA 4.14 FORMATO DE DESCRIPCIÓN DE LA FUENTE

V	P	SC	PT – 202	Length
SSRC/CSRC of the sender				
Type			Length	text
Fract. Lost			Cum. nº of packet lost	
Text continued				
...				
Lost chunk				

4.2.41 CÁLCULO DE LAS MÉTRICAS DE QOS Cálculo del retraso fin-a-fin (RTT)

De acuerdo con la RFC 3550 (Sesión 6.4.1) el RTT es calculado por la

Ecuación:

$$D = A - LSR - DLSR \quad 4.13$$

- **D = RTT**
- **A** = instante de tiempo en que el **RR** es recibido por el *sender*
- **LSR** (*Last Sender Report*) = *last timestamp received from this source*, tomado del **RR**
- **DLSR** (*Delay since Last Sender Report*) = *delay since the receiving the LSR from this source*, tomado del **RR**

4.2.42 EJEMPLO DE CÁLCULO DEL RTT

Datos colectados con *Wireshark*:

TIMESTAMP

[10 NOV 1995 11:33:25.125 UTC]

> *SENDER REPORT*

- > **MSW:** 0xb44db705 = 3024992005 **seg**
- > **LSW:** 0x20000000 = **LSW/16** = 1.25 **seg**
- > *Sender's packet count* = 2
- > *Sender's octet count* = 320
- > **RECEIVER REPORT** [10 **NOV** 1995 11:33:36.5 **UTC**]
- > *Cummulative number of packets lost* = 0
- > *Extended highest sequence number received* = 0
- > *Interarrival jitter* = 0
- > **Last SR timestamp (LSR = 46853.125 seg.)**
- > **Delay since last SR timestamp (DLSR) = 5.250s**

Utilizando la Ecuación:

$$D = A - LSR - DLSR \quad 4.14$$

- **NTP timestamp del SR** (10 NOV 1995 11:33:36.125) = 3024992005.125 **seg**
- **LSR = 46853.125 seg**
- **DLSR = 5.250 seg**
- **A = 46864.500 seg;**
- **RTT = 46864.500 - 46853.125 - 5.250 = 6.125 seg**
- **Retraso = 6.125 seg.**

Alternativamente:

- **RTT = Timestamp del RR (UTC [Universal Time Coordinated]) Timestamp del último SR (UTC) – DLSR**
- **RTT = 11:33:36.500 - 11:33:25.125 – DLSR**
- **RTT = 11.375 - 5.25 = 6.125 seg. Packets lost = 3**

- $Packets\ lost = 3$

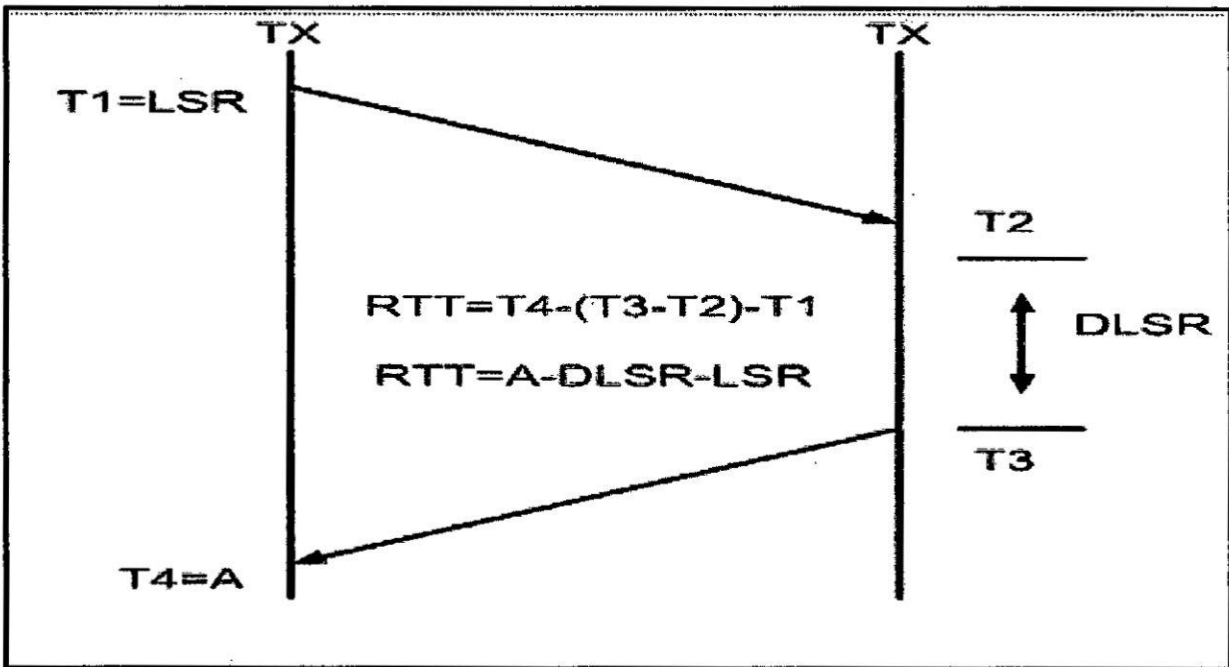
- $IPLR = \frac{3}{50} \times 100 = 6$

4.15

De forma equivalente:

La Fig. 4.5 resume gráficamente los parámetros envueltos en los cálculos arriba indicados.

Fig.4.5
CÁLCULO DEL IPDT



4.2.43 CÁLCULO DEL JITTER (IPDV)

La definición de "jitter" es bastante compleja. Algunos investigadores definen el *jitter* como la diferencia entre el máximo y el mínimo valor durante la transmisión de paquetes dentro de un cierto tiempo. Otros, como la máxima diferencia entre el retraso de la transmisión de dos paquetes consecutivos durante un período de tiempo. Aquí se utilizará la definición de la recomendación

RFC 1889 (el desvío medio de la diferencia **D** del espaciado de 2 paquetes consecutivos en el lado receptor comparado al lado transmisor). El *jitter* ocurre cuando paquetes consecutivos encuentran diferentes retrasos en las filas de los dispositivos de la red. Los retrasos dependen del modelo de servicio de cada fila y del "tráfico interferente" en el enlace de transmisión.

Otra fuente de *jitter* puede ser el servidor, durante la codificación de los datos *real-time*. Pasado un tiempo significativo entre la codificación del paquete y el envío del paquete para la red, debido, por ejemplo, el caso de la **CPU** (*Central Processing Unit*) del 97 servidor estar ocupada, estos cálculos del *jitter* pueden no ser válidos. El *jitter* generado por el servidor puede llevar a estimativas erróneas de las condiciones de la red. Testes de laboratorio indican que el *jitter* sólo no es una herramienta muy confiable para evaluar las condiciones de la red. Para obtener una estimativa confiable de las condiciones de la red, se recomienda que sea utilizado en conjunto con otros parámetros como, por ejemplo, "packet loss".

La estimación del *jitter* en el **RTCP** es hecha observando el tiempo de tránsito entre 2 paquetes consecutivos.

Si en los paquetes *i* y *j* son colocados los *timestamps* **Si** y **Sj**, cuando son enviados y son recibidos en los tiempos **Ri** y **Rj** respectivamente, entonces la Ecuación del cálculo $D = (Rj - Ri) - (Sj - Si)$ es la diferencia de tiempos de tránsito en unidades de *timestamps*. Los *timestamps* de envío **Si** y **Sj** son aproximados del *timestamp* del encabezado del paquete **RTCP** y **Ri** y **Rj** son obtenidos

restauración correspondiente. La **QoS** será calculada con la Ecuación 4.1. Sabiendo también que las pérdidas de los paquetes son originadas por dos motivos principales: 1) *overflow* de *buffer* en el enlace congestionado; 2) debido a *bits* errados, y que en la mayoría de las redes la probabilidad de existencia de *bits* errados es muy pequeña, suponemos que la pérdida de paquetes ocurre principalmente por congestión, en esas circunstancias, una reducción rápida de la banda utilizada podría reducir la pérdida de grande cantidad de paquetes, por tanto, la degradación de la calidad de la comunicación.

4.2.46 EL SISTEMA DE QoS

El proceso total de monitoreo y control, es visualizado de la siguiente forma: las informaciones recibidas en los informes del **RTCP** serán utilizadas como punto de partida para implementar un sistema de *software* que será llamado "Sistema de **QoS**", cuya función general es la implementación de un mecanismo "tipo" control de flujo que actuará sobre la tasa de transmisión del "sender", ajustándola de acuerdo a los valores calculados de **QoS**. La nueva tasa de transmisión será implementada vía cambio de *codec* para un otro de menor banda. Alternativamente podrá ser detallada la implementación cambiando el tamaño del "payload" para reducir la banda. Los *codecs* de audio y vídeo soportan bien este tipo de ajuste, pues ellos usan *frame sizes* de longitud fija, de forma que la tasa de transmisión puede ser alterada fácilmente por 100 medio del cambio del *codec*, "on the fly". Se estima que después de 2 **RR** recibidos con **QoS** válida (dentro de los límites), la operación regresará a condiciones normales de operación (regreso hasta el *codec* "default"), en la secuencia inversa de *fallback*.

En resumen, el algoritmo de **QoS** a ser implementado funcionará de la forma indicada a continuación:

1. Los receptores envían sus informes para el servidor de origen (fuente) en la forma y frecuencia normal de operación.
2. El servidor (fuente) recibe esos informes y analiza las estadísticas en los campos mencionados a continuación y calcula las métricas **IPDT**, **IPDV** e **IPLR**, después calcula la **QoS** de la red y, a continuación, los compara con los valores límites pre- definidos.
3. Para el cálculo de la métrica **IPDT** serán utilizados los campos del **RR**, **LSR** (*timestamp* del último *sender report* recibido), **DLSR** (*delay* desde el último *sender report* recibido) y el parámetro "A" derivado del *timestamp* (**NTP**) del **RR** en el instante en que es recibido en el servidor y utilizando la Ecuación $RTT = A - DLSR - LSR$ calculase el **RTT** y, a continuación, calculase el **IPDT** = $(RTT/2)$ (retraso en un sentido).
4. Para el cálculo de la métrica **IPDV** utilizase el valor directo del campo "interarrival *time*" dividido por la tasa de muestreo del *codec* utilizado.
5. Para el cálculo de la métrica **IPLR** será utilizado el campo del **RR** "cumulative *number of packets lost*" y "extended *highest sequence number received*", de acuerdo con la Ecuación $IPLR (\%) = \frac{\text{cumulative number of packets lost}}{\text{extended highest sequence number received}} \times (100)$.
6. Son considerados 3 posibles estados de la red: 1) Normal cuando los parámetros de **QoS** están bajo de los límites; 2) cargada cuando está próxima de los límites; 3) congestionada cuando está por encima de los límites.

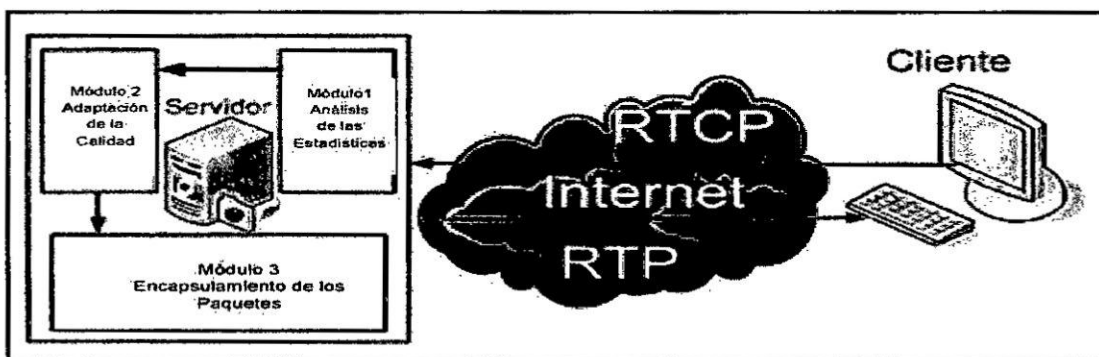
7. De acuerdo con la situación de la red, la aplicación multimedia podrá tener su banda aumentada (si regresara de una situación de congestión), mantenida (normal) o disminuida (si esta congestionada).

A continuación, son detalladas las características de cada uno de los 3 módulos componentes del Sistema de **QoS** para el modo de transmisión *unicast* (caso de la aplicación **VoIP**). La forma *multicast* podrá ser objeto de investigaciones futuras.

4.2.47 LOS MÓDULOS DEL SISTEMA DE QOS

La propuesta considera la implementación de un mecanismo de transmisión adaptativo basado en el modelo cliente-servidor con todas las actividades efectuadas solamente en el servidor y compuesto de 3 módulos: 1) Módulo de análisis de las estadísticas; 2) módulo de adaptación de la calidad; 3) módulo de encapsulamiento y transmisión de los paquetes en la nueva velocidad (Fig. 4.6).

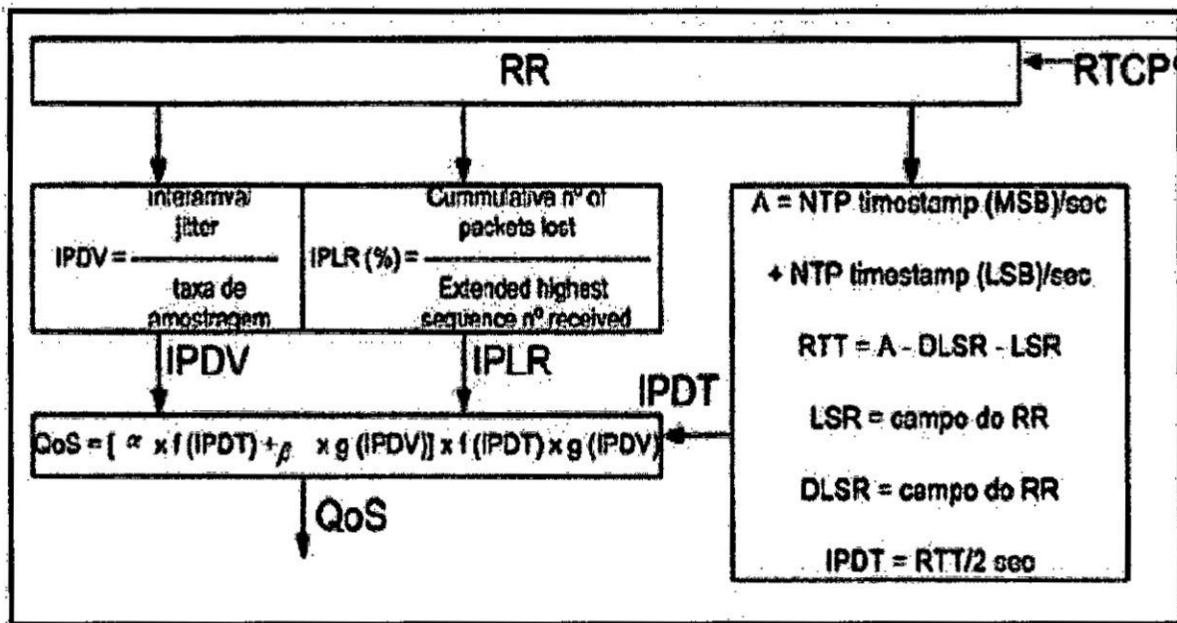
Fig. 4.6 LOS MÓDULOS DEL SISTEMA DE QOS



A. MÓDULO 1: ANÁLISIS DE LAS ESTADÍSTICAS

Su función es el monitoreo continuo de la red analizando los campos de los informes RR indicados en la Fig. 4.7 y envueltos en el cálculo de las métricas de IPDT, IPDV o IPLR. Después de calcular las métricas indicadas, será usada la Ecuación 4.1 para determinar los valores de QoS que serán pasados para el módulo de adaptación de la calidad para determinar las condiciones de la red.

