

T.225

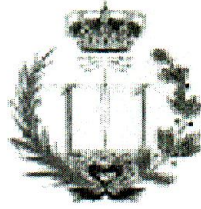
Reserv

UNIVERSIDAD DE SEVILLA
SECRETARÍA GENERAL

Queda registrada esta Tesis Doctoral
al folio 54 número 29 del libro
correspondiente.

Sevilla, 3 de mayo 2002
El Jefe del Negociado de Teles.

Rafael María Estepa Alonso



UNIVERSIDAD DE SEVILLA
ESCUELA SUPERIOR DE INGENIEROS

TESIS DOCTORAL

CONTRIBUCIONES AL
SOPORTE DE CALIDAD EN REDES DE VOZ SOBRE IP

AREA DE INGENIERÍA TELEMÁTICA

DEPARTAMENTO DE SISTEMAS Y AUTOMÁTICA

UNIVERSIDAD DE SEVILLA

Depositado en DPTO. ING. SISTEMAS Y AUTOMÁTICA.
de la ESC. TECN. SUP. INGENIEROS
de esta Universidad desde el día 7/05/02
hasta el día 28/05/02

Sevilla 7 de MAYO

del 2002

EL DIRECTOR DE

Autor: Rafael María Estepa Alonso
Director: Dr. Juan M. Vozmediano Torres

Rafael María Estepa Alonso

Abril 2002



Rafael María Estepa Alonso

Juan M. Vozmediano Torres

FDO: RAFAEL M^a ESTEPA ALONSO

FDO: JUAN M. VOZMEDIANO

A mis padres y a Lourdes

Resumen

Las redes de Voz sobre IP (VoIP) permiten el transporte de voz digitalizada sobre redes de conmutación de paquetes que emplean la familia de protocolos TCP/IP. La ventaja económica de utilizar una única infraestructura de red, junto con el amplio grado de normalización de las redes de VoIP, permiten augurar un gran despliegue de las mismas, en detrimento a las actuales redes de telefonía basadas en la conmutación de circuitos (RTC).

Para la progresiva implantación de las nuevas redes de VoIP en el mercado de las comunicaciones vocales, es necesario que las redes de conmutación de paquetes TCP/IP sobre las que se apoyan puedan ofrecer niveles de calidad de servicio (QoS) en el transporte de datos sensibles al retardo, similares a los ofrecidos por las redes RTC. Para ello se han desarrollado protocolos como RSVP, MPLS o DiffServ que permiten la dispensación de un trato prioritario a ciertos paquetes, posibilitando la existencias de diversos niveles de calidad en el transporte a través de la red IP.

Por tanto, sobre las redes TCP/IP así mejoradas se podrán ofrecer servicios de telefonía con distintos niveles de calidad y, previsiblemente, con tarifas asociadas al nivel seleccionado. Esta situación representa una ventaja sobre el servicio de telefonía tradicional, donde el nivel de calidad es único. El nuevo rol que deben desempeñar los operadores de VoIP en este escenario, es gestionar la provisión de una calidad variable en las conversaciones telefónicas con el menor consumo de recursos posible.

El objetivo fundamental de la Tesis es la definición de un modelo que permita a los operadores de VoIP ofrecer a sus clientes una conversación telefónica con la calidad requerida por ellos, minimizando la utilización de los recursos de transporte.

Para la definición de dicho modelo es necesario especificar dos aspectos:

- Cómo alcanzar un determinado nivel de calidad en la llamada minimizando los recursos utilizados en la red de transporte.
- Qué señalización se utilizará para comunicar las diversas entidades que componen el modelo.

La primera tarea se aborda proponiendo un algoritmo que permite, en tiempo real, hallar los parámetros óptimos para conseguir el nivel de calidad especificado. Dicho algoritmo se apoya en un modelo que permite predecir la calidad que alcanzará una conversación telefónica, conocidos los parámetros de funcionamiento.

La señalización que se utilizará en el modelo se basa en la Arquitectura de Referencia

propuesta de la ETSI para QoS en redes de VoIP. Para poder utilizarla, y debido a que las normas que definen dicha Arquitectura se encuentran aun incompletas, se ha realizado una especificación de los parámetros de calidad en lenguaje ASN.1.

El modelo final ha sido implementado y validado en un banco de pruebas diseñado a tal efecto, lo que ha permitido confirmar el correcto funcionamiento del mismo.

Abstract

Voice over IP networks (VoIP) allow the transport of digitalized voice over the TCP/IP family protocols packet switching networks. The economic advantage of utilizing a unique transport network and the highly developed recommendations about this subject provides a optimistic future for this kind of networks.

Those data which are delay sensible need a quality in their transport similar to the quality offered by the Plain Old Telephony Service (POTS). To facilitate the implementation of VoIP networks, it would be desirable that TCP/IP networks were quality of service (QoS) enabled, as the best effort quality model can not offer any warranties about bounding delay or packet loss. Certain protocols (i. e. RSVP, MPLS or DiffServ) have been designed to prioritize specific information flows (i. e. voice sample packets) in any network load condition.

Thus, these enhanced TCP/IP networks will be able to offer a telephony service with different quality levels which can be related to a different billing. This is a goal with respect to the current telephony service in which the same high level of quality is imposed to all the customers. The new rol for VoIP carriers is to manage this advantage minimizing the network resources consumption.

The aim of this Thesis is to **define a model which allow VoIP operators offer a customers-selected quality minimizing the transport network resources** which is supposed to have a certain cost for operators.

To define such a model it is necessary to specify:

- How to reach to a certain level of quality in a call minimizing the transport network resources.
- Signalling between the different entities involved in the model.

The first item needs to define a real-time algorithm which calculate the most adequate parameters in the communication devices. That algorithm lies in a model that allows to predict the conversation quality known certain working parameters.

The signalling utilized is based in the reference architecture proposed in the ETSI recommendations for providing QoS in VoIP networks. Some work needed to be done in order to complete certain undefined points in the recommendations. Also a formal specifying the quality parameters needed to be done in order to implement the model.

The proposed model have been implemented and validated in a test bed designed for this purpose, obtaining satisfactory results.

Agradecimientos

Desde aquí quiero agradecer el apoyo de todas aquellas personas que han contribuido al desarrollo de la presente Tesis Doctoral. Especialmente quisiera dejar constancia de mi gratitud a D. Juan Manuel Vozmediano Torres, director de la misma, por su continua colaboración y valiosos consejos que han permitido llevar esta Tesis a buen puerto.

También quisiera manifestar mi más sincero agradecimiento a los miembros del proyecto PISCIS, financiado por CICYT y la UE (C. N. 1FD97-1003-C03-03), en cuyo marco se ha desarrollado la presente Tesis Doctoral. De entre ellos quisiera destacar a mi hermano Antonio, por sus inestimables consejos y observaciones. Asimismo he de dar las gracias a los miembros del Área de Ingeniería Telemática, en especial a Ángel García por sus charlas *filosóficas* sobre la realización de Tesis Doctorales, que han supuesto para mí una continua fuente de motivación. Gracias también a mi profesor y amigo D. Ladislao Rubio.

Quisiera finalizar mencionando el apoyo moral que tanto mis padres como mis hermanos María José y Juan Enrique me han prestado durante el tiempo dedicado a la realización de la presente Tesis Doctoral. A mi mujer, Lourdes, he de agradecer las innumerables horas que ha dedicado a escuchar la continua evolución de la misma, sin su paciencia y apoyo hubiera sido difícil conservar el optimismo en algunos momentos.

Muchas gracias a todos ellos.

Índice general

1. Introducción y Objetivos	1
1.1. Introducción	1
1.2. Motivación y entorno de la Tesis Doctoral	2
1.3. Objetivos de la Tesis Doctoral	3
1.4. Organización de la memoria	3
I Estado del Arte	5
2. Calidad de Servicio en la conversación	7
2.1. Concepto de Calidad de Servicio	7
2.2. Factores físicos	9
2.3. Métodos de medida de la calidad en telefonía	16
2.3.1. Métodos Subjetivos	16
2.3.2. Métodos Objetivos	17
2.4. Conclusiones	22
3. Esquemas existentes para el soporte de VoIP	23
3.1. Modelo de referencia de la ITU-T: H.323	23
3.2. Propuesta del IETF para VoIP: SIP	27
3.2.1. Señalización de QoS en redes TCP/IP	30
3.3. Modelo de QoS propuesto por la ETSI	35
3.3.1. Introducción	36
3.3.2. Caracterización de la QoS	39

3.3.3.	Entidades funcionales e Interfaces normalizadas de QoS	41
3.3.4.	Reserva de QoS a través de dominios de servicio y transporte	48
3.3.5.	Procedimientos de señalización de QoS	51
3.4.	Conclusiones	52
II	Contribuciones	55
4.	Modelo para el soporte de QoS con mínimo coste	57
4.1.	Requisitos del modelo	58
4.2.	Arquitectura propuesta	61
4.3.	Especificaciones de Diseño	62
4.3.1.	Ubicación de las Entidades Funcionales	62
4.3.2.	Ubicación de la señalización de calidad	63
4.3.3.	Comienzo y Fin de las llamadas	64
4.4.	Funcionamiento Básico	66
4.5.	Entidades Funcionales del modelo	70
4.5.1.	Clientes con Soporte de Calidad (QoSM del Dominio de Usuario)	71
4.5.2.	Gestor de Calidad de Servicio (QoSM del Dominio de Aplicación)	72
4.5.3.	Gestor de Recursos de Transporte (TRM)	76
4.5.4.	Entidad de Política de Transporte (TPE)	77
4.5.5.	Función de Interconexión (ICF)	79
4.5.6.	Función de Transporte (TF)	80
4.6.	Módulo de Optimización	82
4.6.1.	Requisitos de partida	83
4.6.2.	Algoritmo propuesto	85
4.7.	Validación del Modelo	91
4.7.1.	Pruebas Generales de Señalización e Interfuncionamiento	92
4.7.2.	Pruebas de Robustez en la Señalización	93
5.	Conclusiones y Líneas de avance	95
5.1.	Conclusiones	95

5.2. Líneas de avance	97
A. Definición de parámetros y primitivas	99
A.1. Grupos de parámetros	99
A.1.1. QoS Service Class	99
A.1.2. Codec Type and Packetization	100
A.1.3. Transport QoS Parameters	100
A.1.4. Traffic Descriptor	101
A.1.5. Caller and Called IDs	101
A.1.6. Transport Addresses	102
A.1.7. Application Data Transport Protocol	102
A.1.8. Packet Transport Protocol	103
A.1.9. QoS Policy	103
A.1.10. QoS Mechanism	103
A.2. Primitivas de QoS	104
A.2.1. Interfaz QC1	104
A.2.2. Interfaz QS4	105
A.2.3. Interfaz QT2	106
A.2.4. Interfaz QI4	108
A.3. Procedimientos de señalización	109
A.3.1. Encapsulado de mensajes de QoS	111
A.3.2. Especificación de las PDU de QoS	113
A.3.3. Procedimiento auxiliar de Registro	115
B. Diseño de las aplicaciones	117
B.1. Cliente de QoS	117
B.1.1. Análisis estático	117
B.1.2. Análisis dinámico	119
B.2. Servidor QoSM	125
B.2.1. Análisis estático: GateKeeper	125
B.2.2. Análisis estático: QoSM	127
B.2.3. Análisis dinámico: GateKeeper	129

B.2.4. Análisis dinámico. QoS	131
B.2.5. Optimización	136
B.3. Servidor QoSPE	139
B.3.1. Análisis estático	139
B.3.2. Análisis dinámico	141
B.4. Entidades de transporte	141
B.4.1. Servidor de QoS. Análisis estático	142
B.4.2. Servidor de QoS. Análisis dinámico	147
B.4.3. Módulos instalables. Análisis estático	149
B.4.4. Módulos instalables. Análisis dinámico	151
C. Cálculo de los parámetros de entrada al Modelo-E	155
C.1. Introducción	155
C.2. Determinación de los valores de entrada al Modelo-E	155
C.3. Medida del Retardo en Terminales basados en PC	157
D. Interfaz de Gestión	163
D.1. Introducción	163
D.2. Diseño	164
D.2.1. Parte Servidor	164
D.2.2. Parte Cliente	166
D.3. Implementación	167
E. Librerías	171
E.1. PWLib	171
E.1.1. Estructura de aplicaciones PWlib	171
E.1.2. Traductor ASN.1	172
E.2. OpenH323	174
E.3. NISTNet	175
E.3.1. Características	175
E.4. Netfilter	176
E.4.1. Arquitectura netfilter	176

E.4.2. Librería IPTClib	176
-------------------------------	-----

Índice de figuras

2.1. Esquema general de los métodos objetivos basados en señales	18
2.2. Entradas utilizadas por el Modelo-E para el cálculo de la calidad de una llamada telefónica	19
2.3. Relaciones entre los parámetros del Modelo-E	21
2.4. Relaciones entre factor R y MOS	21
3.1. Torre de protocolos H.323	24
3.2. Señalización en la llamada H.323	26
3.3. Establecimiento de llamada en SIP	28
3.4. Ejemplo de servidores SIP	29
3.5. RSVP extremo-a-extremo	32
3.6. Arquitectura DiffServ. Estas funcionalidades se pueden dar en cada encaminador con capacidad DiffServ.	33
3.7. Etiqueta MPLS utilizada para encapsular la cabecera IP.	34
3.8. Esquema estructural de la arquitectura TIPHON	36
3.9. Arquitectura general TIPHON con control de QoS extremo a extremo por plano de servicio	38
3.10. Arquitectura general TIPHON con control de QoS extremo a extremo por plano de transporte	38
3.11. Parámetros de QoS de usuario, aplicación y transporte	40
3.12. Arquitectura general TIPHON con control de QoS extremo a extremo por plano de transporte	43
3.13. Modelo de flujos de información de QoS	47
3.14. Reserva de los presupuestos de QoS TIPHON	49
3.15. Elementos funcionales involucrados en un establecimiento de llamada mediante intermediario	52