FIG. 4.7 EL MÓDULO 1



* En el cálculo de QoS normalmente son considerados sólo las métricas más significativas IPDT e IPDV. El IPLR cuando aumenta es considerado como inicio de una situación de congestión. (Indicación de inicio de congestionamiento).

A. MÓDULO 2: ADAPTACIÓN DE LA CALIDAD

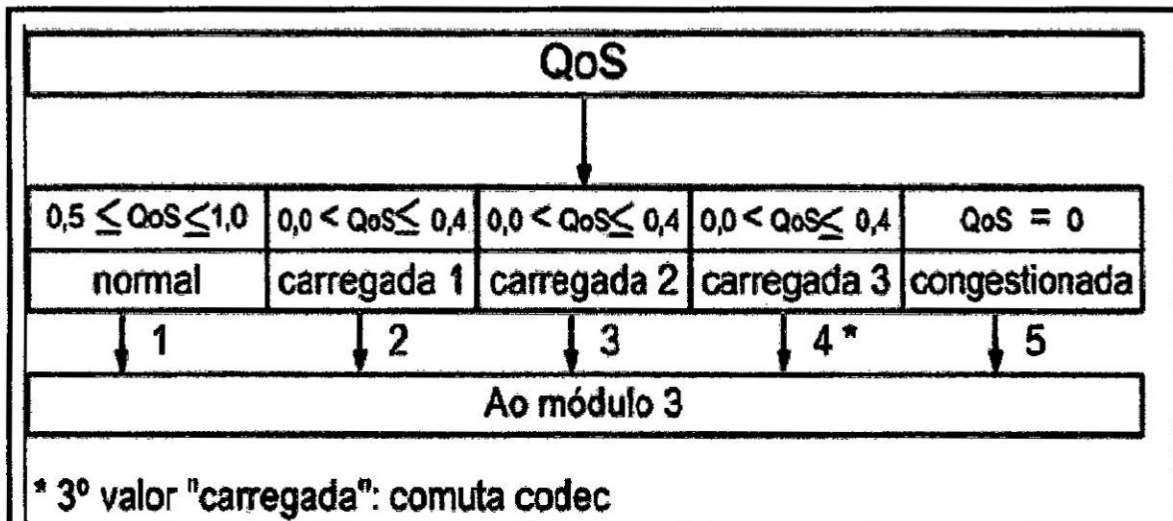
Es responsable por la identificación de las condiciones de la red a partir de la calidad de la voz en ese instante.

Este módulo está indicado en la Fig. 4.8 y recibe los valores de **QoS** del módulo 1 determinando el estado de la red dentro de una de las 3 alternativas siguientes:

1) Normal; 2) cargada; 3) congestionada.

Estos estados de la red son pasados para el módulo 3.

Fig. 4.8 EL MÓDULO 2



Si la opción "cargada" es aplicable, espera por un tercer valor si este fuera de la misma calidad (cargada 3) conmutará para un *codec* de menor banda. Caso contrario, mantendrá su condición en curso. Esto visa reducir efectos transitorios u oscilaciones en la red.

B. MÓDULO 3: ENCAPSULAMIENTO DE PAQUETES

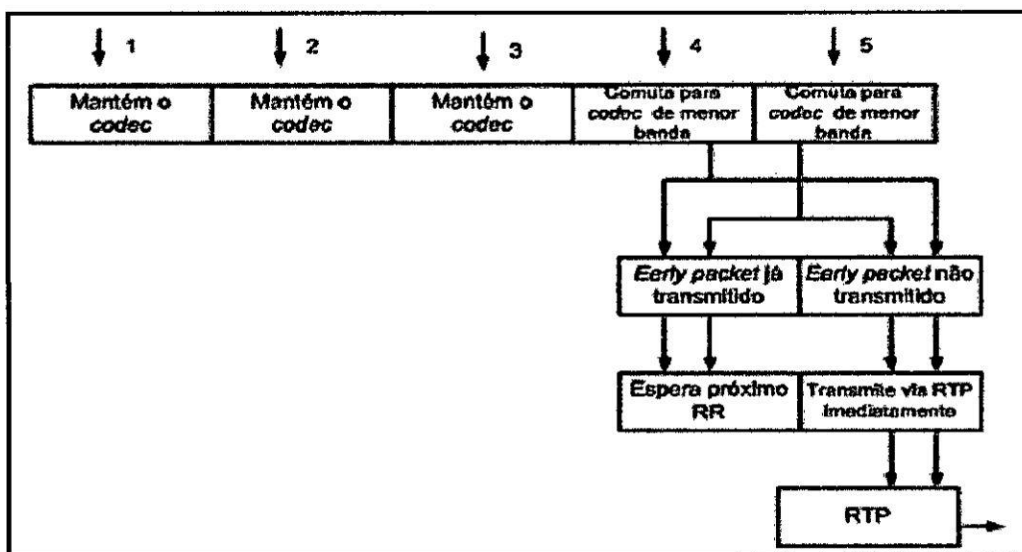
En función del estado de la red, este módulo indicado en la Fig. 4.9 actúa de la siguiente manera: 1) Mantiene el *codec* actual; 2) aumenta para un *codec* de mayor banda; 3) o disminuye (conmuta para un *codec* de menor banda).

De forma general el módulo funciona de la siguiente manera:

- Del módulo 2, recibe el valor del campo 1 (normal) significando que debe mantener el *codec* actual.
- De los campos 2 y 3 puede recibir los valores 1, 2 y 3 significando que debe mantener el *codec* en curso.
- Recibiendo un valor 4 (congestionada 3) deberá conmutar para un *codec* de menor banda, después de la confirmación del *early packet*. Recibiendo un valor mejor que los anteriores, deberá mantener el *codec* en curso.
- Recibiendo el valor 5 deberá efectuar la conmutación para un *codec* de menor banda inmediatamente (después de la confirmación del *early packet*).

Nota: La conmutación para un *codec* de menor banda implica en la verificación previa del módulo de la transmisión, si ya efectuada o no la transmisión de un *early-packet*. Si ya fue transmitido deberá esperar por el siguiente **RR**, caso contrario, será enviado inmediatamente.

FIG. 4.9 EL MÓDULO 3



4.2.48 LOS CODECS UTILIZADOS EN ESTE TRABAJO

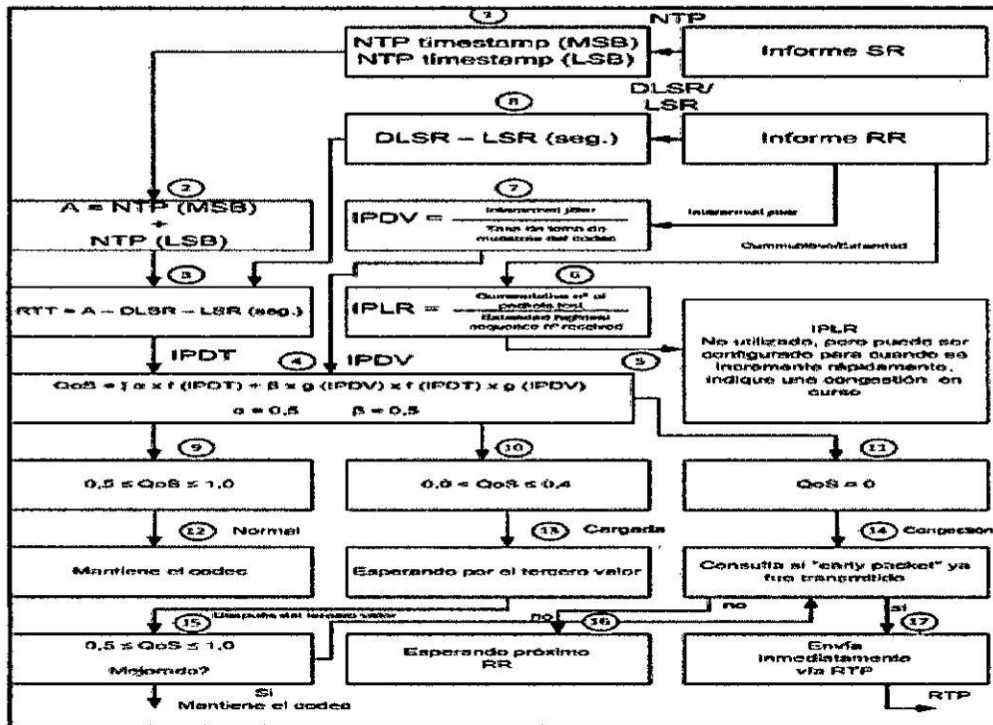
En esta investigación están siendo utilizados los siguientes *codecs*: 1) G.711; 2) G.729; 3) **GSM**; 4) G.723.1 A (**ACELP**). Esto representa una banda (utilizando protocolo de camada 2 **PPP** [*Point-to-Point Protocol*] o *frame relay*) de 81144 - 82800 **bps** para G.711, 45098 - 45600 **bps** para G.729, 31164 - 31800 **bps** para **GSM** y 17600 - 18190 **bps** para G.723.1 A.

Supongamos que inicialmente (*default*) será utilizado el *codec* G.711 (*toll quality*). La degradación de la calidad, medida por la **QoS** llevará hasta el "fallback" para G.729, después para **GSM** y, finalmente, para G.723.1 A, en esta secuencia.

Es bueno mencionar que la relevancia de esta investigación se basa en el procedimiento propuesto y no necesariamente en los *codecs* aquí utilizados, una vez que otros *codecs* podrán ser escogidos.

Se espera que después de transcurridos 2 informes consecutivos con métricas dentro de los límites, será iniciado el regreso para el *codec* "default" de la comunicación.

FIG. 4.10 EL MÓDULO DE QOS COMPLETO



La tabla 4.15 indica, de arriba hacia abajo, la secuencia de *fallback* prevista.

Tabla 4.15 SECUENCIA DE FALLBACK

Codec		Banda (bps)	Posición
1	G.711	83200	Default
2	G.729	46400	↙
3	GSM	32400	↙
4	G.723.1	18133	↙

4.2.49 EJEMPLO DE CÁLCULO DE LA BANDA NECESARIA Y MÍNIMA PARA EL

CODEC G.711

La banda necesaria con todos los encabezados para el protocolo *frame relay* es calculada de la siguiente manera:

- *Sample period* = 20 ms.
- *Packets/sec* = 50.
- *Payload* = 160 bytes.
- *Overhead* para protocolo *frame relay* = 48 bytes.
- Banda nominal necesaria = 83200 bps.
- Banda mínima para un IPLR de 3% = 0,03 x 50 *packets/sec* = 1,5 *packets* (aproximado a 2 *packets*).

Por tanto 2 paquetes representan:

$$2 \times (160 + 48) = 2 \times 208 = 416 \text{ bytes} \times 8 = 3328 \text{ bps} \quad 4.19$$

Banda mínima necesaria: 83200 - 3328 = 79872 bps.

4.2.50 EL ALGORITMO PROPUESTO

El pseudo code del algoritmo.

```
#include <iostream.h>
```

```
#include <stdio.h>
```

```
#include <iomanip.h>
```



```

#include <string.h> void main ()

{

int i, x = 0, valor, y = 0, z = 1, cn = 0, cc = 0, cg = 0, p = 0, r = 3, sw = 0, inicia = 0;
double qs, coded [] = {83200,46400,32400,18133};

double msb, lsb, a, dlsr, lsr, rtt, ipdt, ipdv, interarrival, _pdt, gipdv;

char qualidade [50];

strcpy (qualidade, "");

bool seguir = true; char resp;

while (seguir)

{

cout<<"ingressar MSB"

cin>>msb

cout<<"ingressar LSB"

cin>>lsb

cout<<"ingressar DLSR"

cin>>dlsr

cout<<"ingressar LSR"

cin>>lsr 108

```

```
cout<<"ingressar a"  
  
cin>>a;  
  
cout<<"ingressar interarrival";  
  
cin>>interarrival;  
  
rtt = a-dlsr-lsr;  
  
ipdt = rtt/2;  
  
printf ("O valor de a %10.10f ",a);  
  
cout<<endl;  
  
printf ("valor de rtt %10.10f ",rtt);  
  
cout<<endl;  
  
printf ("valor de ipdt %10.10f ",ipdt);  
  
cout<<endl;  
  
ipdv = interarrival/8000;  
  
_pdt = (150-ipdt)/150;  
  
gipdv = (50-ipdv)/50;  
  
qs = (0,5 * _pdt + 0,5 * gipdv) * (_pdt * gipdv);  
  
printf ("O valor de qs %10.10f ", qs);
```

```
cout<<endl;

// Processo

if (qs>= 0,5 && qs<= 1);

{

Strcpy (qualidade, "normal:");

Valor = coded [0]; i = 0; cn = 0; sw = 0;

}

if (qs>0 && qs<= 0,4);

{ Strcpy (qualidade, "carregada:");

l = 2; cc = 1;

if (inicia==1)

{ cn = 0; sw = 0;}

}

if (qs==0)

{ Strcpy (qualidade, "congestionada:");

l = 3; cg = 1; sw = 0;

}
```

```
if (sw==0) // sw=0, baixando

{

if (i==0) // normal

{

if (cc==1)

valor = coded [x];

}

if (i==2) // carregada anterior foi normal

{

if (cn==0)

{ Valor = coded [x];

Cn =1; inicia = 0; r = 3; p = 0;

} else

{ Valor = codec [x];

if (p==1)

{

X = x + 1;
```

```
Valor = codec [x];
```

```
P = -1;
```

```
}
```

```
P = p + 1;
```

```
If (x >= 3)
```

```
{ valor = coded [3]; sw = 1; x = 3; cg = 1;}
```

```
}
```

```
}
```

```
if (i==3) // congestionada e anterior foi normal ou carregada
```

```
{
```

```
if (cn==0 || cc==1)
```

```
{
```

```
x++;
```

```
Valor = coded [x];
```

```
if (x >= 3)
```

```
{Valor = codec [3]; sw = 1; x = 3;}
```

```
}
```

```
}  
  
}  
  
else // retrocede  
  
{  
  
If (cg==1)  
  
{  
  
Valor = coded [3];  
  
Cg = 0;  
  
}  
  
else  
  
{  
  
Valor = coded [r];  
  
if (z==2)  
  
{  
  
If (r==0)  
  
{valor = coded [0];}  
  
else
```

```
{  
  
R = r - 1;  
  
Valor = coded [r];  
  
Z = 0;}  
  
}  
  
z++;  
  
if (r==0)  
  
{inicia =1; x = r; sw = 0;}  
  
}  
  
X = r;  
  
}  
  
cout<<_< qualidade;  
  
cout<<_<valor;  
  
cout<<_<"\n";  
  
cout<<_<"Deseja seguir s/n:";  
  
cin>>resp;  
  
if (resp == 's')
```

```
    seguir = true;  
  
    else  
  
    seguir = false;  
  
    }  
  
    cout<<endl;  
  
    }
```


4.2.51. VALIDACIÓN DE LAS SIMULACIONES

En esta experiencia son validadas las simulaciones con tráfico interferente cuando se encaminan dos flujos simultáneos (voz y **TCP**) y (voz y **UDP**), en las velocidades ya indicadas y para cada uno de los *codecs* estudiados. La topología de testes está indicada en la Fig. 4.14 separada de la Fig. 4.12 para fines de una mejor visualización.

4.2.52 CON FLUJO INTERFERENTE UDP

Las tablas 4.22, 4.25, 4.28, 4.31, 4.34, 4.37 muestran los resultados de las pruebas efectuados con equipos "reales" para el caso de un flujo de voz

4.2.53 VALIDACIÓN DEL ALGORITMO PROPUESTO

Este fue validado de forma manual, de acuerdo con lo indicado en los ítems 4.2.6 y 4.2.7. 120 **4.2.66 LAS HERRAMIENTAS DE PRUEBAS** A seguir una breve descripción del *software* y *hardware* utilizado en las pruebas.

WIRESHARK

Es un *software* analizador de protocolos que soporta más de 700 tipos de protocolos. Es mantenido bajo licencia **GPL**: captura datos directamente de la red o de archivos previamente grabados. Trabaja en modo promiscuo o no promiscuo.

V. RESULTADOS

5.1 RESULTADOS DE LAS SIMULACIONES

5.1.1 UN FLUJO ÚNICO DE VOZ EN EL ENLACE WAN DE 128K- CODEC G.711

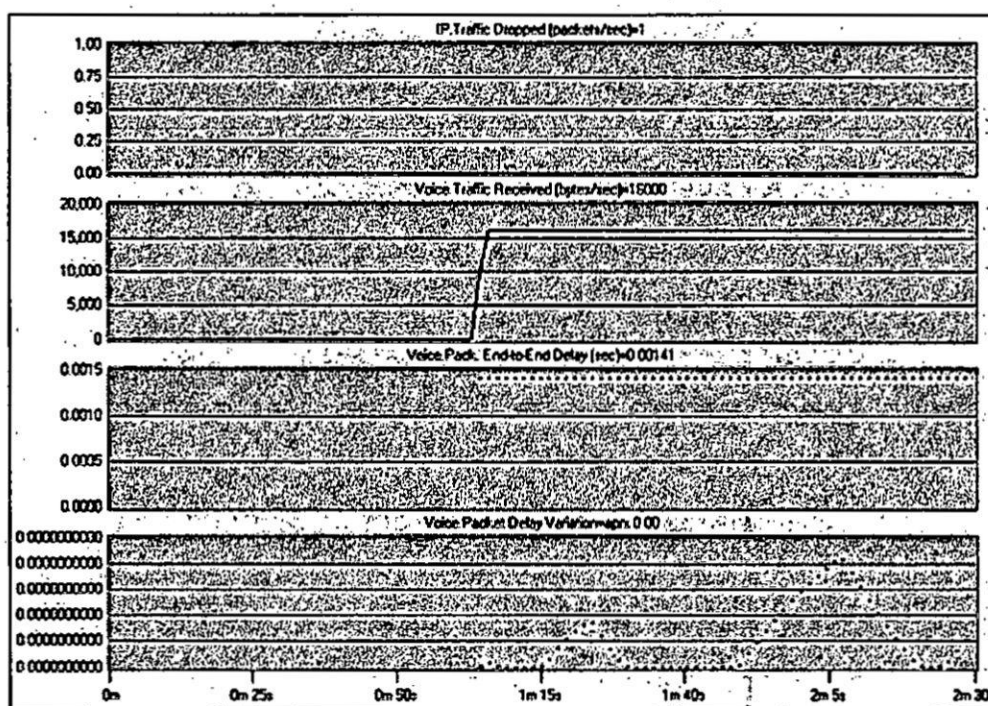
En las figuras y tablas siguientes son presentados los valores y gráficos de las métricas de **QoS**, **IPDT**, **IPDV** e **IPLR**, resultantes de la simulación de tráfico "único" de voz y su comportamiento cuando el enlace WAN es compartido con tráfico interferente **TCP (FTP)** o **UDP (vídeo)**, para cada uno de los *codecs* de voz utilizados en esta investigación.

Se utilizó la topología de la Fig. 4.11 con el enlace **WAN** configurado de 128 Kbps hasta (1.5) Mbps. Los gráficos resultantes reflejan el comportamiento de las métricas hasta esta velocidad, aunque son presentados como ejemplo solo en la velocidad de 128 **Kbps**. En los cuadros los datos están completos hasta 1.5 **Mbps**.

Las Figs. 5.1 a 5.12, como mencionado arriba muestran las características de las métricas de **QoS** de voz cuando los tres tipos de tráfico se encuentran compartiendo el enlace **WAN** (dos a dos, voz/**TCP** y voz/**UDP**), en la faja de 128 **Kbps** a **DS1 (1.544 Mbps)**. Estas figuras, inician mostrando sus valores cuando trafican solas en el enlace **WAN**, sin interferencia ninguna y como es esperado, es la mejor situación para comunicaciones, ella representa nuestra referencia inicial para este estudio visto que en esta situación el tráfico de voz es el único, utilizando todo el ancho de banda disponible y después es mostrado cuando es transmitido en conjunto con tráfico interferente **TCP** y cuando es transmitido en conjunto con tráfico **UDP**.

En cuanto a los títulos de los gráficos, Las figuras con el título "Métricas de QoS para Voz (GSM)" por ejemplo sin ninguna extensión adicional indica que este gráfico es la 124 referencia antes mencionada. Cuando es del tipo Voz (GSM) + UDP, significa que será evaluado el efecto del tráfico interferente UDP sobre los señales de voz con codec GSM. De la misma forma para el tráfico interferente TCP es mostrado en el grafico 5.2.

Fig. 5.1 MÉTRICAS DE QOS DE VOZ (G.711) EN ENLACE DE 128 KBPS

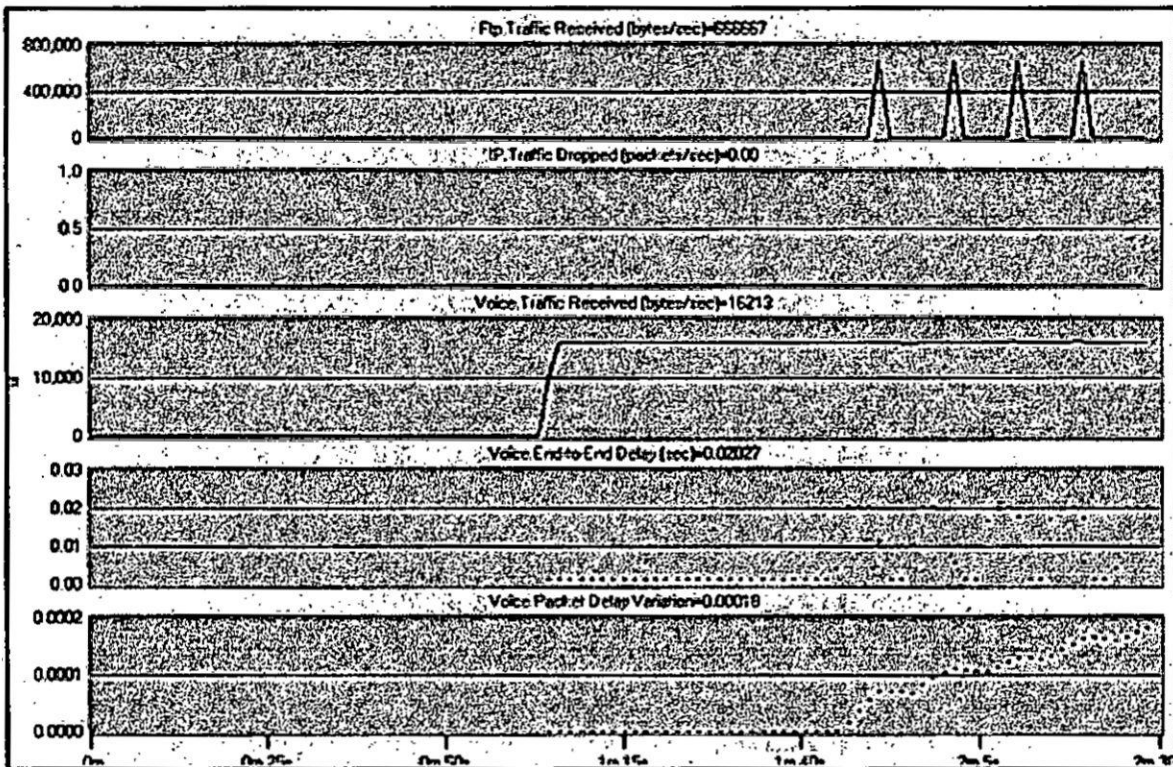


Observe en la Fig. 5.1 que el IPDV e IPLR son ambos iguales a cero como era de esperar, indicando que no existen pérdidas y el atraso E2E(Fin a Fin) tampoco existe o su valor es insignificante

5.1.2 DOS FLUJOS (VOZ+TCP) EN UN ENLACE DE 128 KBPS

Fig. 5.2

MÉTRICAS DE QOS PARA VOZ (G.711+TCP) EN ENLACE DE 128 KBPS

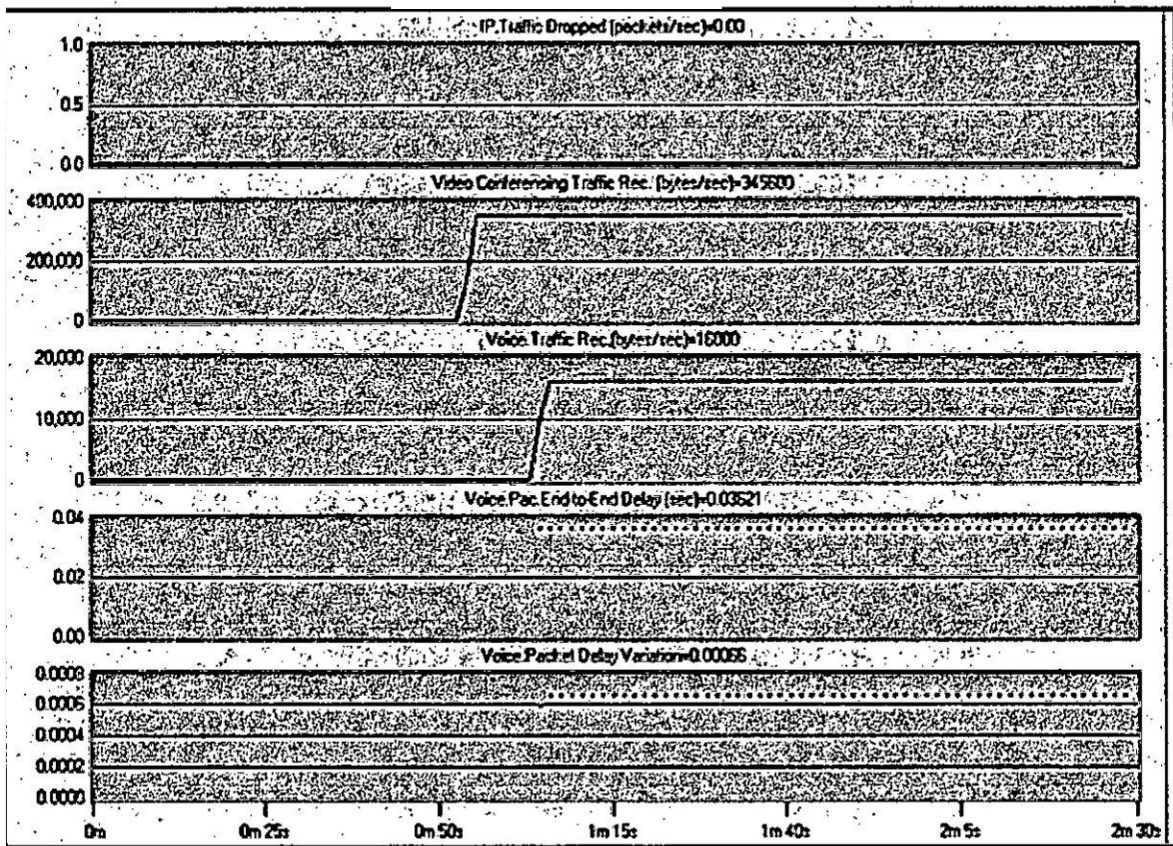


Los resultados muestran que: 1) No presenta pérdida de paquetes; 2) los valores del IPDT no exceden los límites establecidos; 3) el tráfico simultáneo ocupa toda la banda disponible en el canal **WAN**.

5.1.3 DOS FLUJOS (VOZ + UDP) EN EL ENLACE DE 128 KBPS-CODEC G.T11

La Fig. 5.3 muestra los resultados cuando en el enlace de 128 Kbps se encamina vídeo + voz, simultáneamente (tráfico interferente: videoconferencia por ejemplo). La siguiente figura. Muestra el tráfico de voz enviado de los 4 codecs en paquetes/seg.

Fig. 5.3 Métricas de QoS para Voz (G.711) + UDP en un enlace de 128K



Los resultados de las Fig. 5.1 a 5.3 muestran que para una banda de 128 Kbps, en la simulación efectuada y con las topologías indicadas, tiene las siguientes características.

1) No se registran pérdida de paquetes; 2) el retraso de voz se incrementa según indicado en la tabla 5.3; 3) el ancho de banda del enlace **WAN** ocupa toda la banda disponible cuando se comparte Voz (**G.711**) más tráfico interferente. Lo mismo es mostrado en las figuras. 5.4 a 5.11 para los codecs **G.729**, **GSM**, **G.723.1 A** más tráfico interferente.