3.16. Flujos de información en un establecimiento de portadora mediante intermediario	53
4.1. Funcionamiento de la optimización en el modelo	58
4.2. Arquitectura del Modelo	61
4.3. Diagrama del Modelo Propuesto	67
4.4. Diagrama SDL del cliente QoS	73
4.5. Diagrama SDL del Gestor de Calidad de Servicio	75
4.6. Diagrama SDL del TRM	78
4.7. Diagrama SDL del TPE	79
4.8. Diagrama SDL de la ICF	80
4.9. Diagrama SDL de la TF	81
4.10. Diagrama Causal de la influencia de los parámetros sobre la calidad	82
4.11. Diagrama Causal de la influencia de los parámetros en el coste	83
4.12. Distribución de puntos por MOS	87
4.13. Banco de Pruebas	92
4.14. Fichero de captura de tramas	94
A.1. Elementos funcionales involucrados en el establecimiento de llamada	111
A.2. Flujos de información	112
B.1. Jerarquía del cliente de voz	118
B.2. Jerarquía simplificada del cliente de voz	120
B.3. Establecimiento de llamada	121
B.4. Recepción de llamada	123
B.5. Intercambio de mensajes en el interfaz QC1	124
B.6. Jerarquía de clases del Gatekeeper – QoSM	126
B.7. Jerarquía de clases del QoSM	128
B.8. Procesamiento de ARQ (solicitud de admisión para efectuar llamada)	132
B.9. Procesamiento de ARQ (solicitud de admisión para recibir llamada)	133
B.10. Ordenación de códecs según coste	134
B.11. Jerarquía de clases del servidor QoSPE	140
B.12. Subestructura de registro	143

B.13. Subsistema de proceso	144
B.14. Subsistema de carga de módulos	146
B.15. Proceso independiente de servicio	148
B.16. Módulo de interfaz QT2	150
B.17. QT2. Proceso de petición de adquisición de recursos	152
B.18. QT2. Proceso de petición de liberación de recursos	153
C.1. Medidas de retardo en los terminales	160
C.2. Señal recibida y emitida tras atravesar la red.	160
D.1. Lectura de bloque sin reajuste	165
D.2. Lectura de bloque con reajuste	165
D.3. Visor de estado	166
D.4. Interfaz del monitor Web	167
D.5. Consola de monitorización	168
E.1. Jerarquía de clases para el manejo de sockets	173
E.2. Jerarquía de clases empleada por el parser ASN.1	174

Índice de cuadros

2.1. Influencia de los factores físicos en la percepción subjetiva	9
2.2. Codecs estandarizados	15
2.3. Tabla de Puntuación MOS	16
3.1. Grupos de parámetros QoS	48
A.1. Parámetros transportados por las primitivas de QoS	110
B.1. Valores límite en vectores válidos	137
C.1. Valores por omisión en el cálculo del Modelo-E	156
C.2. Factor de degradación debido al codec I_e^{codec}	157
C.3. Factor de degradación debido a la pérdida de paquetes I_e^{PL}	158
C.4. Componentes Utilizados	159
C.5. Resultados del Experimento	161

Capítulo 1

Introducción y Objetivos

1.1. Introducción

Históricamente se han desarrollado redes dedicadas al transporte de un solo tipo de información, lo que ha dado lugar a la coexistencia de diversas redes especializadas, como la red telefónica que soporta tráfico con información vocal sensible al retardo, o Internet, para el soporte del tráfico de datos.

La instalación y mantenimiento de una sola infraestructura de red para el transporte de datos, voz y vídeo disminuye costes a los operadores de red y reduce el tiempo que se tarda en ofrecer nuevos servicios, lo que proporciona una ventaja competitiva en el actual mercado libre de las telecomunicaciones.

La tecnología de conmutación utilizada por excelencia para el transporte de voz es la conmutación de circuitos, basada en la reserva de recursos extremo a extremo a lo largo de todo el trayecto que une origen y destino. Esta tecnología, si bien ofrece retardos mínimos en la transferencia de información se muestra ineficiente para el transporte de tráfico de datos debido a que la información a transmitir no se genera de forma continua.

En las redes especializadas en el transporte de datos la tecnología utilizada es la conmutación de paquetes, que mediante la multiplexión estadística facilita el aprovechamiento máximo de la infraestructura de la red a costa de retardos variables y pérdidas de paquetes. Entre las redes que utilizan este tipo de conmutación se encuentra la popular Internet, que emplea la familia de protocolos TCP/IP, y goza de una gran difusión entre los usuarios finales.

La flexibilidad para introducir nuevos servicios, el bajo coste de los nodos de red y el mejor aprovechamiento de la infraestructura instalada, hacen que la tecnología de conmutación de paquetes TCP/IP, sea la que mayor interés económico despierte como método de conmutación único en la infraestructura de red.

Las redes de Voz sobre IP (VoIP) permiten el transporte de voz digitalizada sobre redes de conmutación de paquetes que emplean la familia de protocolos TCP/IP. La ventaja económica de

utilizar una única infraestructura de red, junto con el amplio grado de normalización de las redes de VoIP, permiten afirmar que estas redes irán sustituyendo progresivamente a las actuales redes de telefonía basadas en la conmutación de circuitos.

1.2. Motivación y entorno de la Tesis Doctoral

Para la provisión del servicio de telefonía en las redes TCP/IP es necesaria la especificación de nuevos protocolos que permitan realizar dos tareas básicas:

- Transporte de los medios (voz digitalizada) entre origen y destino.
- Tareas realizadas por la señalización en las redes de telefonía tradicionales (RTC), como el establecimiento y liberación de conexiones, así como la provisión de servicios avanzados.

Para este último cometido existen en las redes de VoIP dos esquemas posibles. Uno de ellos ha sido normalizado por la ITU-T en su Recomendación H.323 y asociadas. El otro ha sido recientemente definido por el IETF y se denomina SIP (Session Initiation Protocol).

Respecto al transporte de medios en las redes TCP/IP, el IETF ha definido un protocolo que permite el envío de paquetes con datos sensibles al retardo, (como la de voz digitalizada), denominado RTP (Real Time Protocol) [1].

RTP fue diseñado para permitir a los receptores la compensación de la variación del retardo que sufren los paquetes IP, así como la posible llegada fuera de orden de dichos paquetes. Junto con RTP se definió otro protocolo denominado RTCP (Real Time Control Protocol), con el objeto de transportar información extremo a extremo sobre la calidad de la transmisión (tasa de pérdida de paquetes y variaciones de retardos).

Normalmente RTP y RTCP se aplican sobre UDP, pues el esquema de retransmisión de TCP no resulta adecuado para los datos que deben transportarse con retardos mínimos (datos de voz digitalizada). Por ello, RTP y RTCP no tienen ninguna influencia sobre el comportamiento de la red IP ni sobre la calidad que ésta ofrece.

Para la progresiva implantación de las nuevas redes de VoIP en el mercado de las comunicaciones vocales, es necesario que las redes de conmutación de paquetes TCP/IP sobre las que se apoyan puedan ofrecer niveles de calidad de servicio (QoS) en el transporte de datos sensibles al retardo, similares a los ofrecidos por las redes RTC. Para ello, se han desarrollado protocolos como RSVP, MPLS o DiffServ que permiten la dispensación de un trato prioritario a ciertos paquetes, posibilitando la existencias de diversos niveles de calidad en el transporte a través de la red IP.

Por tanto, sobre las redes TCP/IP así mejoradas se podrán ofrecer servicios de telefonía con distintos niveles de calidad y, previsiblemente, con tarifas asociadas al nivel seleccionado. Esta situación representa una ventaja sobre el servicio de telefonía tradicional, donde el nivel de calidad es único. El nuevo rol que deben desempeñar los operadores de VoIP en este escenario, es gestionar la provisión de una calidad variable en las conversaciones telefónicas con el menor consumo de recursos posible.

El deseo de contribuir al desarrollo de un servicio de telefonía para las redes de VoIP donde el nivel de calidad sea seleccionable por el usuario constituye el principal motivo que ha conducido a la realización de la presente Tesis.

El entorno de la Tesis estará formado por una red de VoIP que responde al modelo propuesto por la ITU-T, sin conexión con las redes de conmutación de circuitos. El principal motivo para ello es que la calidad en las redes de conmutación de circuitos (como la RTC), es muy superior a la que presentan las redes de VoIP, por lo que apenas representa influencia sobre la calidad total. Además, la calidad en las redes RTC se encuentra suficientemente estudiada en las Recomendaciones de la ITU-T (serie P), por lo que su inclusión en este estudio provocaría una complejidad innecesaria desde el punto de vista de las aportaciones realizadas.

1.3. Objetivos de la Tesis Doctoral

El objetivo fundamental de la Tesis es la **definición de un modelo que permita a los operadores de VoIP ofrecer a sus clientes una conversación telefónica con la calidad requerida por ellos, minimizando la utilización de los recursos de transporte.**

Para llevar a cabo esta tarea se contemplan las siguientes metas parciales:

- Analizar los factores físicos que afectan a la calidad de las conversaciones en VoIP.
- Especificar un modelo que permita obtener un nivel de calidad elegido por los usuarios para sus llamadas, minimizando los recursos de red utilizados. Para ello será necesario:
 - Definir la señalización a utilizar en dicho modelo
 - Determinar un algoritmo que permita alcanzar un determinado nivel de calidad en la llamada con el mínimo consumo de recursos de transporte.
- Validar dicho modelo. Para ello se realizará un protocolo de pruebas sobre una implementación del modelo.

1.4. Organización de la memoria

La memoria de la Tesis se ha estructurado de la siguiente manera:

En el capítulo 2 se presenta el estado del arte en el ámbito de la calidad del servicio de telefonía tradicional. En él se describen y analizan los conceptos de calidad en telefonía, así como los factores que la afectan en las redes de VoIP. Por último, se realiza una descripción de los principales métodos que permiten determinar el grado o nivel de calidad de una conversación telefónica, haciendo especial énfasis en el Modelo-E, método basado en parámetros definido por la ETSI.

Los modelos existentes para la provisión del servicio de telefonía en las redes TCP/IP son introducidos en el capítulo 3, donde se analizan las propuestas de la ITU-T y del IETF para VoIP. Tras ello, se describe el esquema de señalización de la ETSI para el soporte de calidad en las redes de telefonía IP, en el que se ha basado la señalización del modelo propuesto en la presente Tesis.

En el capítulo 4 se describe el modelo para obtener una llamada con la calidad seleccionada minimizando los recursos necesarios en la red de transporte, detallándose además los aspectos relacionados con el funcionamiento, validación y algoritmo de optimización de dicho modelo.

Finalmente, las principales contribuciones de la Tesis se resumen en el capítulo 5, apuntándose a una serie de posibles líneas de investigación futuras.

Para una comprensión más profunda del trabajo realizado en la presente Tesis Doctoral se presentan cinco Apéndices al final de la memoria. En el Apéndice A se especifica el código ASN.1 empleado para los parámetros y primitivas utilizados en el modelo propuesto. Así mismo se define el procedimiento de señalización en las diversas interfaces normalizadas.

Las aplicaciones realizadas se abordan en el Apéndice B, donde se describen mediante un análisis estático y otro dinámico.

En el Apéndice C se introducen los procedimientos necesarios para obtener todos los parámetros de entrada utilizados en el Modelo-E. Así mismo se acompaña este anexo de un protocolo de medidas que permite obtener el retardo de los terminales utilizados en VoIP.

El Apéndice D realiza una descripción de la interfaz de Gestión creada para la monitorización de las diversas entidades.

Por último el Apéndice E es utilizado para describir someramente las principales librerías utilizadas en el desarrollo del modelo propuesto en el capítulo 4 de la presente Tesis Doctoral.

Parte I

Estado del Arte

Capítulo 2

Calidad de Servicio en la conversación telefónica: Factores Dominantes y Métodos de Medida

La calidad en la conversación telefónica de las actuales redes RTC ha sido analizada en profundidad por organismos como la ITU-T o la ETSI. El estudio riguroso de la calidad en la conversación telefónica constituye un requisito previo al análisis de la calidad en las redes de VoIP, ya que la finalidad última de estas redes es ofrecer el servicio de telefonía con una calidad similar a la proporcionada por las redes RTC.

En este capítulo se abordarán todos los aspectos relacionados con la calidad en la conversación telefónica; esto es, concepto de calidad, factores dominantes y métodos de medida.

2.1. Concepto de Calidad de Servicio

Un usuario declarará estar satisfecho con un servicio en la medida que vea cumplidas sus expectativas de calidad. Por tanto, el nivel o grado de calidad en el servicio de telefonía depende exclusivamente de la opinión que tengan los usuarios sobre la calidad de las llamadas.