Los resultados están indicados en las siguientes:

Fig. 5.4 MÉTRICAS DE QOS PARA VOZ (G 729) EN ENLACE DE 128-KBPS

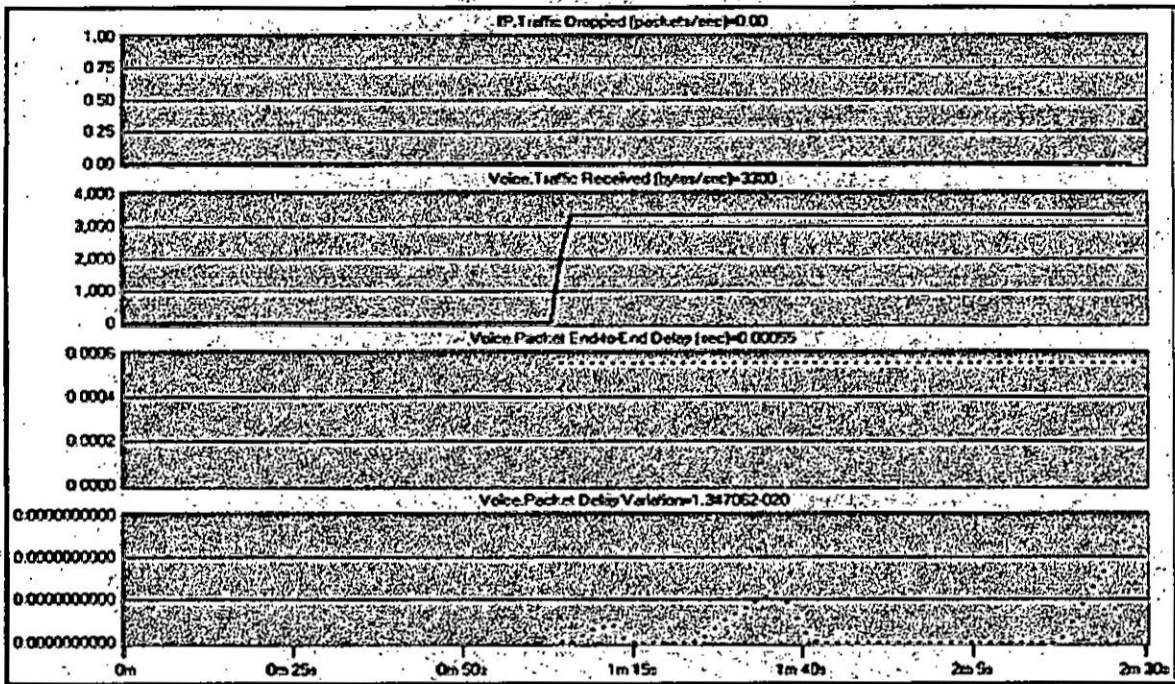


Fig. 5.5 MÉTRICAS DE QOS PARA VOZ (G 729) + TCP EN ENLACE DE 128 KBPS.

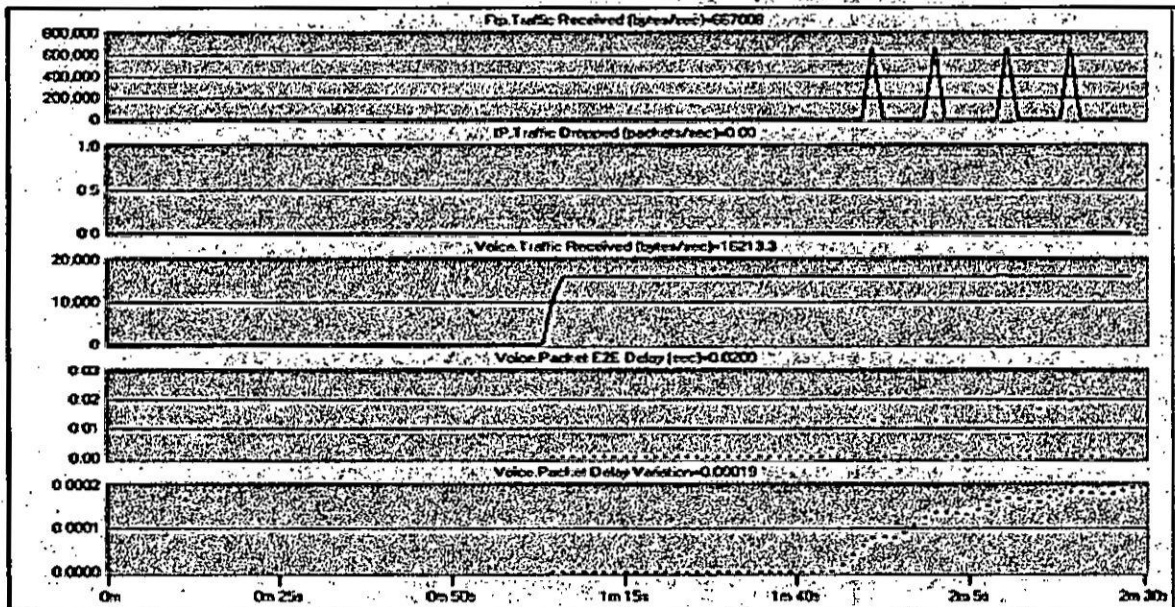
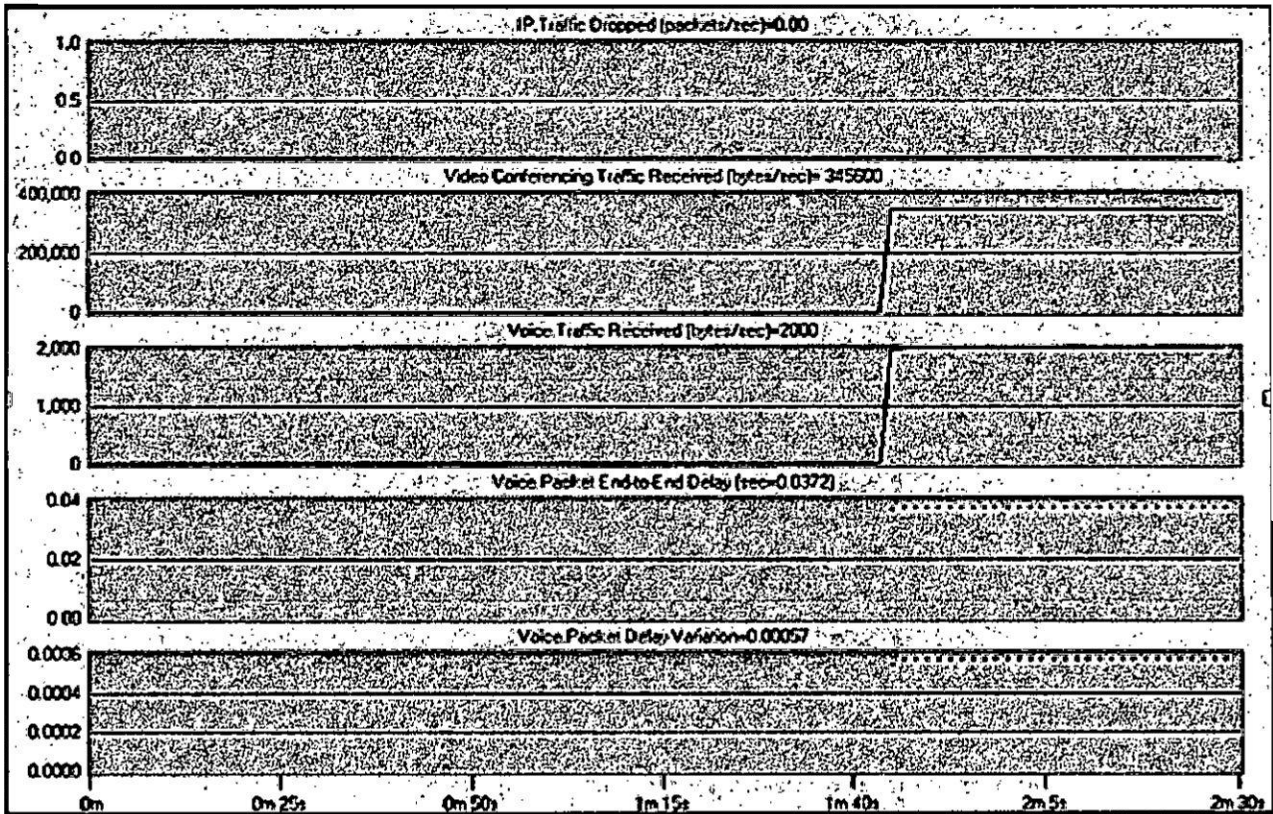


FIG 5.6 METRICAS DE QOS PARA VOZ (G 729) + UDP EN ENLACE DE 128-KBPS



5.2 RESULTADOS DE LAS EXPERIENCIAS OBTENIDAS POR SIMULACION

5.2.1 VALIDACIÓN DE LA SIMULACIÓN CON FLUJO INTERFERENTE TCP Y UDP

En esta experiencia fueron validadas las simulaciones con tráfico interferente cuando se encaminan dos flujos simultáneos (voz y TCP) y (voz y UDP), en las velocidades ya indicadas y para cada uno de los *codecs* estudiados, con equipos routers "reales". La topología de testes está indicada en la Fig. 5.13 separada de la Fig. 4.22 para fines de una mejor visualización.

La experiencia empezó transmitiendo sólo el tráfico de voz en el enlace configurado, con el objetivo de obtener el referencial para el comportamiento del flujo único de voz y con tráfico TCP y UDP.

5.2.2 FLUJO ÚNICO EN EL ENLACE

Las tablas 5.20, 5.23, 5.26, 5.29, 5.32 y 5.35 muestran los resultados de los testes efectuados con equipos "reales" para el caso de un flujo único de voz.

5.2.3 CON FLUJO INTERFERENTE TCP

Las tablas 5.21, 5.24, 5.27, 5.30, 5.33 y 5.36 muestran los resultados para el caso de un flujo de voz simultáneo con tráfico interferente **TCP**.

El terminal PC1 se comunica con el PC3 con tráfico de voz (G711) y simultáneamente el PC2 se comunica con el PC4, transportando el tráfico interferente **TCP** y **UDP** uno de cada vez. El tráfico **TCP** fue generado a través del *software FileZilla (FTP)*, transmitiendo un archivo de datos entre PC2 y PC4. Las características de esta FTP es: tamaño del archivo= 1000000 bytes, que en la velocidad de 128 Kbps representa un tiempo de transmisión de 62.5 seg.

5.2.4 CON FLUJO INTERFERENTE UDP

Las tablas 5.22, 5.25, 5.28, 5.31, 5.34 y 5.37 muestran los resultados de los testes efectuados con equipos "reales" para el caso de un flujo de voz simultáneo con tráfico interferente **UDP**. Este tráfico fue generado por el software **IPERF**.

5.2.5 EL PROCEDIMIENTO DE LAS PRUEBAS

Esta experiencia fue efectuada sobre la red mostrada en la Fig. 5.14, en la cual han sido configuradas las velocidades desde 128 **Kbps** hasta 1.5 **Mbps**, fueron contados los paquetes recibidos y transmitidos con *software Wireshark*

(*Ethereal*), registrando los valores de los parámetros relacionados con las métricas de **QoS**.

Inicialmente el PC3 y el PC1 se comunican entre si utilizando una banda que soporta la banda del códec G.711 generada por el **IPERF**. Manteniendo esta comunicación en curso, fue efectuada una transferencia de archivos entre las *workstations* PC2 y PC4 (**TCP**) con cargas equivalentes a 1.5 **MB** (62,5 segundos en el enlace de 128000 **bps**) usando los mismos valores de la simulación. En la *workstation* PC4 fue instalado el analizador de protocolo *Wireshark* para la captura de paquetes **UDP** y **TCP** cuando hay tráfico sólo de voz y mide los nuevos valores cuando se adiciona tráfico interferente **TCP** o **UDP**; para **UDP** fue generada por el **IPERF** una banda igual a la configurada para simular una situación de congestión. Han sido evaluados los efectos de estos flujos interferentes sobre cada uno de los *codecs* estudiados y los resultados están registrados en las tablas 5.20 a 5.36. Las pruebas tuvieron las características parecidas a esos simulados. El valor de **IPDT** fue conseguido a través del comando *ping*, durante la transferencia de archivo en **TCP** y la generación de tráfico en **UDP**.

FIG. 5.7 TOPOLOGÍA PARA VALIDACIÓN DE LAS MÉTRICAS OBTENIDAS POR SIMULACIÓN

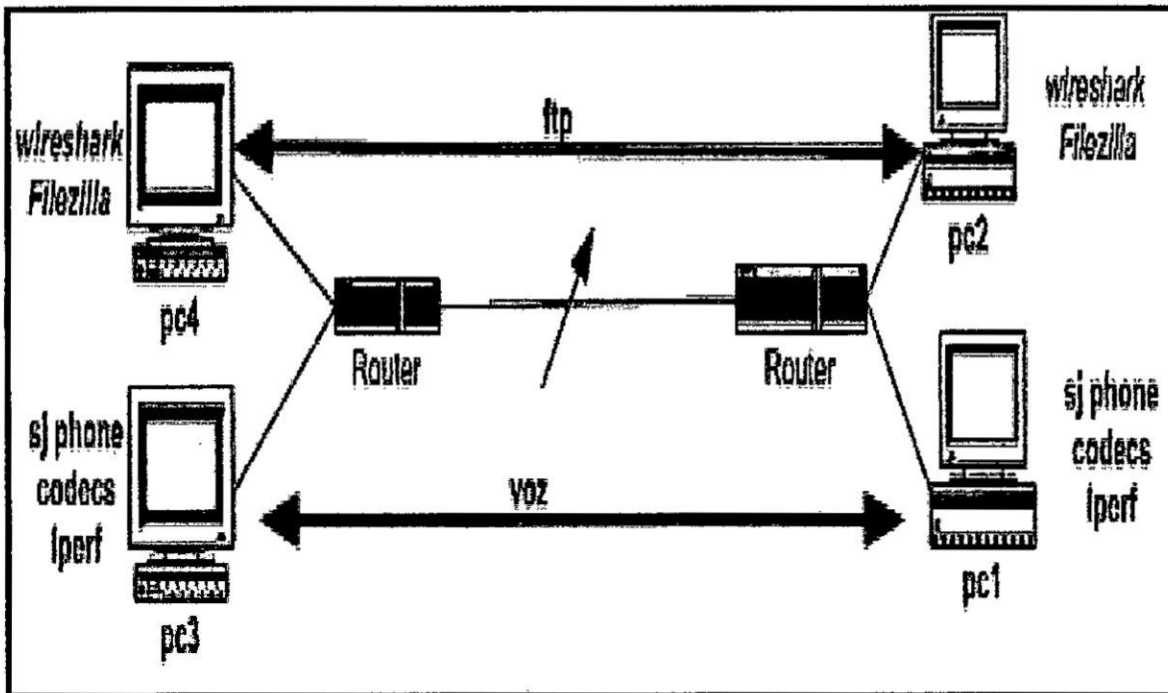


TABLA 5.1 FLUJO ÚNICO DE VOZ EN ENLACE DE 1300 KBPS

Codec	Banda bps	IPDT mseg	IPDV mseg	IPLR mseg	QoS	acción
G711	82800	0.0050	0.0014	P,00	1,9386	Mantiene
G729	45600	0.0050	0.0023	0,00	1,9206	Mantiene
GSM	31800	0.0050	0.0027	0,00	1,9126	Mantiene
G723.1A	18190	0.0050	0.0018	0,00	1,9306	Mantiene

TABLA 5.2 FLUJO ÚNICO DE VOZ EN ENLACE DE 2000 KBPS

Codec	Banda bps	IPDT mseg	IPDV mseg	IPLR mseg	QoS	acción
G711	82800	0.0025	0.0010	0,00	1,9633	Mantiene
G729	45600	0.0020	0.0075	0,00	1,9466	Mantiene
GSM	31800	0.0020	0.0015	0,00	1,9566	Mantiene
G723.1A	18190	0.0020	0.0001	0,00	1,9766	Mantiene

RESUMEN DE LA EXPERIENCIA DE VALIDACION

La Fig 5.15 muestra gráficos del IPDT de voz resultado de datos obtenidos con equipos y comunicaciones. “reales”.

Puede observarse que el comportamiento de la métrica IPDT en los cuatro gráficos, simulados y reales, son bastante semejantes.(Fig.5.1). Lo mismo sucede con el IPDV y IPLR. Por tanto para los fines de este trabajo podemos considerarlo, lo mínimo coherente con la propuesta.

5.2.6 VALIDACIÓN DEL ALGORITMO DE RESTAURACIÓN PROPUESTO

La validación del algoritmo propuesto fue efectuada en 2 actividades indicadas a seguir: 1) Análisis del comportamiento de los *codecs* en situaciones de congestión de la red (para diversos anchos de banda) y 2) el funcionamiento del algoritmo

5.2.6.1 EL COMPORTAMIENTO DE LOS CODECS EN LAS SITUACIONES DE

CONGESTIONAMIENTO DE LA RED.

El procedimiento general puede resumirse a lo siguiente: 1) fueron utilizadas Cuatro estaciones de trabajo de red LAN. Dos estaciones para la transferencia de archivo TCP y UDP simultánea con la voz y las otras dos para el canal de audio/voz. El enlace WAN foi efectuado configurando las velocidades de 128K hasta DS1 en los mismos routers, en sus puertas seriales. La conexión de las puertas seriales fue hecha con cable serial cross, Cualquiera de los lados terminales (pc4 o pc3) o (pc2, pc1) puede iniciar una comunicación telefónica con el otro lado utilizando o software VoIP SJPhone. Manteniendo esta comunicación activa, se procede con los microcomputadores restantes a hacer una transferencia de archivos de tamaño igual al de la simulación, uno de cada lado (1 MB), utilizando el software Filezilla Este procedimiento fue repetido 5 veces.

Las herramientas de *software* siguientes fueron también utilizadas: el analizador de protocolo *Wireshark*, para la captura y contaje de los paquetes y medidas de retraso y *jitter*. El *software* **IPERF** para medida de banda y generación de tráfico **UDP**. El *software* *FileZilla*, tanto como cliente como servidor de **FTP**, para la generación de tráfico interferente TCP. El *software* *ping plotter* básicamente para la medida del **RTT** y retraso fin-a-fin. Finalmente el *software* **VoIP SJPHONE** para pruebas de los codecs: **GSM**, 151 G.711 **LAW** μ , G.711 **LAW** A. Los resultados están incluidos en las tablas 5.38 y figuras 5.8 a 5.13 y el resumen en la tabla 5.39. Estas tablas y figuras son productos de las experiencias exclusivas con el

Wireshark y SJPHONE comunicándose en una red local Ethernet de 10 Mbits con cada uno de los codecs citados en el modo punto a punto.

5.2.6.2 RESULTADOS DEL COMPORTAMIENTO DE LOS CODECS DEL SJPHONE

A seguir están indicados los resultados de comunicaciones VoIP entre dos computadores, punto a punto y la captura de algunos paquetes con Wireshark.

FIG.5.8 COMUNICACIÓN CON CODEC GSM, 10 MBPS.

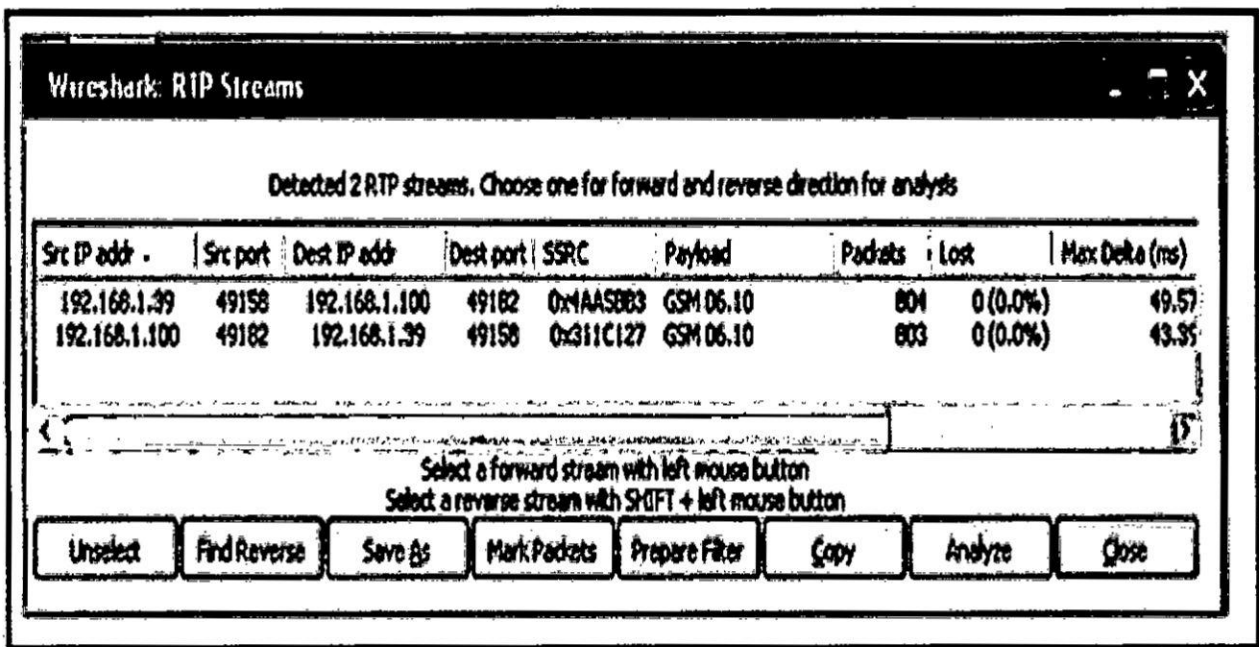


TABLA. 5.3 COMUNICACIÓN CON CODEC G711 LEY U

Dirección IP de origen	Puerto
Puerto de origen	49158
Dirección IP de destino	192.168.1.100
Puerto de destino	49182
Fuente de sincronismo	0x4AAS863
Tipo de paquete	GSM 96.10
Paquetes enviados	80

Máximo IPDV (mseg)	20
IPDV medio	19,63

Paquetes perdidos	0 0%
Máximo delta (mseg)	49.57

La tabla 5.38 arriba resume los resultados obtenidos (de máximo IPDT, máximo IPDV e IPDV medio) para una comunicación punto a punto con *codec* GSM.

FIG. 5.9 COMUNICACIÓN CON CODEC G.711 LAW A

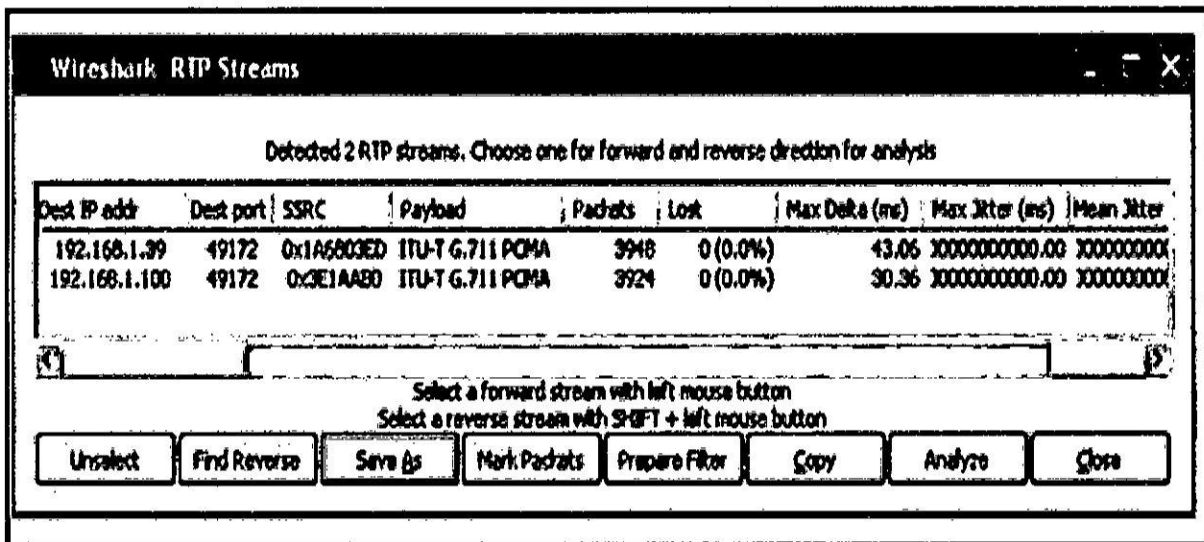


FIG 5.10 COMUNICACIÓN CON CODEC G.711 LAW μ

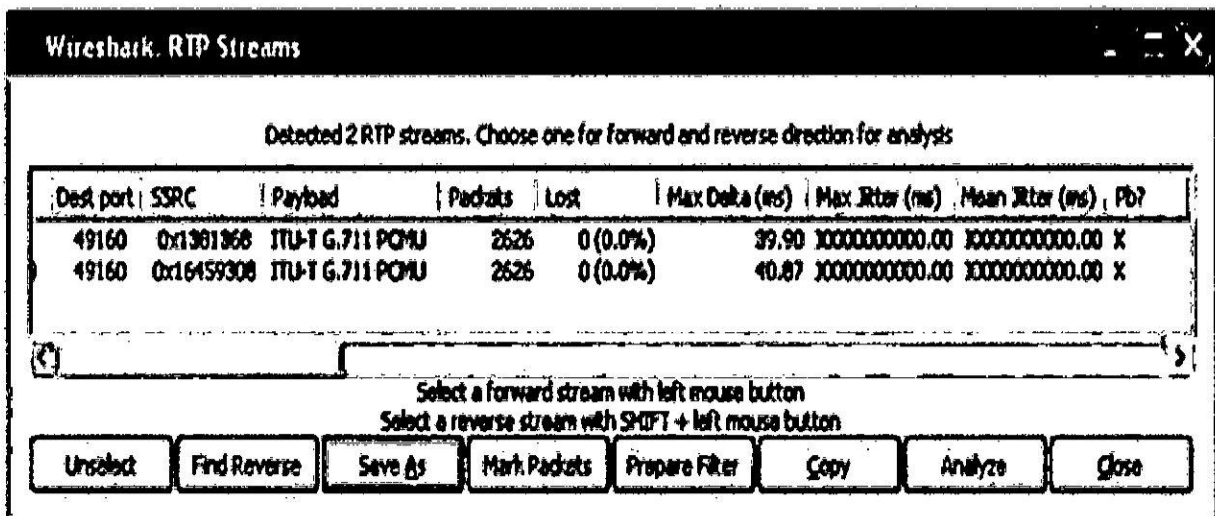


FIG 5.11 INFORMACIONES DEL PAQUETE 3505 – G711 PCMA

Time	Source	Destination	Protocol	Info
3500	216.787118	192.168.1.36	192.168.1.38	RTP Payload type=ITU-T G.711 PCMA, SSRC
3501	216.804929	192.168.1.36	192.168.1.38	RTP Payload type=ITU-T G.711 PCMA, SSRC
3502	216.814625	192.168.1.36	192.168.1.38	RTP Payload type=ITU-T G.711 PCMA, SSRC
3503	216.822106	192.168.1.36	192.168.1.38	RTP Payload type=ITU-T G.711 PCMA, SSRC
3504	216.827096	192.168.1.36	192.168.1.38	RTP Payload type=ITU-T G.711 PCMA, SSRC
3505	216.832180	192.168.1.36	192.168.1.38	RTP Payload type=ITU-T G.711 PCMA, SSRC
3506	216.846949	192.168.1.36	192.168.1.38	RTP Payload type=ITU-T G.711 PCMA, SSRC
3507	216.865114	192.168.1.36	192.168.1.38	RTP Payload type=ITU-T G.711 PCMA, SSRC
3508	216.867119	192.168.1.36	192.168.1.38	RTP Payload type=ITU-T G.711 PCMA, SSRC
3509	216.887031	192.168.1.36	192.168.1.38	RTP Payload type=ITU-T G.711 PCMA, SSRC
3510	216.894686	192.168.1.36	192.168.1.38	RTP Payload type=ITU-T G.711 PCMA, SSRC
3511	216.902186	192.168.1.36	192.168.1.38	RTP Payload type=ITU-T G.711 PCMA, SSRC
3512	216.907113	192.168.1.36	192.168.1.38	RTP Payload type=ITU-T G.711 PCMA, SSRC
3513	216.912274	192.168.1.36	192.168.1.38	RTP Payload type=ITU-T G.711 PCMA, SSRC
3514	216.927077	192.168.1.36	192.168.1.38	RTP Payload type=ITU-T G.711 PCMA, SSRC
3515	216.945060	192.168.1.36	192.168.1.38	RTP Payload type=ITU-T G.711 PCMA, SSRC
3516	216.946329	192.168.1.36	192.168.1.38	RTP Payload type=ITU-T G.711 PCMA, SSRC
3517	216.967051	192.168.1.36	192.168.1.38	RTP Payload type=ITU-T G.711 PCMA, SSRC
3518	216.974606	192.168.1.36	192.168.1.38	RTP Payload type=ITU-T G.711 PCMA, SSRC
3519	216.982311	192.168.1.36	192.168.1.38	RTP Payload type=ITU-T G.711 PCMA, SSRC
3520	216.987075	192.168.1.36	192.168.1.38	RTP Payload type=ITU-T G.711 PCMA, SSRC
3521	216.992369	192.168.1.36	192.168.1.38	RTP Payload type=ITU-T G.711 PCMA, SSRC
3522	217.007109	192.168.1.36	192.168.1.38	RTP Payload type=ITU-T G.711 PCMA, SSRC
3523	217.023700	192.168.1.36	192.168.1.38	RTP Payload type=ITU-T G.711 PCMA, SSRC

FIG 5.12 ETHEREAL COM UDP (ETH: IP:UDP:RTP)

```

Frame 3505 (214 bytes on wire (171 bytes captured) on interface eth0)
  Arrival Time: Feb 11, 2006 11:08:39.950859000
  [Time delta from previous packet: 0.005084000 seconds]
  [Time since reference or first frame: 216.832180000 seconds]
  Frame Number: 3505
  Packet Length: 214 bytes
  Capture Length: 214 bytes
  [Protocols in frame: eth:ip:udp:rtp]
  Ethernet II, Src: 192.168.1.38 (00:50:b6:00:93:dc), Dst: 192.168.1.36 (00:08:34:39:35:8c)
  Internet Protocol Version 4, Src: 192.168.1.36, Dst: 192.168.1.38
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x60 (DSCP 0x18: Class Selector 3; ECN: 0x00)
    Total Length: 200
    Identification: 0x1b3b (6971)
    Flags: 0x00
    Fragment offset: 0
    Time to live: 128
    Protocol: udp (17)
    Header checksum: 0x9aef [correct]
    Source: 192.168.1.36 (192.168.1.36)
    Destination: 192.168.1.38 (192.168.1.38)
  User Datagram Protocol, Src Port: 5000, Dst Port: 5000
    Length: 180
    Checksum: 0x1234 [correct]
    RTP Payload Type: PCMA
    SSRC: 0x12345678
    Sequence Number: 123456789
    Timestamp: 987654321
    Synchronization Source: 0x12345678
  RTP Payload: PCMA (8000 Hz, 16-bit, mono)
    Sample Rate: 8000 Hz
    Channels: 1
    Sample Format: 16-bit signed integer
    Sample Offset: 0
  