Existen varios factores que afectan a la opinión que los usuarios tienen sobre la calidad:

- Disponibilidad de la red telefónica: que siempre que deseen efectuar una llamada, puedan hacerla, o que las conversaciones no terminen súbitamente de forma involuntaria debido a un mal funcionamiento de la red.
- Precio o tarifa: un usuario puede declararse satisfecho con un cierto nivel de calidad medio a cambio de un determinado precio.
- Tiempo de espera para el establecimiento de la llamada: tiempos mayores de 3 segundos comienzan a afectar a la opinión de los usuarios sobre la calidad.

- Ventajas de acceso: permitir efectuar llamadas en movimiento (móviles).
- Calidad de la conversación percibida por el oyente en su terminal. Hay que tener en cuenta que:
 - No existe una definición absoluta basada en parámetros físicos de la calidad de la conversación, la única guía para su cálculo es la percepción subjetiva de las personas.
 - La calidad de la conversación es, en última instancia, un fenómeno psicoacústico que envuelve una compleja interacción entre los parámetros del proceso de percepción humana, y los parámetros que pueden ser medidos eléctricamente.
 - En el entorno actual de liberalización de las telecomunicaciones la responsabilidad de ofrecer calidad por parte del operador llega hasta el punto de terminación de red, pues los terminales dependen de la elección de los clientes.

Los cuatro primeros factores pueden ser controlados en tiempo de diseño o bien mediante políticas administrativas. La disponibilidad, al igual que el tiempo de establecimiento de conexión, está relacionada con la capacidad instalada y se mejora sobredimensionando en el diseño de la red. La ventaja de acceso es propia de la red a instalar por lo que no puede nunca cambiarse. Será la calidad de la conversación el factor sobre el cual debe basarse el estudio de la calidad máxima alcanzable por las redes de VoIP, ya que el resto de los factores son equivalentes a los de las redes RTC existentes.

La percepción auditiva humana es un fenómeno altamente no lineal y complejo. En ella existen una serie de factores objetivos (medibles físicamente) así como otros individuales (que dependen de la persona). Por tanto, una expresión cuantitativa de la calidad será siempre una media estadística que incluirá tanto factores objetivos como medias de sensibilidad y factores psicológicos de las personas.

Los parámetros psicológicos más importantes de la percepción auditiva, son:

- Inteligibilidad: calidad en la percepción del significado o del contenido de información de lo que el hablante dice.
- Naturalidad: grado de fidelidad con respecto a la voz del hablante.
- Volumen: grado de volumen absoluto en el lado del receptor.

Aunque existen otros como:

- Calidad del sonido: calidad del sonido que percibe el receptor (parecido a la naturalidad).
- Capacidad de hablar y escuchar: capacidad de mantener una conversación interactiva (hablar y escuchar simultáneamente).
- Esfuerzo del oyente: Para escuchar o bien para mantener la conversación.

Factores Físicos	Parámetros Psicológicos			
	Calidad General	Volumen	Inteligibilidad	Naturalidad
Pérdidas	+++	+++	+++	+
Ancho de Banda	+++	++	++	+++
Ruido	++		++	+
Distorsión	++		++	++
Efecto Local	+	+		++
Eco (hablante)	+++		+	++
Retardo	++			+

Cuadro 2.1: Influencia de los factores físicos en la percepción subjetiva

Los parámetros psicológicos anteriores dependen en gran medida de los parámetros físicos (medibles) de la red y terminales, como nivel de la señal, ruido, retardo o eco (ver cuadro 2.1). En la siguiente sección se analizarán los principales parámetros físicos que intervienen en la calidad de la conversación.

2.2. Factores físicos

Los parámetros físicos de terminales y red determinan la percepción subjetiva de la calidad en la conversación. Estos factores han sido ampliamente estudiados por la ITU-T para las redes de telefonía tradicionales. A continuación se describen cuales son y los valores que toman en las actuales redes de telefonía fija. Así mismo se realizará una particularización de dichos factores a las nuevas redes de VoIP.

■ Índices de sonoridad

Los índices de sonoridad pueden describirse como las pérdidas electroacústicas que sufre la señal de voz al transmitirse por la red telefónica.

Se compone de diferentes términos:

- *OLR* (Overall Loudness Rating) Índice de sonoridad general:

Define las pérdidas existentes entre el sonido que emana de la boca del hablante y el que recibe el oído del llamado. Un *OLR* de 0 implicaría que el oído receptor está situado junto a la boca del hablante. En [2] se proporciona el índice de sonoridad general óptimo (aproximadamente 10 dB).

- *SLR* (Send Loudness Rating) Índice de sonoridad en emisión:

Define las pérdidas existentes entre la potencia de la señal sonora que sale de la boca del hablante y la señal eléctrica que sale de su aparato telefónico (terminal), incluyendo las pérdidas en el bucle de abonado. Está formada por las pérdidas de la conversión electroacústica

del micrófono (sensibilidad [3]), las posibles pérdidas por reflexión en la bobina híbrida del aparato telefónico y las pérdidas en el bucle de abonado. Los valores recomendados pueden encontrarse en [2] (aproximadamente 7 dB)

- *RLR* (Receive Loudness Rating) Índice de sonoridad en recepción:

Define las pérdidas existentes entre la señal que llega al bucle de abonado del receptor y la potencia de la señal sonora que sale del auricular del aparato telefónico. Su valor óptimo es aproximadamente 3 dB.

- *CLR* (Circuit Loudness Rating) Índice de sonoridad del circuito:

Pérdidas en los diferentes circuitos que componen una conexión telefónica. Cada trayecto tendrá unas pérdidas que deberán sumarse.

Las pérdidas en una comunicación telefónica que atraviesa por i trayectos cumplen que:

$$OLR = SLR + \sum_{i=1}^n CLR_i + RLR$$

Cuando la transmisión de la señal sea digital (como es el caso de VoIP) las pérdidas en el circuito son siempre nulas, por ello:

$$OLR = SLR + RLR$$

En los terminales de VoIP basados en ordenadores PC las ganancias pueden ajustarse fácilmente por los propios usuarios a niveles confortables. Esto permite seleccionar los valores óptimos de pérdidas de sonoridad, por lo que se puede controlar la influencia sobre la calidad total.

■ Efecto Local (Sidetone)

En una conversación telefónica, el oído en contacto con el auricular puede escuchar la propia voz y el ruido de la sala a través del aparato telefónico. Ambos casos corresponden al denominado efecto local. Cuando es la voz del hablante la que llega a su propio oído a través del terminal telefónico, se denomina efecto local del hablante. Si es el ruido ambiente el que llega al oído a través del teléfono se denomina efecto local del oyente. Hay que tener en cuenta que los mecanismos de escucha hacen que los sonidos percibidos por el oído que no está en contacto con el auricular sean prácticamente ignorados por el cerebro.

Las pérdidas en el camino del efecto local del hablante deben permanecer dentro de ciertos límites para que se pueda mantener una conversación confortable. Si las pérdidas son muy altas quien habla no escucha su propia voz y tiende a subir la intensidad de la misma,

mientras que si son muy bajas tiene la sensación de estar hablando demasiado alto y tiende a reducir su propio volumen.

El camino del efecto local del oyente debe, sin embargo, tener unas pérdidas mayores que un cierto umbral para que el ruido ambiente no afecte a la conversación.

Tanto el efecto local del hablante como el del oyente se miden mediante índices de sonoridad, denominándose: *STMR* (SideTone Masking Rating) a las pérdidas de sonoridad en el camino del hablante y *LSTR* (Listener SideTone Rating) a las del camino del oyente.

En los terminales telefónicos analógicos estas pérdidas provienen de las desadaptaciones en impedancias y bobinas híbridas.

Las pérdidas *STMR* y *LSTR* se encuentran relacionadas por la siguiente fórmula:

donde el factor *D* depende sólo del diseño del teléfono. Para micrófonos lineales (no de carbón) valdrá 3 en el caso de teléfonos normales y 0 para teléfonos cortos. En el caso de micrófonos de carbón el valor es aún mayor, aunque la calidad del teléfono es menor. El factor *D* se discute ampliamente en [3].

Los valores ideales para *LSTR* son mayores que 15 dB, por lo que *STMR* deberá estar comprendido entre 10 y 15 dB.

En el caso de redes de VoIP con terminales digitales basados en PC es posible, mediante ajuste del mezclador, obtener los niveles óptimos de estas pérdidas.

■ Eco y Estabilidad

Se pueden definir dos tipos distintos de eco:

- Eco del hablante: cuando el hablante recibe su propia voz con un cierto retardo
- Eco del oyente: Cuando el oyente recibe múltiples ecos de la señal directa.

La pérdida de potencia entre la señal que emite el hablante y la que se recibe en forma de eco se denomina *TELR* (Talker Echo Loudness Rating), y miden el eco del hablante. La relación entre las potencias de la señal directa y la que recibe el oyente en forma de múltiples ecos reflejados se denomina *WEPL*, y mide el eco del oyente.

La principal fuente de eco en las redes de telefonía tradicionales es la desadaptación de impedancias y de bobinas híbridas en el paso de 2 a 4 hilos. En redes digitales la única fuente de eco es el propio terminal, bien por acoplamiento eléctrico de los circuitos o bien, mayormente en caso de terminales manos libres, por el camino acústico. En los terminales digitales las pérdidas de eco debidas al terminal se denominan *TCLw* (Weighting Terminal Coupling Loss) y en los terminales analógicos *EL* (Echo Loss). La expresión analítica para las pérdidas de eco es:

$TELR = SLR + RLR + EL$, donde *SLR* y *RLR* incluyen las posibles pérdidas del circuito.

Cuando existan múltiples caminos de eco en la práctica sólo se tomará el valor del que tenga la menor pérdida (mayor influencia por tanto).

Si el sumatorio de pérdidas menos ganancias en el camino de la señal es negativo, la red será inestable, pudiéndose entonces producir fenómenos de acoplamiento. En general, para

asegurar la estabilidad de una red el sumatorio de todas las pérdidas y ganancias en bucle abierto debe ser superior a 4 dB.

El efecto que el eco produce en la calidad de la conversación depende fundamentalmente de dos factores: el valor de las pérdidas en el camino de eco (cuantificadas mediante los parámetros *WEPL* y *TELR*) y el retardo que sufra la señal en su camino de eco. Cuando mayor sea dicho retardo el efecto del eco se hace más patente, empeorando así la calidad de la comunicación. Para mantener el eco en niveles aceptables se debe asegurar unas pérdidas grandes y un retardo pequeño. Con retardos menores de 25 ms el control del eco se puede realizar únicamente aumentando las pérdidas. No obstante, si las pérdidas (*OLR*) son muy grandes, la señal que llega al otro extremo resulta demasiado débil, por lo que se suele adoptar una solución de compromiso que sitúa las pérdidas en bucle abierto ($A_{b.a.}$) en torno a 14 dB. La expresión analítica de dichas pérdidas es:

$$A_{b.a.} = OLR + EL - \sum_{i=1}^n G_i$$

Donde G_i representa las ganancias de cada tramo del trayecto.

Para retardos mayores a 25 ms se hace necesaria la utilización de equipos especiales como canceladores y supresores de eco, que también influyen en la calidad percibida por el oyente.

En un escenario compuesto únicamente por dos terminales basados en PC unidos por una red IP, la principal fuente de eco reside en el camino acústico, ya que todos los circuitos son a 4 hilos. Por tanto, las fuentes de eco se localizan en los terminales siempre y cuando no existan pasarelas hacia redes de conmutación de circuitos. Los terminales con altavoces y micrófono son los que más se verán afectados por el eco acústico, ya que las pérdidas de eco en dichos casos son muy pequeñas. Cuando la amplificación de la voz en los altavoces es excesiva pueden producirse problemas de estabilidad, al alcanzar la señal el punto de saturación de los circuitos de conversión analógico-digital. Para los terminales digitales con microteléfono o con auriculares, las pérdidas de eco son típicamente mayores de 45 dB [4], lo que implica un efecto casi inapreciable del eco acústico. En terminales con altavoces y micrófonos, se necesita un control de eco muy exhaustivo, ya que las pérdidas de eco pueden caer hasta 15 dB (menores incluso que en las redes de telefonía tradicionales).

■ Retardo Total (Boca - Oído)

El retardo total tiene una doble influencia sobre la calidad de las conversaciones telefónicas:

- A mayor retardo, mayor es la influencia del eco.
- Retardos grandes afectan a la interactividad de las conversaciones, consiguiendo la apariencia de una comunicación semi-duplex.

En [5] se pueden encontrar la influencia del retardo sobre la opinión de calidad. En función del retardo total se tiene que:

25 ms no existen dificultades para mantener una conversación normal

25-150 ms la calidad es aceptable para la mayoría de los usuarios pero se hace necesaria la utilización de canceladores de eco

150-400 ms hay dificultades para seguir una conversación (caso de las comunicaciones vía satélite). Es necesaria la utilización de canceladores de eco de gran rendimiento (supresores de eco).

≥ 400 ms las dificultades para mantener una conversación interactiva hacen que la calidad sea inaceptable para la mayoría de los usuarios

En las redes de VoIP el retardo total se encuentra repartido entre el retardo debido a los terminales y el retardo en la red IP. La influencia de la red IP en el retardo es doble, ya que la fluctuación del mismo deberá ser compensada mediante colas en los terminales, que añaden un retardo extra al que ya proporciona la red. Las causas del retardo en las redes IP son los tiempos en las colas de espera de cada encaminador atravesado. También influye en el retardo de red (aunque en menor medida) el tiempo de transmisión en cada nodo.

Para conocer la influencia del retardo en las redes IP se puede hacer uso de simuladores (en tiempo de diseño) o de medidas reales en la red (en tiempo de explotación).

En los terminales las principales causas de retardo son:

- **Codecs:** Tienen varias fuentes de retardo: retardo algorítmico, que depende de la potencia de procesamiento del terminal y del tipo de codec, retardo de predicción, debido a que ciertos algoritmos necesitan muestras futuras de la voz, y retardo de llenado de tramas. Por ejemplo, el codec G.723.1 necesita un retardo de predicción de 7,5 ms y la duración de sus tramas es de 30 ms.
- **Cabeceras de los protocolos:** Cada protocolo de las diferentes capas, añade su cabecera y consume tiempo de procesamiento. El retardo que se induce debido a ello dependerá del número de tramas de voz que porten los paquetes IP. Si estos llevan pocas tramas por paquete, la sobrecarga de cabeceras será mayor, mientras que el tiempo de llenado de los paquetes IP será menor.
- **Retardo de transmisión:** en el terminal, se incrementa de forma lineal
- **Retardos de almacenamiento *Encolado*:** Debido a los dispositivos asociados a un sistema operativo, como zonas de almacenamiento (*colas*) en la tarjeta de sonido, interfaz de red, y aplicaciones de telefonía.
- **Cola de emisión de sonido en el terminal destino (*cola de reproducción*):** Tras recibir una trama de voz, el terminal destino esperará un cierto tiempo antes de emitir el sonido contenido en dicha trama por el sistema de audio. Esto se realiza para compensar posibles variaciones en el retardo de las tramas que llegan de la red. Los paquetes que lleguen después del tiempo programado para su emisión al sistema de audio (incluida la *cola de reproducción*), serán descartados y se contarán como paquetes perdidos. No obstante, una cola de emisión grande incrementará el retardo total de la comunicación.

■ **Ruido y distorsión de cuantización**

El ruido presenta gran influencia en la calidad de las conversaciones cuando se utilizan redes analógicas. El efecto del ruido ambiente en un sitio se caracteriza por su valor en dB(A) o también por la potencia sonora que llega al microteléfono en dBPa(A). Se pueden encontrar tablas con los valores medios para diferentes tipos de condiciones en [6].

El ruido ambiente se introduce en la conversación en dos puntos distintos: en el lado del hablante *Pos* dB(A) y en el lado del receptor *Por* dB(A).

El efecto que tiene el ruido ambiente estará relacionado con los parámetros *Pos*, *Ds*, *SLR* y *RLR*. Por otro lado, el ruido que se recibe procedente del ambiente de la sala dónde se encuentra el propio oyente está relacionado con *Por* y *LSTR*, ya que dicho ruido se escucha sólo a través del oído pegado al auricular (influencia del sidetone).

Respecto a la distorsión de cuantización, solamente afecta cuando el número de codecs en tandem es grande. Para llamadas de clientes de VoIP a clientes de otras redes se producirá como mucho una transcodificación, por lo que no tendrá apenas influencia en la calidad. Los codecs con baja tasa de bits serán tratados como equipos con funciones especiales, y cuantificados en el siguiente subapartado.

■ Equipos con funciones especiales

Dispositivos de control de eco: Son necesarios para controlar el eco cuando el tiempo medio de transmisión en un sentido supera los 25 ms. Su misión es suprimir o compensar las señales que se acoplan en el camino del eco. Ambos actúan sobre el parámetro de pérdida en el camino del eco, por lo que disminuyen la influencia de este. Por otro lado, pueden introducir degradación en la recepción de la voz debido a cortes muy bruscos en los caminos de recepción y emisión [7].

Además del incremento de las pérdidas de eco, sería deseable que los canceladores de eco minimizaran los efectos de otros problemas que afectan a la calidad, realizando tareas como ajuste automático de ganancia [8], reducción del nivel de ruido, o inserción de ruido confortable en los periodos de inactividad de la voz. En la Recomendación [9] se citan las principales características que debe reunir un cancelador de eco.

En las redes de VoIP la comunicación entre clientes atraviesa únicamente encaminadores de nivel IP, por lo que la posible existencia de canceladores o supresores de eco se limita a los terminales de usuario.

Debido a los grandes retardos típicos en las redes de VoIP, en caso de terminales manos libres es necesario la utilización de canceladores de eco, por lo que también influirán sobre la calidad total.

Codecs de baja tasa de bit: Existen varios tipos de codecs de baja tasa. La ITU-T ha normalizado algunos, como: G.723.1, G.726, G.728, G.729. La ITU-T ha clasificado los codec en tres tipos:

- Codecs de forma de onda: reproducen la forma de onda original de la forma más fiable posible. Son más robustos ante problemas pero su tasa de bit es más alta.
- Vocoders o códecs de parámetros: utilizan el modelo de la generación de voz para transmitir sólo algunos parámetros importantes de la misma. Este tipo de

	G.711	G.729	G.723.1	GSM 6.10	GSM 6.20	GSM 6.60
Fecha	1972	1995	1995	1988	1994	1996
Bit Rate (kbit/s)	64	8	6,3 / 5,3	13	5,6	12,2
MIPS	0,1	22	16/18	2,5	17,5	15,4
T. trama (ms)	0,13	10	30	20	20	20
Predicción -retardo-(ms)	0	5	7,5	0	0	0
Robustez	normal	buena	normal	buena	buena	buena
Max. núm. en serie	14	2	2 / 1	2	2	2
Tipo Codec	PCM	CS-ACELP	MP-MLP /ACELP	RPE-LTP RPE-LTP	VSELP	ACELP

Cuadro 2.2: Codecs estandarizados

codec tiene como característica más brillante el bajo régimen binario que genera a la salida. Sin embargo, son muy susceptibles a pérdidas y no proporcionan una calidad muy buena

- Códex híbridos: combinan lo mejor de ambos tipos, ofreciendo un régimen binario moderado y una buena calidad de voz.

La influencia de los codecs de baja tasa de bit sobre la calidad se puede hallar mediante el empleo de test subjetivos normalizados [10].

La utilización de codec de baja tasa de bit es muy común en las redes de VoIP pues permiten minimizar el consumo de régimen binario. La influencia que estos presentan sobre la calidad general es grande, ya que tanto la distorsión intrínseca al codec como la posible influencia de pérdidas de tramas se computan en el factor de equipos especiales (Ie).

Cualquier codec de baja tasa de bit [11] provocará una mayor distorsión y ruido de cuantización que la codificación efectuada por el codec G.711. La respuesta de estos codec de baja tasa de bit, también puede verse afectada por parámetros tales como nivel de la señal de voz, tono de la voz, ruido ambiente, errores de transmisión y variaciones del retardo.

El impacto de la pérdida de paquetes en la calidad de la voz depende directamente del codec que se emplee. Con el codec G.711 la pérdida de un paquete supondrá un silencio de 0,125 ms de duración, imperceptible para la mayoría de los oyentes, mientras que en un codec de baja tasa de bit, la pérdida de un paquete afecta a la calidad de la voz durante un periodo de tiempo mucho mayor. El efecto que estas pérdidas de paquetes suelen producir en la voz que se está escuchando es el de una interrupción inesperada y tras esto una reanudación abrupta de la misma (*clipping*) [12].

En la tabla 2.2 se reflejan las principales características de los codecs de banda estrecha [11].

Calidad	Puntuación
Excelente	5
Buena	4
Aceptable	3
Pobre	2
Mala	1

Cuadro 2.3: Tabla de Puntuación MOS

2.3. Métodos de medida de la calidad en telefonía

Para poder cuantificar la calidad de la conversación de forma objetiva es necesario eliminar los aspectos individuales de las medidas. Inicialmente esto se lograba mediante métodos de medida subjetivos, es decir, experimentos de escucha cuidadosamente diseñados y realizados a un gran número de personas de tal forma que ofrecieran una idea de la verdadera calidad. Los participantes debían escuchar las muestras de voz y calificarlas con un valor entero en una escala de opinión como la que presenta el cuadro 2.3

La puntuación resultante se utilizaba para calcular un valor medio a juicio de los participantes, y se denomina “Valor Medio de Opinión” (MOS, Mean Opinion Score).

Recientemente se han introducido los métodos objetivos que permiten realizar una estimación de la calidad basándose en muestras de la conversación, intentando predecir el resultado que se obtendría en un test subjetivo. Estos nuevos métodos son atractivos ya que pueden ser automatizados y requieren menos esfuerzo que la realización de test subjetivos.

A continuación se describen los principales métodos utilizados para la evaluación de la calidad de las conversaciones.

2.3.1. Métodos Subjetivos

Los resultados en los tests subjetivos de opinión deben ser fiables y repetibles. En ellos debe tenerse en cuenta los siguientes factores:

- **Material de voz:** La percepción depende del hablante, su pronunciación, el lenguaje, el contenido de las muestras grabadas, la habitación donde se realiza el test y las características de los equipos.
- **Diseño del experimento:** Los resultados también dependen de la nacionalidad de los hablantes, experiencia previa en test de escucha, duración de las sesiones y orden de presentación de las muestras de voz.
- **Condiciones de Escucha:** El volumen o intensidad de las muestras de voz, así como los equipos utilizados (auriculares, teléfonos, ...) pueden influir en la nota final.

En [13] se ofrecen las guías para la realización de experimentos fiables. Existen varias formas de recoger la opinión de los participantes, las principales son:

Absolute Category Rating (ACR) En él, los participantes deben dar una nota absoluta de la calidad de las muestras de voz escuchadas. La nota oscila entre 1 (Calidad mala) y 5 (calidad excelente) (ver cuadro 2.3). El resultado se denomina *MOS* y se obtiene estadísticamente.

Degradation Category Rating (DCR) Con el método anterior se tiende a perder sensibilidad para pequeñas diferencias de calidad, por lo que se puede aplicar este otro método. En él se presentan pares de muestras grabadas, donde una es la referencia y el participante votará la diferencia entre ésta y la otra. El resultado se denomina *DMOS*.

Entrevistas Otra forma de averiguar la calidad es realizando entrevistas a los usuarios de la red telefónica.

Test Conversacionales Son más completos que los tests de audición. En ellos, dos participantes deberán conversar a través de una conexión. De esta forma se simula con más realismo las redes. Permiten tener en cuenta parámetros como el retardo, el eco o las variaciones de nivel en la voz durante una conversación. Se valora también la capacidad de mantener una conversación interactiva, cosa que no puede ser tenido en cuenta con los anteriores métodos. Información sobre la realización de dichas pruebas puede encontrarse en [14].

Existen otros tipos de test subjetivos como:

- **Doble Habla:** Permiten evaluar en detalle la calidad de la conversación en los periodos donde ambos interlocutores hablan a la vez. Con ellos se puede medir los efectos de la cancelación de eco.
- **Habla y Escucha:** Cubren la calidad en los periodos que una persona habla y se escucha a sí mismo. Con ellos se evalúan dificultades en la escucha y la influencia del eco.
- **Sólo escucha:** Diseñados para evaluar diferencia en la calidad de los distintos terminales o de algoritmos de implementación.

2.3.2. Métodos Objetivos

Los métodos objetivos permiten predecir el nivel de calidad que se obtendría en los test subjetivos de forma sistemática, por lo que pueden ser utilizados para el dimensionamiento de la red en la fase de diseño.

La determinación objetiva de la calidad se puede realizar de dos maneras:

1. **Métodos basados en señales:** Comparan muestras de voz antes y después de cruzar la red.
2. **Métodos basados en parámetros:** Utilizando un modelo y los parámetros de la red permiten predecir la calidad que tendrán las conversaciones.

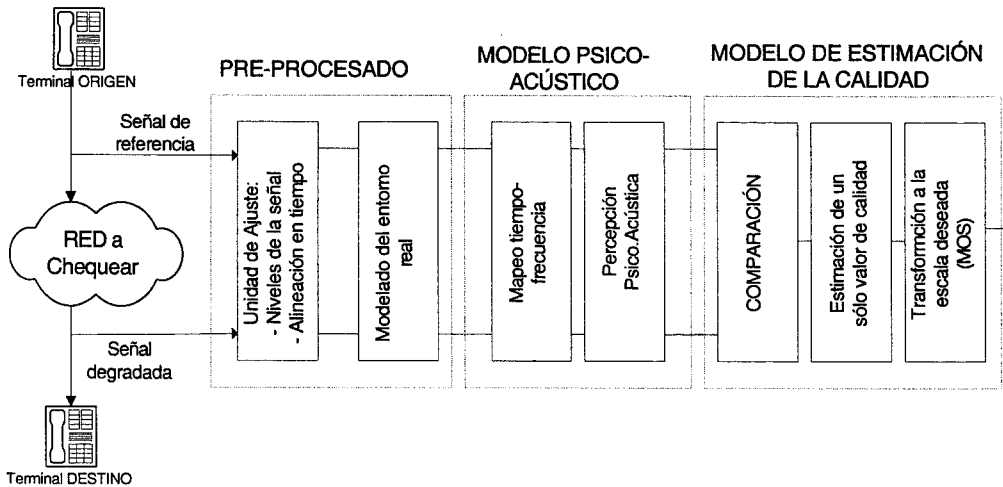


Figura 2.1: Esquema general de los métodos objetivos basados en señales

Basados en Señales

Utilizan dos señales como entrada: la original, utilizada como referencia, y la correspondiente a la salida tras atravesar la red de la que se pretende averiguar la calidad.