```

FIG 5.13 ÉTHEREAL COM RTP (RFC 1889)

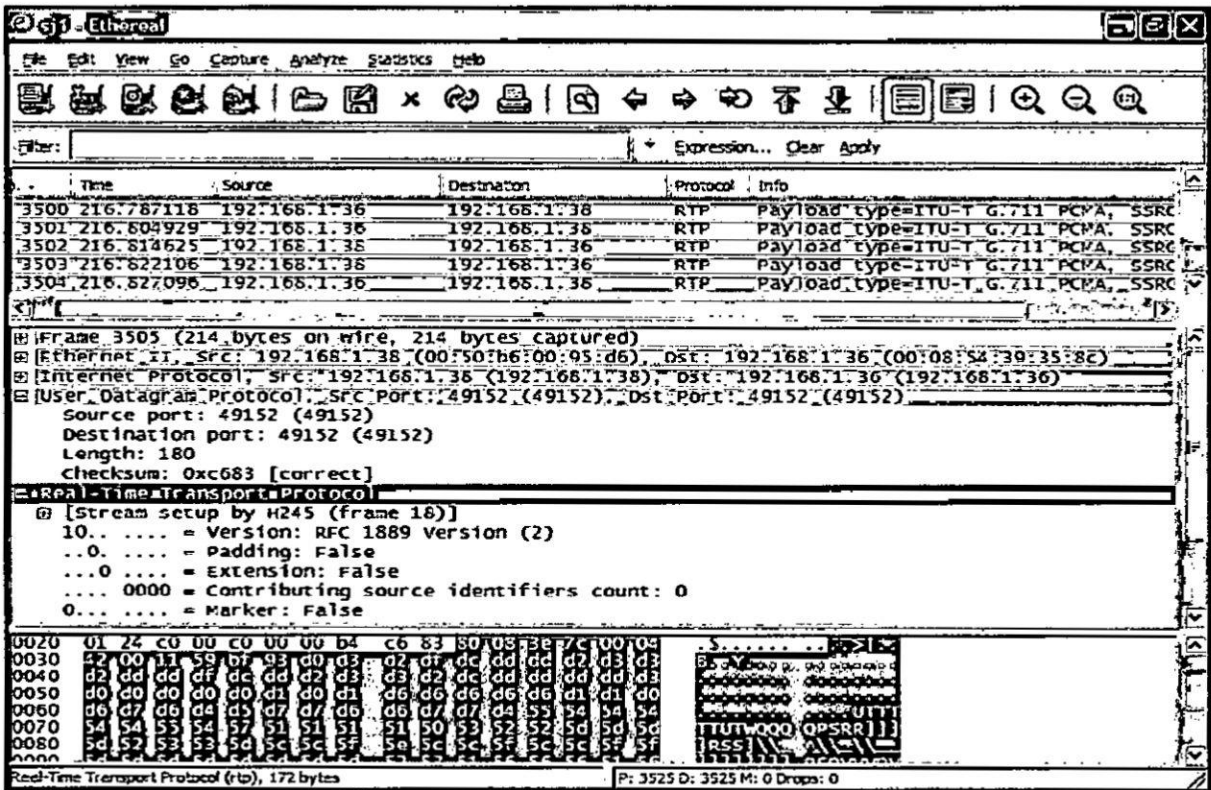
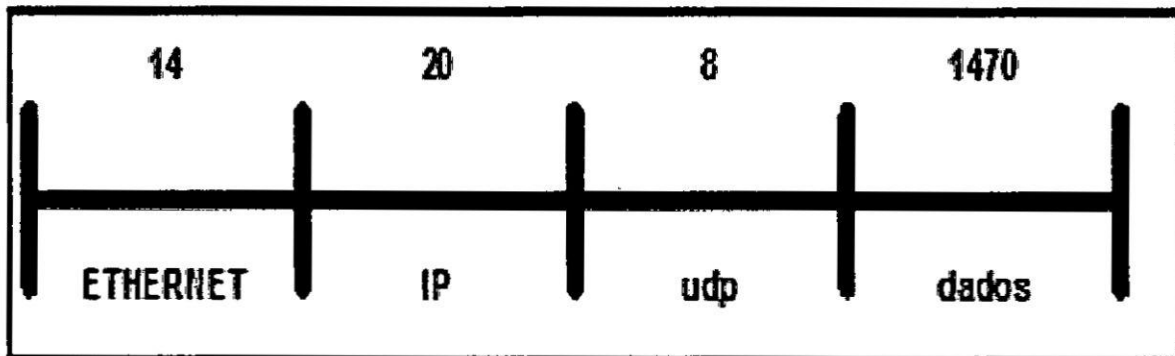


FIG. 5.14 PAQUETE IPERF UDP CON ETHEREAL



5.2.6.3 EVALUACION DEL SJPHONE CON ETHEREAL

El gráfico abajo muestra el resultado de la activacion de SJPHONE tanto en la Laptop como en la PC3, visto por el Ethereal. Observamos que ambos terminales negocian en el inicio el codec que va ser usado en la conexion, en funcion de la

banda disponible. Como la conexión es *ethernet* 10 Mbps, en este caso, intentarían conectarse con el *codec* de mas alta banda (y calidad), esto es G.711, se este *codec* estuviera implementado.

FIG 5.15 DETALLES ADICIONALES DEL PAQUETE 3505(G711 PCMA)

The screenshot shows the Wireshark interface with the following details:

- Filter:** Expression... Clear Apply
- Table:**

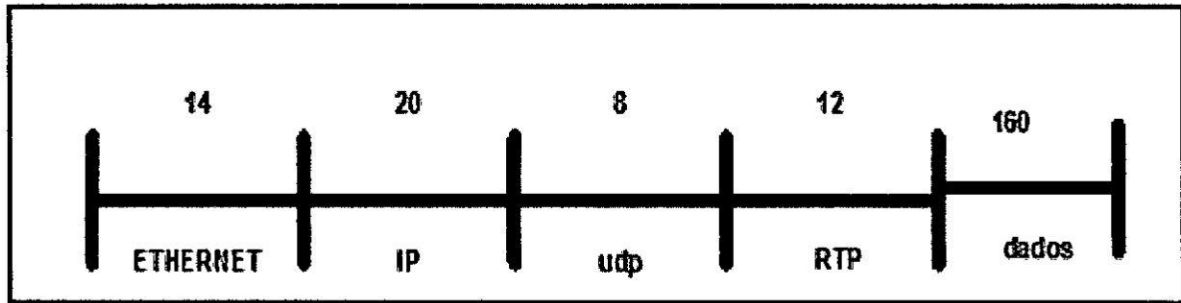
Time	Source	Destination	Protocol	Info
3500	216.787118	192.168.1.36	RTP	Payload type=ITU-T G.711 PCMA, SSRC
3501	216.804929	192.168.1.36	RTP	Payload type=ITU-T G.711 PCMA, SSRC
3502	216.814625	192.168.1.38	RTP	Payload type=ITU-T G.711 PCMA, SSRC
3503	216.822106	192.168.1.38	RTP	Payload type=ITU-T G.711 PCMA, SSRC
3504	216.827096	192.168.1.36	RTP	Payload type=ITU-T G.711 PCMA, SSRC
3505	216.832180	192.168.1.38	RTP	Payload type=ITU-T G.711 PCMA, SSRC
3506	216.846949	192.168.1.36	RTP	Payload type=ITU-T G.711 PCMA, SSRC
3507	216.865114	192.168.1.38	RTP	Payload type=ITU-T G.711 PCMA, SSRC
3508	216.867119	192.168.1.36	RTP	Payload type=ITU-T G.711 PCMA, SSRC
3509	216.867031	192.168.1.36	RTP	Payload type=ITU-T G.711 PCMA, SSRC
3510	216.894666	192.168.1.38	RTP	Payload type=ITU-T G.711 PCMA, SSRC
3511	216.902186	192.168.1.38	RTP	Payload type=ITU-T G.711 PCMA, SSRC
3512	216.907113	192.168.1.36	RTP	Payload type=ITU-T G.711 PCMA, SSRC
3513	216.912274	192.168.1.38	RTP	Payload type=ITU-T G.711 PCMA, SSRC
3514	216.927077	192.168.1.36	RTP	Payload type=ITU-T G.711 PCMA, SSRC
3515	216.945060	192.168.1.38	RTP	Payload type=ITU-T G.711 PCMA, SSRC
3516	216.946329	192.168.1.36	RTP	Payload type=ITU-T G.711 PCMA, SSRC
3517	216.967051	192.168.1.36	RTP	Payload type=ITU-T G.711 PCMA, SSRC
3518	216.974806	192.168.1.38	RTP	Payload type=ITU-T G.711 PCMA, SSRC
3519	216.982311	192.168.1.38	RTP	Payload type=ITU-T G.711 PCMA, SSRC
3520	216.987075	192.168.1.36	RTP	Payload type=ITU-T G.711 PCMA, SSRC
3521	216.992369	192.168.1.38	RTP	Payload type=ITU-T G.711 PCMA, SSRC
3522	217.007109	192.168.1.36	RTP	Payload type=ITU-T G.711 PCMA, SSRC
3523	217.023700	192.168.1.38	RTP	Payload type=ITU-T G.711 PCMA, SSRC
3524	217.025168	192.168.1.36	RTP	Payload type=ITU-T G.711 PCMA, SSRC
3525	217.047095	192.168.1.36	RTP	Payload type=ITU-T G.711 PCMA, SSRC
- Packet Bytes:**

```

0000 00 08 54 39 35 8c 00 50 b6 00 95 d6 08 00 45 60 ..T95..P .....E
0010 00 c8 1b 3b 00 00 80 11 9a ef c0 a8 01 26 c0 a8 ...:.....&..
0020 01 24 c0 00 c0 00 00 b4 c6 e2 80 00 3a 7c 00 04

```
- Status:** File: C:\ethcap\sj1 790 KB 00:03:37 | P: 3525 D: 3525 M: 0 Drops: 0

FIG 5.16 PAQUETE SJPHONE



La tabla 5.39 resume las características IPDT e IPDV de los *codecs* disponibles en el SJPHONE.

5.2.7 EL COMPORTAMIENTO DEL ALGORITMO DE QOS

Esta prueba es también llamada de "Validación Manual", visto que la experiencia no es efectuada dinámicamente con los flujos de RTP/RTCP en curso, si no es evaluada en función de valores digitados por el operador cuando se ejecuta el algoritmo propuesto en un microcomputador que va solicitando los valores de los parámetros de entrada que en la realidad serian automáticos.

5.2.7.1 VALIDACIÓN DEL ALGORITMO

La validación, fue efectuada ejecutando el *software* en un microcomputador, y digitando los valores de los parámetros solicitados.

5.2.7.2 EJEMPLO DE PRUEBA DEL ALGORITMO

En este ejemplo ejecutase el programa del algoritmo que se inicia "solicitando" ingresar **MSB**. El operador deberá digitar el valor solicitado del **MSB** correspondiente en hexadecimal o decimal y así sucesivamente hasta solicitar

ingresar *interarrival*. El ítem 2 parámetros de salida (en **negrito**) representa el resultado de las operaciones del algoritmo.

FIG. 6.2 IPDT PARA LOS 4 CODECS DE VOZ + FTP EN UN ENLACE DE HASTA 2 MBPS

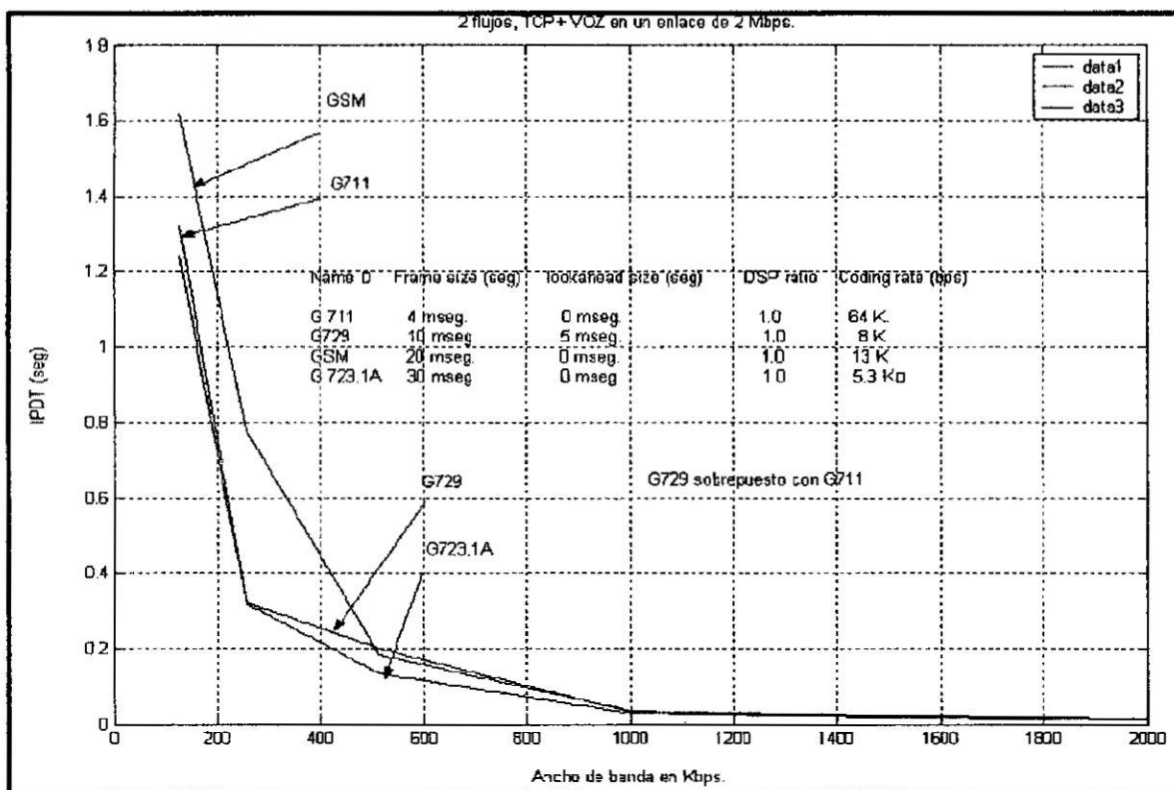


TABLA 6.1 CARACTERÍSTICAS DE LOS CODECS UTILIZADOS

Codec	Frame size (seg)	Lookahead Size (seg)	DSP ratio	Coding rate (bps)
G711	0,004	0,00	1.0	64K
G729	0,010	0.005	1.0	8K
GSM	0,020	0,00.	1.0	13 K
G 723.1A	0.030	0.00	1,0	5.3 K

La Fig. 6.3 muestra el comportamiento del IPDT cuando opera simultáneamente un flujo de vídeo y uno de voz en el enlace de hasta 1.5 Mbps; note que el valor para 256 Kbps es de aprox. 8 seg., superior al anterior de 400 miliseg.