El procesamiento se estructura en tres etapas:

1. **Pre-Procesado:** en el que las dos señales de entrada se ajustan en retardo, potencia, duración y frecuencia. Tras este ajuste inicial se modifican las señales con las condiciones de utilización reales, es decir, en el hablante se utiliza el ruido típico de una habitación y en el oyente se simula ruido, reverberancia y las características del terminal.
2. **Modelo Psico-Acústico:** Se transforma la señal al modelo que utiliza el cerebro humano en la percepción. Para ello se realiza un mapeo tiempo-frecuencia, se ajustan los coeficientes de predicción lineal del modelo de escucha a corto plazo, bandas críticas, enmascaramientos en tiempo y frecuencia, interpretación psicoacústica del volumen, ...
3. **Modelo de estimación de la calidad de la conversación:** Se compara la imagen que el cerebro tiene de la señal original y de la recibida. Para ello se observa la diferencia en los parámetros individuales y se realiza una estimación media, que puede ser la distancia euclídea, distancia ponderada, diferencias asimétricas, ... Tras esto se mapea el resultado a la escala MOS, obteniendo así la predicción.

En la figura 2.1 se muestra el esquema de un modelo típico.

Actualmente los métodos más utilizados se conocen por las siglas QPSM, PEAQ y PESQ [15, 16, 17]. En [18] se puede encontrar una comparativa donde se discute la idoneidad de cada uno para su utilización en las redes de VoIP.

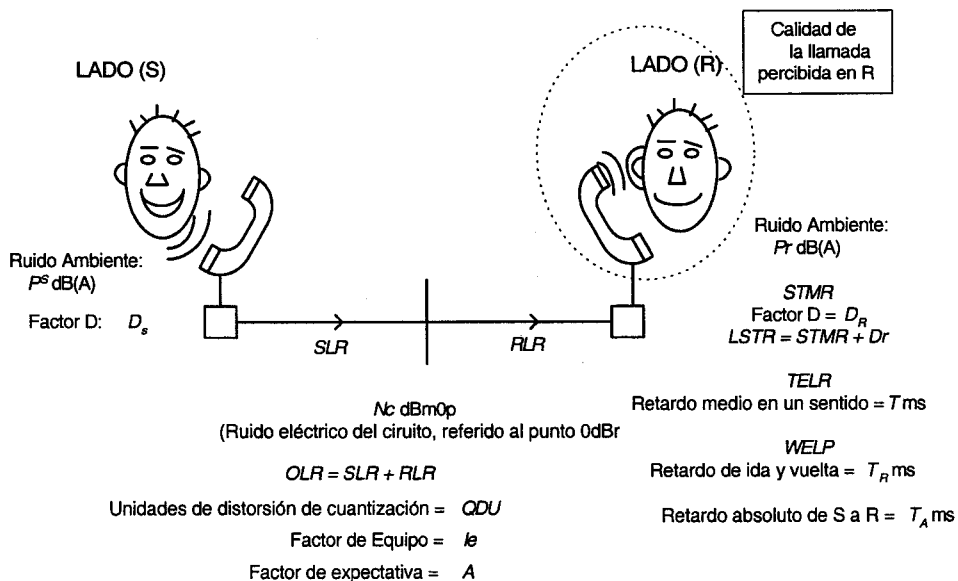


Figura 2.2: Entradas utilizadas por el Modelo-E para el cálculo de la calidad de una llamada telefónica

Basados en parámetros: Modelo-E

En ellos se toma como entrada las características de redes y terminales, permitiendo predecir la calidad que tendrán los usuarios finales. De los diversos métodos [19] el más evolucionado es el modelo propuesto por la ETSI en [6], denominado Modelo-E. Dicho modelo ha sido también adoptado por la ITU-T en [20].

El Modelo-E predice la calidad subjetiva de una llamada telefónica partiendo de los parámetros de la red de transmisión. Se basa en el principio de que todos los problemas de transmisión tienen un efecto aditivo en la escala psicológica [6].

El Modelo-E predice la calidad que percibirá el oyente situado en el lado R, utilizando para ello como entradas los factores que se muestran en la figura 2.2.

El resultado final de aplicar el modelo es un número llamado factor de Tasa de Transmisión (R), con valores entre 0 y 100. El factor R puede ser mapeado a MOS.

La fórmula utilizada para el cálculo de R es:

$$R = R_0 - I_s - I_d - I_e + A \tag{2.1}$$

donde:

- R_0 representa la relación señal a ruido en el punto de 0 dBr, puede expresarse como:

$$R_0 = 15 - 1,5 \cdot (SLR + N_0) \tag{2.2}$$

El término SRL ya ha sido discutido en el apartado anterior. El ruido total N_0 (en dBm0p) es la suma de las potencias del ruido eléctrico del circuito (N_e), y del ruido equivalente del circuito (N_o^S y N_o^R) provocado por el ruido ambiente de las habitaciones (P^S y P^R), y el ruido de planta del lado R (N_{fo}^R)

Valores típicos de estos parámetros se encuentran en [21].

- I_s Tiene en cuenta los problemas de la transmisión que ocurren de manera simultánea al sonido de la voz: volumen muy alto (I_{otr}), pérdidas de efecto local no ideales (I_{st}) y distorsión de cuantización PCM (I_q).

$$I_s = I_{otr} + I_{st} + I_q \quad (2.3)$$

- I_d Incluye los problemas relacionados con el retardo de la transmisión de la señal original: eco de hablante y oyente (I_{dte} y I_{dle}), y retardo excesivamente grandes en ausencia de eco (I_{dd}):

$$I_d = I_{dte} + I_{dle} + I_{dd} \quad (2.4)$$

- I_e Tiene en cuenta los problemas debido a la utilización de equipos especiales como codecs de baja tasa de bit o equipos canceladores y supresores de eco. El impacto de la pérdida de paquetes en las redes IP también se incluye aquí.

En [12] se pueden encontrar valores provisionales. Como este factor depende en gran medida de la tecnología empleada se encuentra en continuo estado de evolución.

- A Factor de expectativa. Representa el hecho de que un usuario puede estar satisfecho con altos niveles de degradación de la señal a cambio de otro tipo de ventajas. Por ejemplo, los usuarios pueden encontrar aceptable la calidad de la voz en GSM debido a la ventaja de poder hablar en movimiento (ventaja de acceso). Esa misma calidad en las redes fijas puede ser considerada inaceptable. En [6] se pueden encontrar valores provisionales.

Los diferentes términos que componen los factores del Modelo-E pueden ser calculados partiendo de medidas efectuadas directamente sobre la red. Las fórmulas empíricas que permiten obtener los términos tratados anteriormente a partir de las medidas efectuadas se encuentran en [6]. La figura 2.3 resume las interdependencias de los distintos factores y los parámetros físicos medibles en la red.

Una guía básica sobre la aplicación del Modelo-E en las redes se encuentra en [21]. Esta guía incluye valores típicos, configuraciones de referencia y ejemplos de aplicación.

La figura 2.4 muestra la relación existente entre el factor R y la MOS.

En el Apéndice C de la presente memoria se proporciona una guía que permite el cálculo de los distintos parámetros utilizados como entradas en el Modelo-E.

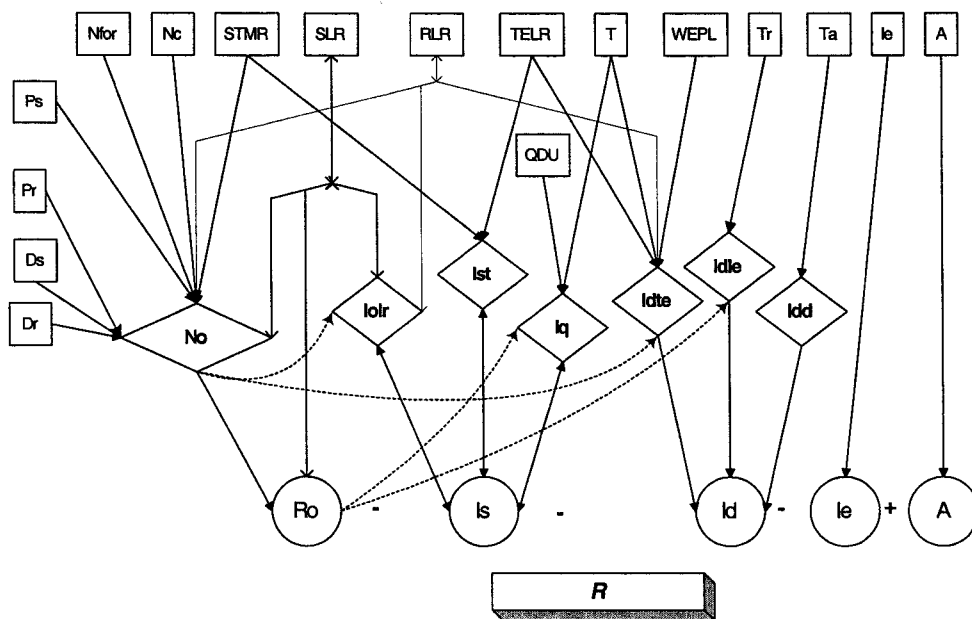


Figura 2.3: Relaciones entre los parámetros del Modelo-E

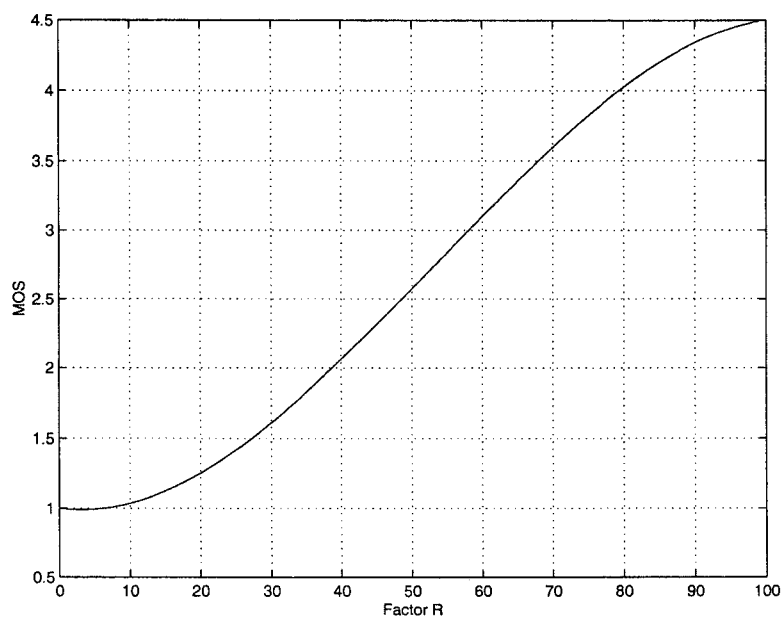


Figura 2.4: Relaciones entre factor R y MOS

2.4. Conclusiones

Este capítulo ha definido el concepto de calidad en el servicio de telefonía aplicable a las redes de VoIP. Posteriormente se han presentado los principales factores que determinan la calidad de la conversación, particularizados tanto para las redes de telefonía tradicionales como para las redes de VoIP.

El concepto de calidad en el servicio de telefonía no depende de la red de transporte utilizada, por lo que debe la calidad debe ser cuantificada utilizando los métodos aplicados en las redes RTC. La ITU-T define tres tipos diferentes de métodos que permiten medir la calidad de la conversación:

- Encuestas o Test subjetivos: son los más fiables, pues el concepto de calidad es en última instancia un fenómeno de percepción subjetiva.
- Métodos objetivos basados en comparación de señales: mediante un modelo de percepción humana permiten predecir la calidad comparando una señal antes y después de atravesar la red. Son menos costosos de realizar que los tests subjetivos.
- Métodos objetivos basados en parámetros: a través de un modelo matemático, permiten predecir la calidad en la conversación partiendo del valor de los factores físicos existentes en la red y terminales.

En el modelo para el soporte de calidad propuesto en el capítulo 4, es necesario utilizar un método que, partiendo de la calidad requerida por el usuario para una llamada, obtenga los valores de los parámetros de terminales y red que la proporcionan. La restricción de trabajar en tiempo real aconseja el uso de un método basado en parámetros, siendo el Modelo-E el seleccionado, ya que es el más evolucionado de los existentes.

Capítulo 3

Esquemas existentes para el soporte de VoIP

Actualmente hay dos modelos de referencia para el soporte de VoIP, propuestos respectivamente por los organismos de normalización ITU-T e IETF. En ambos casos el modelo para el soporte de VoIP se centra en la especificación de protocolos de señalización que permiten realizar llamadas y ofrecer servicios de telefonía avanzados.

La ETSI, en su proyecto denominado TIPHON, realiza una armonización entre ambas propuestas y aborda temas como el soporte de la calidad, el direccionamiento o la interconexión con las redes RTC.

En este capítulo se realizará un resumen de los modelos existentes para el soporte de VoIP, así como de los aspectos relacionados con la calidad en dichos modelos. Posteriormente se detallará la propuesta de la ETSI para el soporte de calidad en las redes de VoIP. Dicha propuesta constituye la base de la señalización utilizada en el nuevo modelo propuesto en el capítulo 4.

3.1. Modelo de referencia de la ITU-T: H.323

La ITU-T fue el primer organismo de normalización que desarrolló una norma para la transmisión de tráfico multimedia sobre las redes de conmutación de paquetes [22]. Dicha norma se recoge en la Recomendación H.323 [23], y fue publicada en 1996 con el nombre de *Sistemas y Terminales de telefonía visual sobre redes de área local sin garantías de calidad de servicio*. La principal aportación de esta Recomendación es el desarrollo de un conjunto de protocolos de señalización que permiten controlar el establecimiento, mantenimiento y liberación de conexiones multimedia sobre redes basadas en la conmutación de paquetes. Para la transmisión de los medios (audio, vídeo y datos) se utilizan los protocolos RTP y RTCP, introducidos brevemente en el primer capítulo. En 1998 apareció la segunda versión de la Recomendación H.323v2 bajo el nombre *Packet based multimedia communications systems*. La versión 4 fue aprobada en noviembre de 2000 y es la que permanece vigente en el momento de la elaboración del presente trabajo. H.323

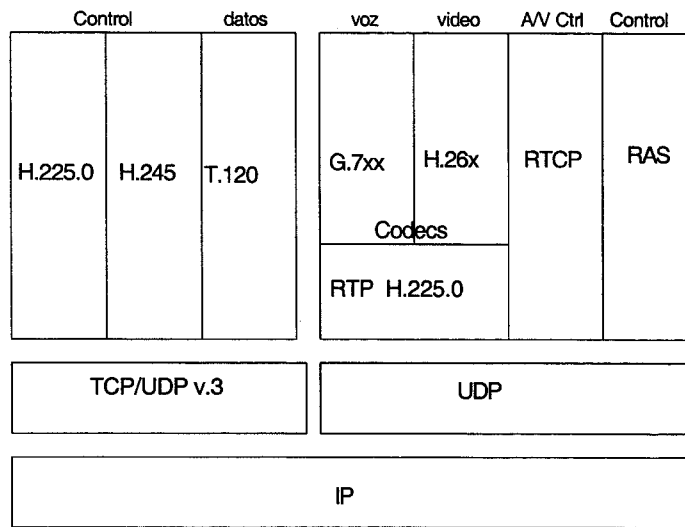


Figura 3.1: Torre de protocolos H.323

define 4 tipos de elementos funcionales:

1. **Terminal H.323:** Es un terminal conectado a una red IP, que proporciona en tiempo real la comunicación bidireccional con otro terminal H.323, pasarela o MCU (unidad de control multipunto). Un terminal debe soportar, al menos, una transmisión de voz digitalizada con el codec G.711. Opcionalmente puede soportar otros tipos de codecs para voz, transmisión de voz y vídeo simultánea así como transmisión de datos junto con las anteriores.
2. **Pasarela H.323 (GateWay):** Es un elemento que permite interoperar a los terminales H.323 con terminales de otras redes de conmutación de circuitos (RTC), como teléfonos fijos o móviles GSM. Las pasarelas se conectan directamente con terminales H.323 o bien con otras pasarelas o terminales pertenecientes a redes RTC. Las principales funciones que realizan son las de adaptación entre flujos de información, y entre los protocolos de control de ambos entornos.
3. **Unidad de Control Multipunto (MCU, Multipoint Control Unit):** Es el elemento que permite soportar comunicaciones multipunto. A diferencia de otros entornos como la RD-SI, la capacidad de transmisión multicast en las redes IP no requiere la utilización de un elemento externo a los terminales para realizar funciones de mezclado de medios. Por esta razón la MCU está dividida en dos partes: el controlador multipunto (MC), que proporciona la capacidad de negociación y control de los miembros del grupo, y el procesador multipunto (MP) que se encarga de realizar las funciones de mezclado de medios (audio, vídeo y datos). Las funcionalidades de la MCU pueden ser integradas dentro de un terminal H.323
4. **Guardián (GK, Gatekeeper):** Es un elemento que proporciona servicios de control centralizados al resto de los elementos. Constituye la base para el desarrollo de servicios y para la

aplicación de esta tecnología en entornos con un número de terminales elevado. El GK es un elemento opcional en la arquitectura, lo que permitió inicialmente el desarrollo de terminales que podían comunicarse directamente entre sí sin necesidad de disponer de GK. Sin embargo, la inexistencia de GK limita el número de servicios ofrecidos. Las funciones que proporciona son, entre otras, traslación de direcciones, autorización de llamadas, gestión y reserva de ancho de banda, servicios de directorio, etc.

La arquitectura de protocolos de H.323 se representa en la figura 3.1. La mayor parte de canales de control utilizan conexiones TCP (también UDP a partir de la versión 3), mientras que el transporte de medios utiliza UDP.

Las entidades H.323 establecen conexiones en diferentes fases. Considerando un escenario sencillo en el que exista un GK, la conexión entre dos terminales registrados en este GK comprende los siguientes pasos (ver figura 3.2):

- a) **Establecimiento de Llamada:** La entidad llamante solicita al GK, mediante el envío del mensaje *ARQ* perteneciente al canal RAS (Registration Admission and Status), autorización para realizar una llamada a un destino. El GateKeeper autoriza dicha llamada y proporciona información sobre el destino (su dirección de red), en la respuesta (mensaje *ACF*). Tras ello se establece una conexión directa con el abonado llamado (canal de señalización H.225 [24]). Sobre esta conexión se envía el mensaje *SET UP* (similar al de Q.931 utilizado en RDSI) que provocará que el abonado llamado contacte con su GateKeeper (el mismo que el llamante en este caso) para solicitar autorización para recibir la llamada. Una vez concedida esta, el llamado devuelve un mensaje *CONNECT* al llamante que indica la aceptación de la llamada.
- b) **Intercambio de Capacidades:** Tras la fase anterior, se establece una nueva conexión (canal de señalización H.245 [25]) con el objeto de negociar las capacidades a utilizar durante la comunicación. Se negocia los codificadores de la voz, el tipo de capacidades (voz, vídeo, ...) y el número de los puertos UDP que van a utilizar los canales que transportan los medios audiovisuales. Esta conexión permanece establecida durante toda la comunicación para posibilitar la modificación de parámetros o la apertura de nuevos canales de medios.
- c) **Intercambio de Información Audiovisual:** En este punto ambos terminales intercambian la información audiovisual (por ejemplo la voz digitalizada) utilizando para ello el protocolo RTP con los puertos establecidos en la fase anterior. Así mismo se establecen canales de control RTCP para monitorizar la calidad de los flujos de información recibidos.
- d) **Terminación de Llamada:** Tras el intercambio de información audiovisual, y al objeto de finalizar la llamada, las entidades H.323 se informan mutuamente y cierran los canales de medios y los canales H.245. Además se informa al GateKeeper para que libere los recursos utilizados y proporcione, entre otras, información de tarificación.

El ejemplo mostrado anteriormente ilustra uno de los problemas detectados en H.323v1: el elevado tiempo de establecimiento de una llamada debido, fundamentalmente, a la necesidad

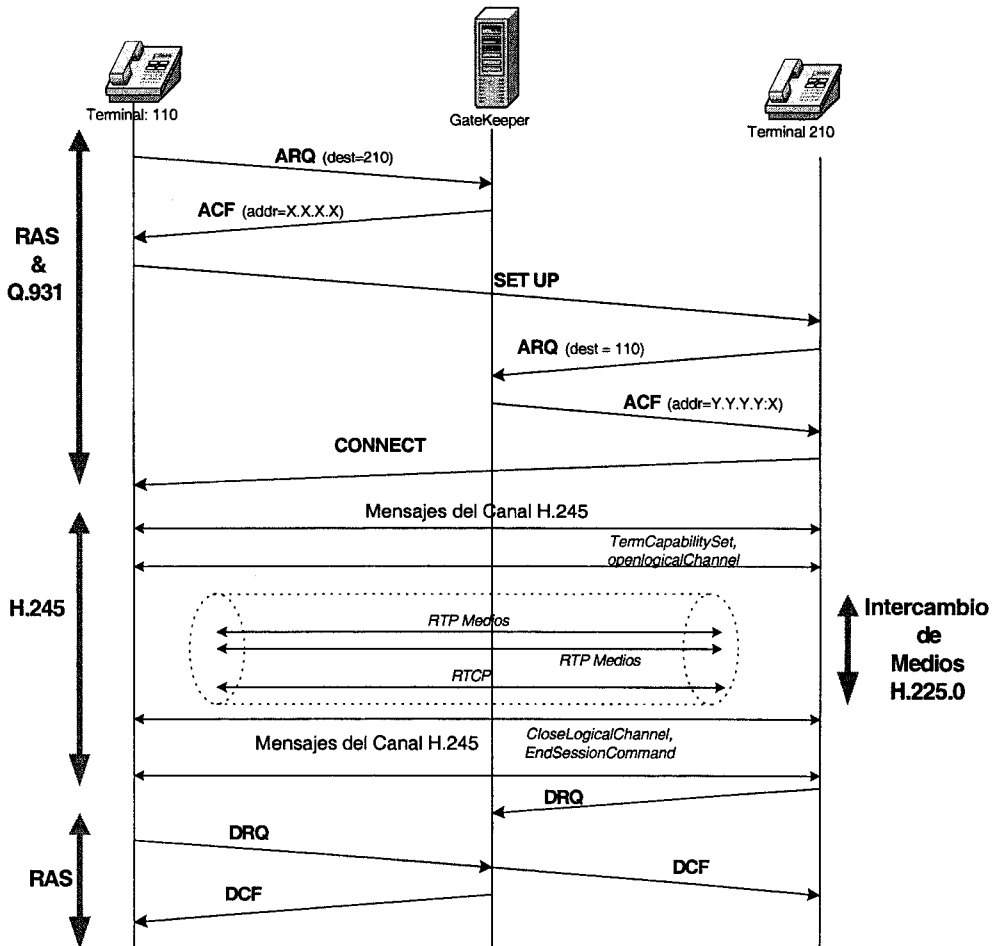


Figura 3.2: Señalización en la llamada H.323

de establecer múltiples conexiones previas al intercambio de información de voz. Esta limitación fue en parte subsanada con la aparición de la versión 2 del protocolo, en la cual se definen los procedimientos de conexión rápida y túneles H.245, que permiten simplificar apreciablemente el proceso de establecimiento de una llamada.

3.2. Propuesta del IETF para VoIP: SIP

El *Session Initiation Protocol (SIP)* [26] es un protocolo de aplicación desarrollado por el IETF dentro del grupo MMUSIC (Multiparty Multimedia Session Control), que se encarga del desarrollo de recomendaciones relacionadas con el soporte de conferencias multimedia. SIP permite a los usuarios participar en sesiones de intercambio de información multimedia soportando mecanismos de establecimiento, modificación y finalización de llamadas.

Uno de los objetivos del grupo MMUSIC consiste en desarrollar mecanismos para informar a los usuarios acerca de las sesiones existentes en la red y de sus requisitos (medios, direcciones, etc.). En este sentido existen dos modos básicos para identificar y participar en sesiones multimedia:

- **Mecanismo de Anuncio:** Las sesiones son anunciadas mediante correo electrónico, páginas web, grupos de noticias o bien mediante protocolos de anuncio de sesiones (SAP, Session Announcement Protocol) como sucede en la red MBONE.
- **Mecanismos de Invitación:** Los usuarios son invitados a participar explícitamente, mediante SIP, en una sesión

SIP ha sido propuesto como mecanismo genérico para el soporte de señalización del servicio de telefonía IP, dado que incluye todas las funciones necesarias, tales como: localización de usuarios, intercambio y negociación de capacidades de los terminales o el establecimiento y liberación de las llamadas. SIP es un protocolo basado en el modelo cliente/servidor: los clientes SIP envían peticiones a un servidor, el cual contesta con una respuesta. Los terminales SIP pueden generar tanto peticiones como respuestas, al estar formados por entidades cliente y servidor.

Los terminales SIP pueden establecer llamadas de voz directamente sin intervención de elementos intermedios, al igual que en caso H.323. La figura 3.3 muestra un ejemplo de conexión entre dos usuarios.

El *usuario1* envía una petición (*INVITE request*) en la que indica al *usuario2* las capacidades de recepción de audio (codificación G.711) y el puerto dónde espera recibir dicho audio (puerto 12345, en este caso). Al recibir la petición, el *usuario2* puede inmediatamente establecer el canal de voz y enviar la aceptación de conexión mediante un mensaje *OK response*, en el que incluye información complementaria para el establecimiento del canal opuesto (codificación GSM, puerto 54321 en este ejemplo) . Tras el intercambio de audio, cualquiera de los dos participantes puede finalizar la llamada mediante el envío del mensaje *BYE request*, que debe ser atendido mediante un mensaje de confirmación (*OK*).