FIG. 6.3 IPDT PARA VOZ + VIDEO EN FUNCIÓN DE LA BANDA

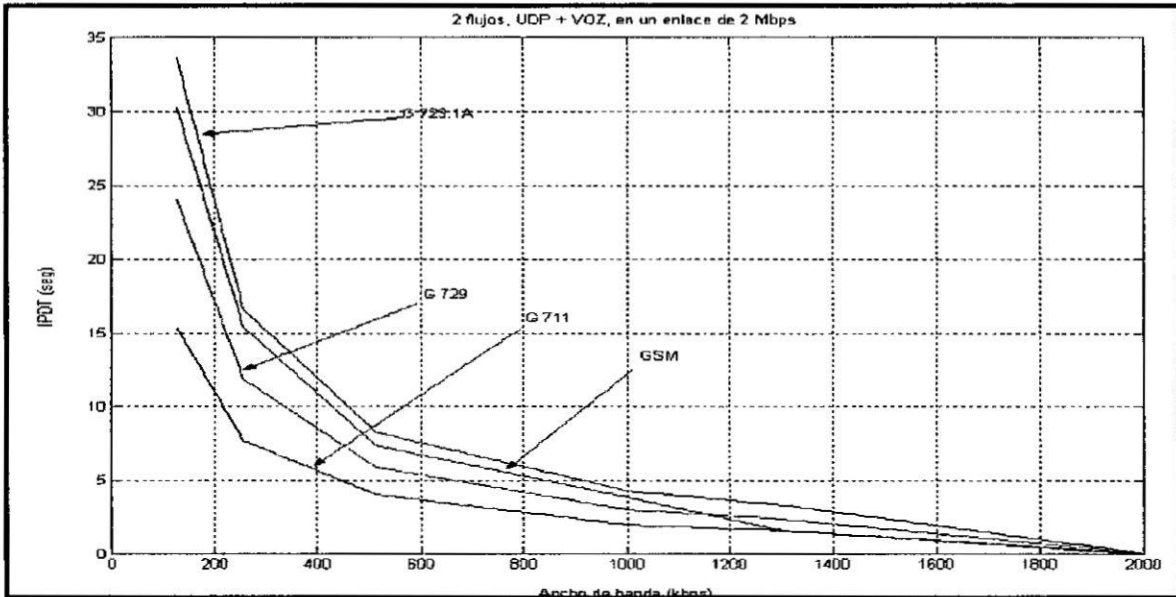
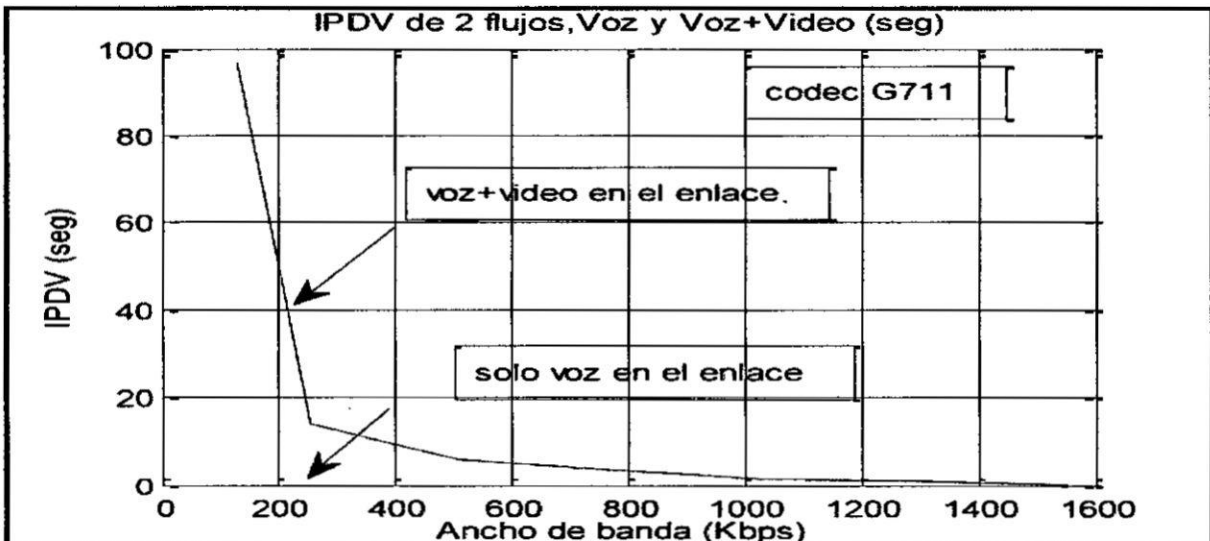


FIG. 6.5 IPDV DE 2 FLUJOS (VOZ + UDP) EN FUNCIÓN DEL ENLACE

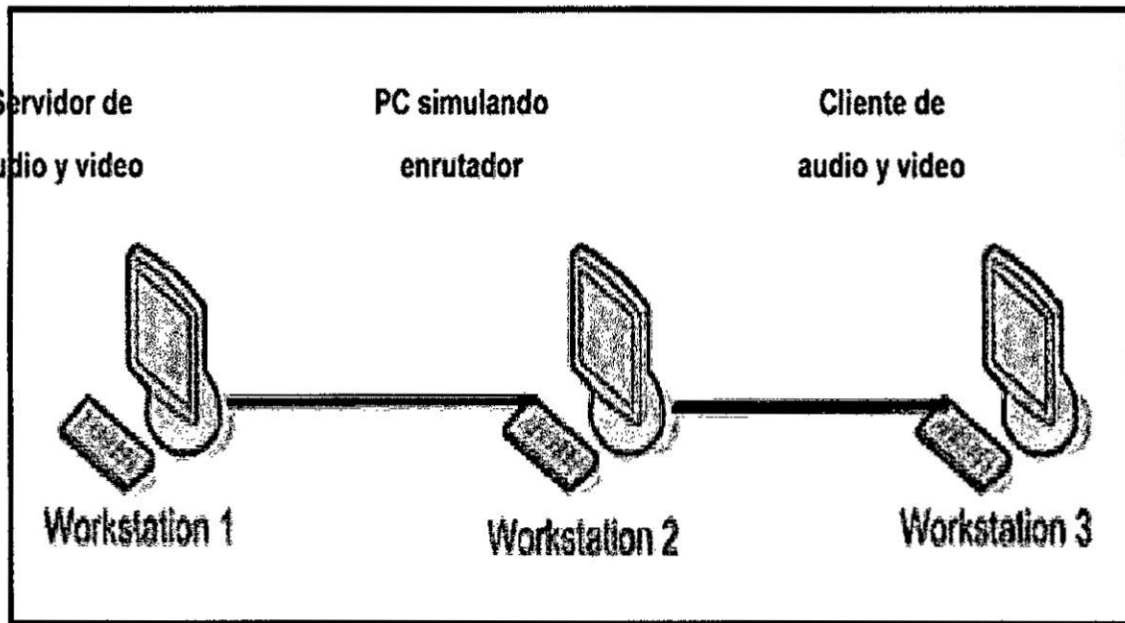


Para esto es necesario analizar en conjunto con el costo adicional o programar los usos de los canales, o también como veremos en el ejemplo anexo, mudar el tipo de servicio “best effort” por otro servicio con protocolos mas elaborados.y eficientes. Los datos hasta aqui presentados -confirman nuestra hipótesis en lo relacionado a IPDT:

6.2. CONTRASTACION DE LOS RESULTADOS CON RELACIÓN A TRABAJOS ANTERIORES

Varios estudios anteriores han abordado este tema con enfoques diferentes cada uno de ellos, tanto a nivel interno de la Escuela Politécnica, como externo a ella. El mas parecido con el nuestro es el indicado a continuación pero que no llega a ninguna conclusión concreta y los valores experimentados están muy fuera de la realidad como por ejemplo experiencias con valores de atraso de la orden de 2000 ms, si sabemos 165 que la norma limita em 150 a 200 ms. En el ámbito interno se utiliza el modelo de caracterización de los parámetros de **QoS** y la fórmula de cálculo de la propia **QoS**. Estos estudios usaran un *testbed* cuya topología se muestra en la Fig. A .

FIG. A TESTBED DE TRABAJOS ANTERIORES



En esta topología no son empleados *routers*, sino un **PC** emulando el *router* con el software emulador *NistNet*, un *software* que permite a un ordenador con sistema operativo *Linux* sea configurado como un *router* y emular muchas de las condiciones de congestión de la red, mediante el control de varios parámetros, como banda disponible, *delay*, *jitter* y pérdida de paquetes. Externamente, podrá ser citado el software *Mview* de la Universidad de Columbia, Nueva York, una herramienta que le permite ver el funcionamiento de la red **Mbone** (*Multicast Backbone*) a través del uso de diversas herramientas de red independientes que recogen informaciones topológicas y estadísticas de rendimiento.

Los detalles se pueden encontrar en el documento "**RTP/RTCP** extensión de **Mview-Mbone** herramienta de seguimiento" o correo electrónico mk@comet.columbia.edu, dirigido a Michael E. Kounaves. *Mview* se instala bajo *Solaris 2.5.1* y requiere la versión 4.1 o superior de **Tcl/Tk**.

VII. CONCLUSIONES

Actualmente la prueba de la calidad de la voz es resultado de una percepción subjetiva y de un proceso de juzgamiento durante el cual, un “escucha”, compara el evento perceptual, con una referencia interna que es juzgada a ser de buena calidad. Resumidamente este proceso incluye:

El promedio de la puntuación dadas por los “escuchas”, el cual es conocido como **MOS** (*Mean Opinión Score*), o como sugerido por la recomendación P.800.1 del ITU-T, **MOS-LQS** (*Listening Quality Subjective*). Las pruebas subjetivas, sin embargo son costosas y consumen mucho tiempo, tornándolas inadecuadas para aplicaciones *on- the-fly* (sobre la marcha) real time.

En las mediciones objetivas de calidad de la voz, el panel de “escuchas” es remplazado por un algoritmo computacional facilitando las mediciones automáticas de aplicaciones *real-time*. Para comunicaciones **VoIP**, los métodos objetivos son clasificados en 2 tipos: 1) basados en señales y 2) basados en parámetros. Los métodos basados en señales usan características perceptuales extraídas de las señales del habla para estimar la calidad. Los métodos basados en parámetros usan los parámetros de conexión **VoIP**, tales como codecs, pérdida de paquetes, *jitter*, *delay*, etc., para computar el deterioro de los factores que son usados para determinar la calidad del habla. Este segundo método, es justamente el caso de este trabajo de investigación (objetivo paramétrico) con los parámetros de calidad obtenidos vía el protocolo **RTP/RTCP** que es presentado en una metodología simple para monitoreo y restauración de la calidad de servicio sobre aplicaciones de voz. La validación visa 167 atender a esta etapa del trabajo realizando el teste

del algoritmo de forma separada del protocolo **RTP**, pero verificando todas y cada una de las actividades que serían efectuadas con el protocolo mencionado.

Se visualiza una mejoría en el modelo de servicio tradicional de las redes **IP**, conocido como "best effort", que sería conseguido con su implementación. Algunos resultados complementares fueron obtenidos en el transcurso de este trabajo y aunque no hacen parte del objetivo fueron brevemente comentados. Las tablas 5.19 a 5.36 muestran los valores de **IPDV** e **IPDT** y consecuentemente **QoS**, obtenidos con equipos "reales" (*routers Cisco*) que se muestran un poco más altos que los valores simulados para la misma situación (único en el enlace, por ejemplo). Similarmente se observa que para la misma banda el tráfico **UDP** presenta valores mayores de **IPDV** que el tráfico interferente **TCP**.

Sin embargo, esto es entendible si consideramos la naturaleza y funcionamiento de los protocolos de transporte. El **UDP** es más crítico y rígido que el **TCP**. El tema mantiene su relevancia mismo cuando comparado con otras propuestas como las resumidas a seguir: Analizando el enfoque de esos trabajos, efectuados por otros investigadores y para uso con aplicaciones en tiempo real (en la *Internet*), *podremos observar lo siguiente*. La primera fue implementada en el **IVS** ("**Institut National de Recherche en Informatique et en Automatique**") **INRIA** (*Videoconference System*), ejecutando sobre el protocolo **RTP/RTCP** y basado en el valor medio de la tasa de pérdida de paquetes de los receptores. Este algoritmo presenta dos inconvenientes, 1) evalúa la red solamente por la tasa de pérdida de paquetes y 2) aumenta su velocidad mediante un incremento presenta sin considerar la banda disponible. Cuando este aumento es inadecuado, la 168 tasa de envío puede oscilar ampliamente ocasionando una grande pérdida de paquetes

a otras aplicaciones que comparten el mismo enlace. Otro algoritmo, intenta ajustar la velocidad de acuerdo con la variación del **RTT** medido antes del aumento de la pérdida de paquetes. Aumenta o reduce la velocidad en valores pequeños dependiendo si el **RTT** está abajo o por encima de un límite previamente definido. Aumento de la velocidad es lo mismo que aumento de ancho de banda y cae en uno de los casos perjudiciales anteriores pues significa un aumento de banda que puede no estar disponible. Un tercer algoritmo, conocido como **RAP** (*Rate Adaptation Protocol*), usa un aumento aditivo y una disminución multiplicativa. Durante la congestión, inmediatamente baja la velocidad para la mitad y la va aumentando aditivamente uno por uno hasta comenzar a perder paquetes donde para. Procede como en el caso de TCP: Su limitación en aplicaciones de tiempo real se encuentra en el requerimiento de la confirmación de cada paquete transmitido. Debido a que su velocidad de transmisión depende de la pérdida de paquetes, esta puede oscilar de forma parecida al **TCP** y una grande oscilación puede ocasionar un grande *jitter* perjudicando así la calidad de la voz y vídeo. Es importante mencionar que además del algoritmo, las actividades de simulación y análisis de tráfico interferente **TCP** y **UDP**; este trabajo, visa mostrar y ayudar a aclarar las dudas sobre las penalidades ocasionadas por situaciones de congestión con tráfico igual o diferente del tráfico principal de voz (**UDP**). Esta forma facilita la visualización del mecanismo propuesto, simple y de bajo costo. Sin duda alguna, si fuese posible disponer de más recursos reales y facilidad de acceso a las redes comerciales o a laboratorios especializados, completaríamos de otra forma las conclusiones aquí presentadas.