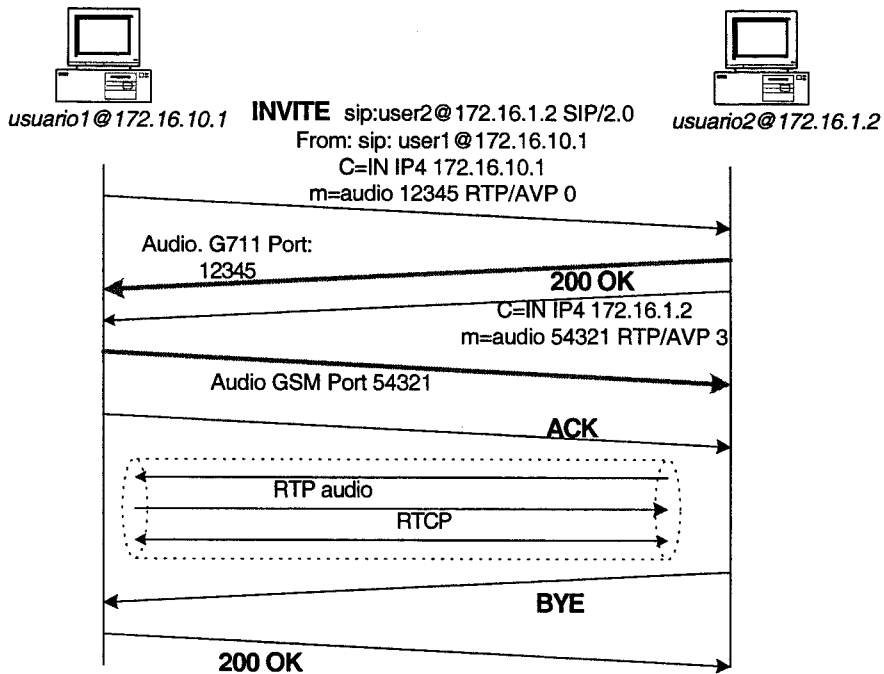


Figura 3.3: Establecimiento de llamada en SIP

Los mensajes SIP son codificados utilizando la sintaxis de los mensajes HTTP/1.1 y el contenido de cada mensaje sigue las recomendaciones del protocolo de descripción de sesiones SDP [27], ampliamente utilizado en el contexto de MBONE para distribuir información sobre las sesiones.

Además de los terminales y las pasarelas a otros tipos de redes, que cumplen la misma función en H.323, la arquitectura SIP define cuatro tipos de servidores:

Servidor Proxy: Realiza el encaminamiento de las peticiones y respuestas hacia el destino final. El encaminamiento se realiza salto a salto de un servidor a otro hasta alcanzar el destino final. Existe un parámetro incluido en las peticiones/respuestas denominado *Via*, que incluye la dirección de los sistemas intermedios que hayan participado en el proceso de encaminamiento. Esto evita bucles y permite forzar que las respuestas sigan el mismo camino que las peticiones. Todo ello afecta únicamente a la información de control, pues el transporte de medios, salvo requerir transcodificación intermedia se realiza directamente entre los terminales origen y destino, mediante los protocolos RTP y RTCP.

Servidor de Redirección: Realiza una función equivalente al servidor Proxy, pero a diferencia de éste no encamina la llamada, sino que contesta a un *INVITE* con un mensaje de redirección que indica cómo contactar directamente con el destino.

Servidor de Registro: Mantiene la localización actual de un usuario. Se utiliza para que los terminales registren la posición en la que se encuentran. Este servidor facilita la movilidad de

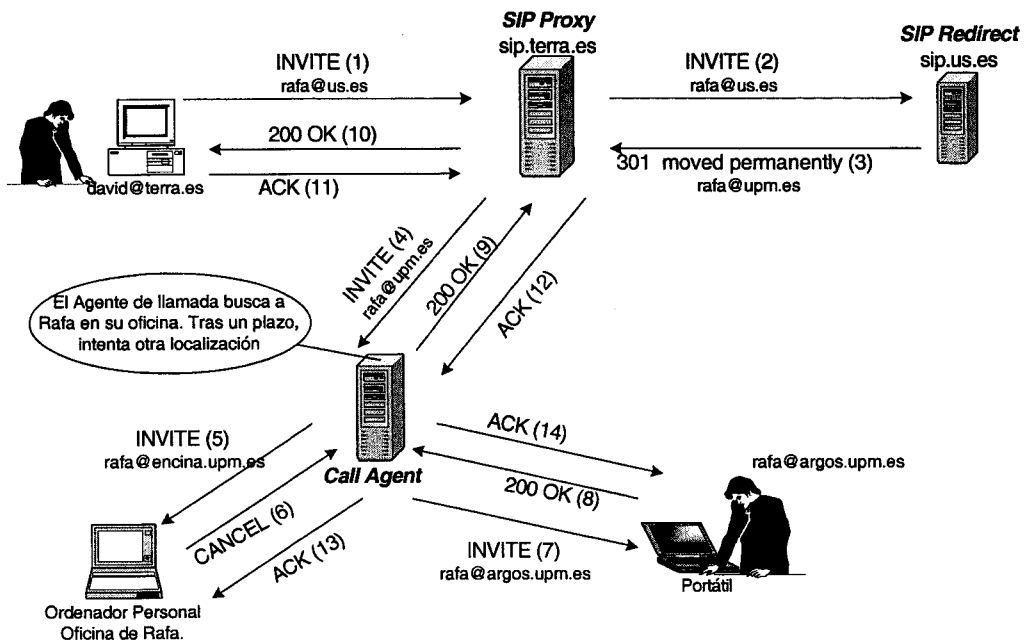


Figura 3.4: Ejemplo de servidores SIP

los usuarios.

Agente de llamada: Realiza las funciones de los tres servidores anteriores, y otra serie de funciones complementarias, como:

- Localizar a un usuario mediante la redirección de la llamada a una o varias localizaciones.
- Implementar servicios de redirección como reenvío si ocupado, reenvío si no contesta, etc.
- Implementar filtrado de llamada en función del origen o el instante de la llamada.
- Almacenar información de administración de las llamadas.
- Realizar cualquier otra función de gestión.

Las direcciones SIP se distinguen mediante identificadores denominados URI (Uniform Resource Identifier), que siguen la estructura:

usuario@máquina

donde *usuario* corresponde a un nombre, identificador o número de teléfono, y *máquina* es el dominio al que pertenece dicho usuario o dirección de red.

La figura 3.4 muestra un ejemplo de interacción entre servidores SIP de los distintos tipos mencionados anteriormente. En él, un usuario (david) trata de establecer una llamada con otro

usuario (rafa) que se encuentra en una localización distinta de la suya habitual. Utiliza para ello los servicios proporcionados por un servidor Proxy, un servidor de Redirección y un Agente de Llamada.

La principal ventaja de SIP frente a H.323 es su simplicidad. Mientras que H.323v1 necesita 5 o 6 intercambios de información previos al intercambio de medios, SIP requiere únicamente uno, que puede utilizar TCP o UDP indistintamente. Esta ventaja queda en parte mitigada con la aparición de las nuevas versiones de H.323.

El modelo propuesto por el IETF, al igual que el de la ITU-T, delegan el soporte de calidad en los mecanismos existentes para ello en la red de transporte. A continuación se describen los principales mecanismos de soporte de calidad existentes en las redes TCP/IP, normalizados por el IETF. La utilización de los mismos es parte fundamental en el soporte de la calidad en este tipo de redes, por lo que serán brevemente presentados es la siguiente subsección.

3.2.1. Señalización de QoS en redes TCP/IP

IP ofrece un servicio de red no orientado a conexión y no fiable, sujeto pues a pérdidas, desencuamamiento y duplicación de paquetes. Estos efectos, junto con el retardo en las colas de los encaminadores, se ven incrementados con el aumento de carga en la red. Por todo ello, el servicio ofrecido en las redes IP se conoce como *best-effort*, y es necesario complementarlo con capas de nivel superior, como TCP, que ofrecen fiabilidad extremo a extremo. A medida que se incrementa el tráfico en la red, el servicio ofrecido por ésta va degradándose. Esto puede causar problemas a las aplicaciones que tienen exigencias de tiempo real, como es el caso de VoIP.

Los protocolos de calidad de servicio (QoS) permiten tratar de forma distinta el tráfico de las aplicaciones con necesidad de tiempo real, del de aquellas que pueden tolerar en su funcionamiento un cierto retraso, fluctuación de éste o pérdidas de paquetes [28].

Actualmente, existen dos estrategias para ofrecer QoS en las redes IP:

- Reserva de Recursos (*IntServ*): los recursos de la red se reparten en función de las peticiones de QoS realizadas por las aplicaciones y de la política de gestión de ancho de banda.
- Priorización (*DiffServ*): el tráfico es clasificado y los recursos de la red se reparten en función de la política de gestión de ancho de banda. Para ofrecer QoS, los encaminadores de la red tratan preferencialmente al tráfico en función de su clase.

Estas estrategias pueden aplicarse tanto a flujos de tráfico individuales como a flujos agregados. Un flujo se define como una corriente de datos unidireccional e individual entre dos aplicaciones (la emisora y la receptora), y se identifica unívocamente por una quintupla (protocolo de transporte y direcciones y puertos origen y destino). Para dar cobertura a las necesidades de QoS existen varios algoritmos y protocolos:

- *Reservation Protocol (RSVP)*: ofrece la señalización necesaria para los servicios integrados. Se suele utilizar aplicado a cada flujo individual.

- *Differentiated services* : ofrece un método sencillo de categorizar y gestionar los flujos agregados.
- *Multi Protocol Label Switching (MPLS)*: ofrece la gestión de ancho de banda para agregados a través del encaminamiento en la red, en función de etiquetas situadas en las cabeceras de los paquetes.

En las siguientes subsecciones se describen brevemente estos protocolos.

Servicios Integrados (*IntServ*) y RSVP

Los servicios integrados [29] hacen posible la coexistencia entre la entrega de datagramas de tipo *best-effort* y clases de servicios de entrega mejoradas respecto al ancho de banda, retardo y pérdida de paquetes. El nivel de QoS ofrecido por estas nuevas clases es programable para cada flujo en función de las peticiones de las aplicaciones finales. Dichas peticiones se envían a los encaminadores usando un protocolo de reserva como RSVP.

IntServ ofrece básicamente dos nuevas clases de servicio (además de la tradicional *best-effort*):

- Servicio Garantizado (*Guaranteed Service*) [30]: este servicio garantiza ancho de banda y establece límites máximos al retardo extremo a extremo. Esto es muy importante para las aplicaciones de VoIP.
- Carga controlada (*Controlled Load*) [31]: este servicio es equivalente al servicio *best-effort* bajo condiciones de baja carga. Es mejor que el *best-effort*, pero no garantiza el cumplimiento de las limitaciones estrictas que garantizan la cobertura de las necesidades del tiempo real.

Respecto al control de tráfico, *IntServ* realiza: control de admisión, clasificación de paquetes y planificación de los paquetes. Además especifica el uso de protocolos de señalización (RSVP)[32], que ofrecen establecimiento y control de la reserva.

Los emisores con capacidad RSVP, caracterizan el tráfico saliente en términos de límites superior e inferior de ancho de banda y variación del retardo. Periódicamente, los emisores transmiten mensajes PATH que contienen esta caracterización del tráfico (*Tspec*) hacia el destino (unicast o multicast). Cada uno de los encaminadores con capacidad RSVP a lo largo del camino de ida, establece un "estado del camino" que incluye la información captada de la dirección origen previa del mensaje PATH.

Para realizar una reserva de recursos los receptores envían un mensaje RESV (petición de reserva) hacia los emisores del flujo. Además del *Tspec*, este mensaje contiene el tipo de servicio solicitado (Carga controlada o Garantizado).

En el camino de vuelta, cuando un encaminador RSVP recibe el mensaje RESV, utiliza el proceso de control de admisión para autenticar la petición y realizar la reserva de recursos para el flujo y manda el mensaje RESV al siguiente encaminador en el camino de vuelta. Las reservas en

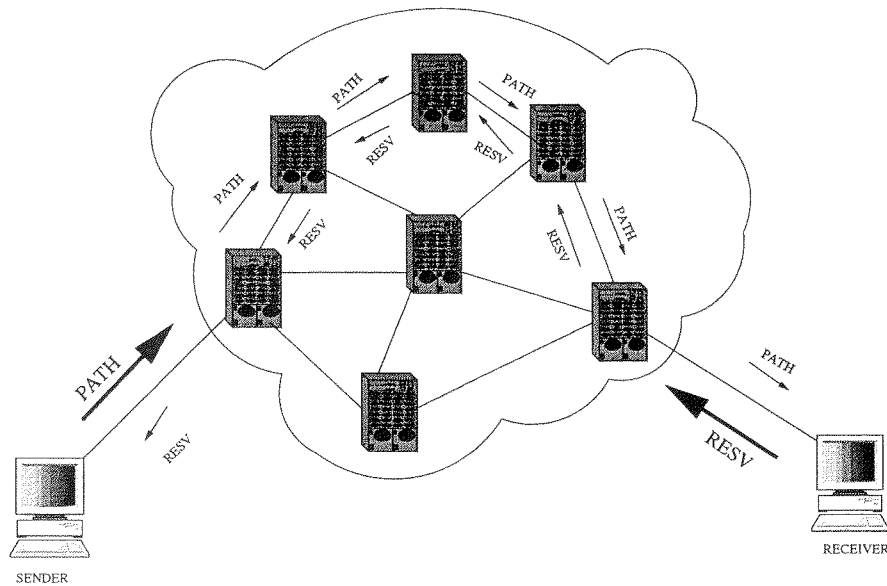


Figura 3.5: RSVP extremo-a-extremo

los encaminadores necesitan un refresco periódico. Por razones de escalabilidad este mecanismo no es aconsejable para los encaminadores del núcleo de la red.