Sin la pretensión de haber completado el asunto, ni considerarlo con el rigor científico necesario correspondiente, se puede observar en los resultados mostrados en las tablas 5.1 a 5.18 en lo que dice respecto a las métricas de QoS, **IPDV** e **IPDT** y en las figuras 5.1 a 5.11, lo siguiente:

1. Un flujo de voz, codificado por cualquier uno de los *codecs* G.711, G.729, **GSM** o G.723.1 A, cuando es transmitido solo en un enlace de 128 **Kbps**, no sufre ninguna degradación significativa (como era de esperar).
2. Cuando un flujo **TCP** entra para competir con el flujo de voz en el mismo enlace se nota que si la banda "residual" es insuficiente para el tráfico **TCP**, fuerza la conmutación del flujo de voz para el *codec* siguiente en orden decreciente y así sucesivamente hasta el *codec* de menor banda, esperando que en al final la situación de congestión finalice, para iniciar el retorno hasta el *códec default*. Los valores obtenidos indican que la banda residual ($128000 - 82800 = 45200$ **bps**) es insuficiente para el tráfico **FTP** analizado, pero se transmite al costo del aumento de las métricas de **QoS**, *delay* y *jitter*..
3. Una situación similar a la anterior fuerza la conmutación cuando un segundo flujo, en este caso **UDP** de vídeo, compite con el flujo de voz, pero de una forma más rígida que en el caso del flujo **TCP** del **FTP**.
4. Considerando que solamente era previsto la propuesta del algoritmo y otros datos surgieron durante la preparación de este documento decidimos incluir los casos más significativos. Falto el caso de la determinación del punto en el cual el tráfico adicional deja de interferir en el flujo de voz, es decir, cual es el ancho de banda mínimo residual que permite al tráfico interferente **TCP** y **UDP** viajar

1300K-Solo voz						
4.13	0,0016	0,9893	0,0000	0,0000	0,9893	Mantiene
	0,0004	0,9973	0,0000	0,0000	0,9973	Mantiene
	0,0004	0,9973	0,0000	0,0000	0,9973	Mantiene
	0,0005	0,9967	0,0000	0,0000	0,9967	Mantiene
1300K- voz + TCP						
4.14	0,0256	0,8293	0,0000	0,0000	0,8293	Mantiene
	0,0223	0,8513	0,0000	0,0000	0,8513	Mantiene
	0,0223	0,8513	0,0000	0,0000	0,8513	Mantiene
	0,0223	0,8513	0,0000	0,0000	0,8513	Mantiene
1000K- Voz+UDP						
4.15	3,1845	-20,2300	1,0690	-20,3800	-40,6100	Conmuta
	2,4232	-15,1547	0,6038	-11,0760	-26,2307	Conmuta
	2,4232	-15,1547	0,6038	-11,0760	-26,2307	Conmuta
	0,0220	0,8533	0,0020	0,9600	1,8133	Mantiene
DS1- Solo voz						
4.16	0,0012	0,9920	0,0000	0,0000	0,9920	Mantiene
	0,0003	0,9980	0,0000	0,0000	0,9980	Mantiene
	0,0003	0,9980	0,0000	0,0000	0,9980	Mantiene
	0,0004	0,9973	0,0000	0,0000	0,9973	Mantiene
DS1-Voz + TCP						
4.17	0,0143	0,9047	0,0001	0,9974	1,9021	Mantiene
	0,0135	0,9100	0,0001	0,9980	1,9080	Mantiene
	0,0135	0,9100	0,0001	0,9980	1,9080	Mantiene
	0,0136	0,9093	0,0001	0,9980	1,9073	Mantiene
DS1- Voz + UDP:						
4.18	0,0192	0,8720	0,0030	0,9400	1,8120	Mantiene
	0,0020	0,9867	0,0003	0,9940	1,9807	Mantiene
	0,0203	0,8647	0,0003	0,9940	1,8587	Mantiene
	0,0189	0,8740	0,0003	0,9940	1,8680	Mantiene

Como puede observarse en la tabla 6.1 la experiencia efectuada con equipos reales, muestra un resultado más estable El simulado conmuta para protección 7 veces (33% de continuidad) y el real, ninguna vez (100% de continuidad). Lo

máximo que llega es a la situación de "cargada". Fueron adoptados como válidos los resultados de la prueba real, confirmando estos valores. Es probable que tenga una influencia física de la red de pruebas, pues los valores simulados incluyen IPDV =0.y IPLR=0.

TABLA 7.2

RESULTADOS DE LA EXPERIENCIA "REAL"

CODEC	IPDT	QOS1	IPDV	QOS2	TOTAL	ACCION
VELOCIDAD/FLUJO/TABLA						
128KBPS/VOZ/ 5.20						
G711	0.0155		0.0011		1.9746	mantiene
G729	0.0110		0.0012		1.9026	mantiene
GSM	0.0095		0.0020		1.8966	mantiene
G723.1A	0.0080		0.0023		1.9006	mantiene
128K/ Voz+TCP/ 5.21						
G711	0.0690		0.0704		0.1320	mantiene
G729	0.0830		0.0044		1.3586	mantiene
GSM	0.0660		0.0474		0.6120	mantiene
G723.1A	0.0750		0.0295		0.9100	mantiene
128K/ Voz+UDP/ 5.22						
G711	0.0072		0.0038		1,876	mantiene
G729	0.0072		0.0038		1,976	mantiene
GSM	0.0072		0.0038		1,876	mantiene
G723.1A	0.0072		0.0038		1,976	mantiene
256/Voz/ 5.23						
G711	0.0060		0.0029		1,9020	mantiene
G729	0.0035		0.0010		1,9566	mantiene
GSM	0.0030		0.0010		1,9600	mantiene
G723.1A	0.0030		0.0010		1,9600	mantiene
256/Voz+TCP/ 5.24						
G711	0.0041		0.0027		1,9186	mantiene
G729	0.0037		0.0041		1,8933	mantiene
GSM	0.0035		0.0051		1,8746	mantiene
G723.1A	0.0035		0.0091		1,7946	mantiene
256/Voz+UDP/ 5.25						
G711	0.037		0.0670		1,6193	mantiene
G729	0.037		0.0067		1,6193	mantiene
GSM	0.037		0.0067		1,6193	mantiene
G723.1A	0.037		0.0067		1,6193	mantiene
512/Voz/ 5.26						

G711	0.0015		0.4910		* -7,8300	mantiene
G729	0.0020		0.4020		*- 6.053	mantiene
GSM	0.0010		0.0010		1,9733	mantiene
G723.1A	0.0015		0.0060		1,8700	mantiene
512/Voz+TCP/ 5.27						
G711	0.0140		0.0015		1,8766	mantiene
G729	0.0165		0.0031		1,8280	mantiene
GSM	0.0150		0.0043		1,8140	mantiene
G723.1A	0.0125		0.0053		1,8106	mantiene
512/Voz+UDP/ 5.28						
G711	0.0190		0.0066		1,7413	Mantiene
G729	0.0190		0.0066		1,7413	mantiene
GSM	0.0190		0.0066		1,7413	mantiene
G723.1A	0.0190		0.0066		1,7413	mantiene
1000 k/Voz/ 5.29						
G711	0.0005		0.0010		1,9766	Mantiene
G729	0.0005		0.1980		1,9766	Mantiene
GSM	0.0005		0.0005		1,9766	Mantiene
G723.1A	0.0005		0.0012		1,9766	Mantiene
1000 k/Voz+TCP/ 5.30						
G711	0.0045		0.0060		1,8500	Mantiene
G729	0.006		0.0060		1,8366	Mantiene
GSM	0.0060		0.0060		1,8400	Mantiene
G723.1A	0.0030		0.0060		1,8600	Mantiene
1000 k/Voz+UDP/ 5.31						
G711	0.0105		0.0048		1,8340	Mantiene
G729	0.0105		0.0048		1,8340	Mantiene
GSM	0.0105		0.0048		1,8340	Mantiene
G723.1A	0.0105		0.0048		1,8340	Mantiene
1300 k/Voz/ 5.32						
G711	0.0050		0.0014		1,9386	Mantiene
G729	0.0050		0.0023		1,9206	Mantiene
GSM	0.0050		0.0027		1,9126	Mantiene
G723.1A	0.0050		0.0018		1,9306	Mantiene
1300 k/Voz+TCP/ 5.33						
G711	0.0040		0.0052		1,8696	Mantiene
G729	0.0035		0.0046		1,8846	Mantiene
GSM	0.0015		0.0020		1,9500	Mantiene
G723.1A	0.0015		0.0016		1,9580	Mantiene
1300 k/Voz+UDP/ 5.34						
G711	0.0050		0.0003		1,9606	Mantiene

G729	0.0050		0.0012		1,9426	Mantiene
GSM	0.0050		0.0049		1,8686	Mantiene
G723.1 ^a	0.0050		0.0012		1,9426	Mantiene
1544k/Voz/ 5.35						
G711	0.0025		0.0010		1,9633	Mantiene
G729	0.0020		0.0075		1,9466	Mantiene
GSM	0.0020		0.0015		1,9566	Mantiene
G723.1A	0.0020		0.0001		1,9766	Mantiene
1544k/Voz+TCP / 5.36						
G711	0.0025		0.0010		1,9633	Mantiene
G729	0.0020		0.7590		1,9626	Mantiene
GSM	0.0020		0.0015		1,9566	Mantiene
G723.1A	0.0020		0.0010		1,9666	Mantiene
1544k/Voz+UDP / 5.37						
G711	0.0055		0.0085		1,7933	Mantiene
G729	0.0055		0.0085		1,7933	Mantiene
GSM	0.0055		0.0085		1,7933	Mantiene
G723.1A	0.0055		0.0085		1,7933	Mantiene

Notas:

1.- En la segunda línea de esta tabla: Velocidad/Flujo/Tabla indican la velocidad del enlace, el flujo de voz (voz, voz + TCP, Voz + UDP), y Tabla, indica el número de la tabla del capítulo V que origino estos datos.

Probablemente un transiente podrá ser investigado si fuese necesario.

VIII. RECOMENDACIONES Y TRABAJOS FUTUROS

Las redes de comutacion de paquetes actualmente ofrecen servicio *best effort* en una base de primer paquete que llega primero a ser atendido (**FIFO**) y de esta forma el atraso puede ser altamente variable. Esto puede no afectar al tráfico de datos, debido a que estos son transmitidos en ráfagas e insensibles al atraso, y cualquier pérdida de paquetes puede ser fácilmente retransmitida sin cualquier restricción de límite de tiempo. Sin embargo estas características no son aceptables por el tráfico "real time," siendo la voz y video dos ejemplos de este tipo de tráfico. Otro aspecto es que la evaluación de la calidad de la voz es hecha principalmente de forma subjetiva en la cual se pide a un grupo de personas para escuchar algunos trechos de conversaciones y después expresar su opinión sobre la calidad. El otro método emergente, es el conocido como método "objetivo" que en su modalidad paramétrica analiza las métricas antes mencionadas, (atraso, variación de atraso, perdida de paquetes, eco, etc), siendo este el cuadro actual de las telecomunicaciones, y **VoIP** considerado el sustituto de la telefonía tradicional, es necesario una actualización en los nuevos conceptos y técnicas de telecomunicaciones. Anteriormente, la red telefónica era el grande exponente de esta área y hoy en día son las redes de paquetes **IP**, redes *frame relay*, redes ópticas, etc. y principalmente la *Internet*.

De aquí nuestra recomendación a quienes participan en esta área sea industrial o académica, a actualizar sus conceptos y particularmente a los de la área

académica, a participar más activamente con trabajos de investigación, publicación de trabajos de graduación/Post graduación o tesis de forma general.

Consideramos los resultados obtenidos altamente promisorios y recomendamos continuar en esta línea de investigación.

La continuidad de trabajos en esta área es facilitada por ser justamente un área con mucho dinamismo visto la gran cantidad de tesis presentadas y las mudanzas tecnológicas ocurriendo cada vez en menor espacio de tiempo. Específicamente, por ejemplo, este trabajo enfoca un tema de **QoS** usando el protocolo **PPP** como protocolo de enlace WAN. Un proyecto alternativo de pruebas podría incluir *frame relay* o *Ethernet*, como protocolo WAN etc, o proponer una topología diferente .El campo es muy variado.

IX. REFERENCIAS BIBLIOGRAFICAS

1. BARRA, C. Caracterização experimental e por simulação e modelagem da Qualidade de Serviço obtida na transmissão de áudio e vídeo em tempo real. Tese de Doutorado, EPUSP (2005).
2. TAKAHASHI, A. et al. Objective assessment Methodology for Estimating Conversational Quality in VoIP, IEEE transactions on audio, speech and language processing. Novembro de 2006.
3. CLARK, D.; SHENKER, S. AND ZHANG, L. "Supporting real-time applications in an integrated services packet network> Architecture and mechanism". ACM SIGCOMM'92. Agosto de 1992.
4. MARCHESE, M. QoS over heterogeneous networks. John Wiley & sons Ltd. 2007.
5. ITU-T. Y 1541, Recommendation Network Performance Objectives for IP-based Services, Febrero de 2003.
6. JAMES, J.H. et al. A Voice Transmission Performance Progress Report. IEEE Communication Magazine. Julio de 2004.
7. Tommi K. Protocol Overview: RTP and RTCP. Nokia. Telecommunications, paper. Junio 2003. tommi.koistineu@ntc.nokia.com.
8. OPNET IT GURU, versão acadêmica 9.1 A. www.opnet.com.
9. lperf. LANR/DAST. <http://dast.nlanr.net/>.
10. SJPhone. www.sjlabs.com.

- 11.** Ethereal - Network Protocol Analyzer. Disponível em: <http://www.ethereal.com> acessado em 22/03/2003.
- 12.** ETSI TIPHON TS101 329-5 Annex E, Method for determining an equipment impairment factor using passive monitoring. Noviembre de 2000.
- 13.** ITU-T Recommendation Y.2011, General principles and general reference model for next generation networks. Octubre de 2004.
- 14.** FERNANDEZ, D.; PASTOR, E.; BELLIDO, L. Experiences Integrating Cooperative Multimedia Applications into WWW. 2 CYTED-RITOS International Workshop on Groupware (CRIWG86), Puerto Varas,
- 15.** SNIR, Y.; RAMBERG, Y.; STRASSNER, J. et al. RFC 3644. Policy Quality of Service (QoS) Information Model. Noviembre de 2003.
- 16.** RFC 3583. Requirements of a Quality of Service (QoS) Solution for Mobile IP. Septiembre de 2003.
- 17.** MOORE, B.; DURHAM, D.; STRASSNER, J. et al. RFC 3670. Information Model for Describing Network Device QoS Datapath Mechanisms. Janeiro de 2004.
- 18.** JIANG, S. Reviewed articles: Granular differentiated queueing services for QoS: structure and cost model. ACM SIGCOMM Computer Communication Review, volume 35 Issue 2. Abril de 2005.
- 19.** CHANG, W. S.; SIMON, R. Performance Analysis for Multi-Service Networks with Congestion-Based Pricing for QoS Traffic. Proceedings of the 38th annual Symposium on Simulation. Abril de 2005.

- 20.** CHEN, I. R.; LI, S. T.; YEN, I. L. Adaptive QoS Control Base don Benefit Optimization for Video Servers Providing Differentiated Services. *Multimedia Tools and Applications*, volume 25 Issue 2 pages: 167-185. Fevereiro de 2005.
- 21.** MATSUKAWA, T.; SATAKE, T.; YAMADA, J. Metrics for Detection of QoS Degradation over Wireless LAN. *Proceedings of the 2005 Symposium on Applications and the Internet (SAINT'2005)*, volume 00. Janeiro de 2005.
- 22.** BOURAS, C.; GKAMAS, A.; PRIMPAS, D. et al. Performance Evaluation of the Impacto of QoS Mechanisms in an IPv6 Network for IPv6-Capable Real-Time Applications. *Journal of Network and Systems Management*, volume 12 Issue 4. Dezembro de 2004.
- 23.** *Computational Management Science*. Springer - Verlag GmbH, Issue: volume 1, number 1, pages: 1-14. Dezembro de 2003.
- 24.** LEEL, I. H.; KIML, S. J. A QoS Improvement Scheme for Real-Time Traffic Using IPv6 Flow Labels. *Lecture Notes in Computer Science*, volume 3043/2004, pages: 278-285. Abril de 2004.