Cuando el último encaminador recibe el mensaje RESV y acepta la petición, envía una confirmación de vuelta al receptor.

El tráfico RSVP puede atravesar encaminadores no-RSVP, pero esta situación crearía un enlace débil en la cadena, donde el servicio volvería a ser de tipo *best-effort*. Otro punto negativo de este protocolo es que la reserva se basa en los receptores y no considera mecanismos especiales para grupos de receptores multicast.

RSVP permite a las aplicaciones solicitar QoS con un alto nivel de granularidad, a costa de una gran complejidad y sobrecarga de cabeceras.

Servicios Diferenciados (Diffserv)

Los servicios diferenciados ofrecen un método simple de clasificación de los servicios ofrecidos a las aplicaciones. Las clases de servicios son identificadas y los paquetes se marcan como pertenecientes a una clase en particular. Los encaminadores examinan las cabeceras para determinar el trato a dispensar al flujo agregado [33].

Actualmente, se han definido dos comportamientos estándar (PHBs) que representan dos niveles de servicio:

- Envío Explícito (*Expedited Forwarding-EF*): tiene un sólo valor de codificación (valor *Diff-serv*). EF minimiza el retardo, fluctuación de éste y ofrece el mayor nivel de calidad de

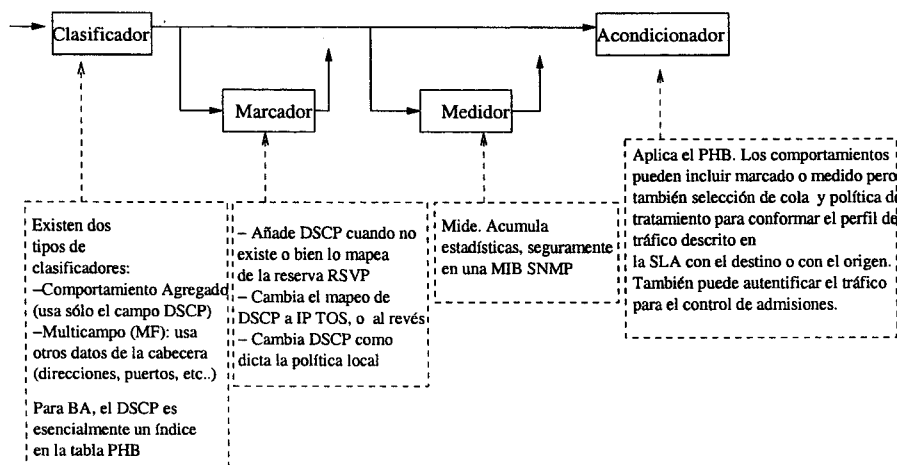


Figura 3.6: Arquitectura DiffServ. Estas funcionalidades se pueden dar en cada encaminador con capacidad DiffServ.

servicio a los agregados. Cualquier tráfico que exceda del perfil de tráfico (traffic profile - definido en la política local) será descartado.

- Envío asegurado (*Assured Forwarding-AF*): tiene cuatro clases y tres preferencias de descarte para cada clase (un total de doce codificaciones). El tráfico AF que exceda de su perfil, no se reparte con la probabilidad garantizada por su perfil de tráfico, por lo que puede demorarse pero no necesariamente descartarse.

Como se muestra en la figura 3.6, los PHBs son aplicados al tráfico por el Acondicionador en el punto de ingreso en la red según el criterio de la política pre-determinada. El tráfico puede ser marcado en ese punto y encaminado según su marca. La marca se quita al salir de la red.

DiffServ supone la existencia de un acuerdo de nivel de servicio (SLA) entre redes adyacentes. El SLA establece el criterio de política y define el perfil del tráfico. Se espera que el tráfico sea tratado en las redes vecinas de acuerdo al SLA. No se ofrecen garantías para el tráfico que no se ajuste a su perfil.

Cuando se utiliza, el mecanismo del protocolo, utiliza los bits del byte DS, que indica el servicio que el paquete debería recibir, y viaja en el campo Type Of Service (TOS) en IPv4 o en el campo Traffic Class Octet (TCO), en IPv6.

La simplicidad de *DiffServ* para priorizar el tráfico le otorga flexibilidad y potencia. A diferencia de RSVP, la cantidad de información de estado depende del número de clases de servicio, y no del número de flujos. Las operaciones complejas de clasificación, autenticación, marcado y conformado del tráfico sólo son necesarias en los bordes de la red, y son los emisores los que solicitan los recursos, no los receptores.

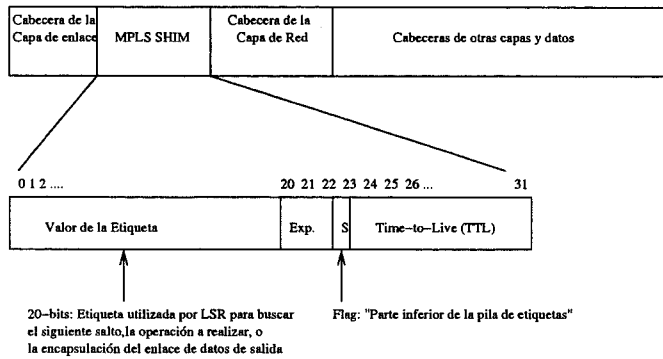


Figura 3.7: Etiqueta MPLS utilizada para encapsular la cabecera IP.

MPLS

El funcionamiento de Multi-Protocol Label Switching (MPLS) [34] se basa en la definición de caminos a través del núcleo de una red IP, para su posterior uso en el envío de paquetes IP.

Cuando un paquete entra en la red, el encaminador frontera de entrada busca la dirección destino del paquete y lo marca con una etiqueta que especifica la ruta, y, opcionalmente los atributos de la clase de servicio (CoS). La idea de MPLS es utilizar las etiquetas para determinar el siguiente salto del camino, de manera que los encaminadores realicen menos trabajo y puedan actuar como simples conmutadores.

Según los paquetes marcados se adentran en la red, cada encaminador utiliza la etiqueta del paquete para seleccionar el destino y, opcionalmente, la CoS del paquete, en lugar de buscar el destino en las tablas de encaminamiento para cada paquete. Cuando el paquete sale del núcleo de la red, el encaminador frontera de salida utiliza la dirección destino del paquete para enviarlo a su destino final. Los paquetes siguientes del mismo flujo son automáticamente etiquetados de esta forma y de esta manera se gana velocidad.

Los encaminadores con conmutación por etiquetas (LSRs) construyen el camino que seguirán los paquetes a través del núcleo de la red (llamado *label switched path: LSP*). Todos los LSRs del núcleo de la red intercambian información de encaminamiento entre sí, mientras que los LSR frontera sólo intercambian información de encaminamiento con los encaminadores y con los LSR inmediatamente adyacentes.

Las etiquetas también pueden utilizarse para identificar al tráfico que debería recibir un trato especial en cuanto a la calidad. Utilizando técnicas de gestión de tráfico avanzadas para los caminos definidos por los encaminadores LSR, pueden garantizarse ciertos niveles de QoS en una red IP.

Un aspecto mas complejo de MPLS es la distribución y gestión de etiquetas entre los encaminadores MPLS de manera que todos coincidan en el significado de estas. El protocolo LDP (Label Distribution Protocol) está diseñado específicamente para este propósito.

Arquitecturas de QoS

Es improbable que los protocolos y técnicas vistas anteriormente se utilicen de manera independiente. De hecho, están diseñados para trabajar conjuntamente, de manera que se pueda asegurar una cierta QoS entre emisores y receptores.

IntServ/DiffServ: [35]

RSVP ofrece recursos para el tráfico de la red, mientras que *DiffServ* simplemente marca y prioriza. RSVP es más complejo y requiere más recursos en los encaminadores que *DiffServ*, de manera que afecta negativamente al núcleo de la red. Por ello se prefiere *DiffServ* en el núcleo.

DiffServ es un complemento perfecto a RSVP. Las aplicaciones pueden utilizar peticiones RSVP con elevada granularidad (como ancho de banda, retardo, fluctuación del retardo...). Los encaminadores de los bordes del núcleo que utilizan *DiffServ*, pueden mapear esas reservas RSVP a clases de tráfico indicándolo en el byte DS. Al salir del núcleo, se podría restaurar el servicio RSVP hasta el destino final.

MPLS y IntServ/DiffServ:

En coexistencia con RSVP, es posible la asignación de etiquetas MPLS en función de los flujos RSVP.

MPLS y *DiffServ* son parecidos en cuanto al soporte cualitativo de QoS, de manera que el mapeo del tráfico *DiffServ* a etiquetas MPLS es relativamente simple. Para soportar el modelo *DiffServ*, también se requiere por parte de MPLS un conjunto de recursos para cada una de las clases en cada encaminador LSR.

3.3. Modelo de QoS propuesto por la ETSI

Los esquemas de calidad en las redes TCP/IP se suelen aplicar en el núcleo de la red de transporte. Elementos como los terminales y, en muchos casos, las subredes de acceso, quedan fuera del amparo de los protocolos anteriores. Sin embargo, dichos elementos tienen una influencia clave en ciertos parámetros que afectan a la calidad como el eco, el retardo total, el efecto local o la degradación del sonido. Es por tanto necesaria la definición de un esquema completo que permita soportar la calidad de una conversación telefónica de forma global.

La ETSI, en su proyecto TIPHON, aborda un modelo global de telefonía IP (VoIP) basado en las normas propuestas por la ITU-T y el IETF. Su grupo de trabajo 5 (WG 5) especifica un modelo de referencia para el soporte de QoS en las redes de VoIP.

A continuación se describen los aspectos fundamentales de la arquitectura propuesta por la ETSI a tal efecto. Este modelo es el único que aborda una señalización propia para el soporte de calidad en VoIP actualmente, por lo que será utilizado como base para la señalización del modelo propuesto en el capítulo 4.

3.3.1. Introducción

La arquitectura general del esquema TIPHON para QoS se muestra en la figura 3.8. El control y señalización de QoS de extremo a extremo requerirá, generalmente, del intercambio de mensajes de QoS en cada uno de los planos.

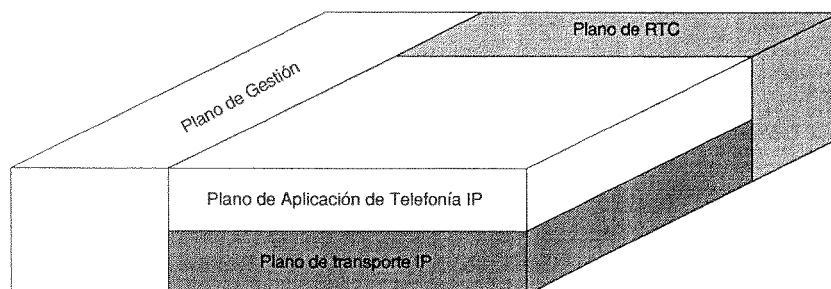


Figura 3.8: Esquema estructural de la arquitectura TIPHON

El modelo de referencia consta de plano de aplicación, plano de transporte, plano de gestión y plano de redes de telefonía tradicionales. Los niveles de QoS requeridos en la conversación se establecen dentro del plano de aplicación de telefonía IP mediante negociación entre el proveedor del servicio de telefonía IP (IPTSP) y los usuarios finales. En este plano se toman también decisiones específicas de la aplicación, que influyen en la QoS, como puede ser el tipo de códec usado, o el número de tramas por paquete. Dentro de este plano, los parámetros de QoS específicos de la aplicación son requeridos, autorizados, señalizados, controlados y computados.

El plano de transporte IP está formado por las entidades de red y transporte de la familia de protocolos TCP/IP. Debe proporcionar soporte de QoS al plano de aplicación. Ambos planos forman dos dominios de responsabilidad diferenciados; así como el plano de aplicación es responsabilidad de un proveedor de servicio, el plano de transporte es responsabilidad de un operador de red. En este plano se controlan aspectos de QoS no específicos de aplicación, como retardo máximo o probabilidad de pérdida de paquetes, y se computan y facturan dichos parámetros.

En el plano de gestión residirán las entidades comunes tanto al plano de aplicación como al de transporte IP. En él terminarán todos los flujos de gestión de QoS.

Dominios de servicio y transporte

En general, el despliegue de una infraestructura ajustada a las especificaciones de TIPHON no estará en su totalidad bajo el control de un único proveedor de servicio sino que existirán múltiples dominios, cada uno de los cuales representará el ámbito de control de un proveedor de servicio determinado. Cada uno de estos dominios estará normalmente restringido a la funcionalidad del plano de aplicación de telefonía IP, con elementos como Gatekeepers, agentes de llamada (*call agents*), etc.

De la misma forma, es posible concebir la existencia de múltiples dominios de transporte

controlados por diferentes operadores de red. Estos dominios estarán restringidos a funcionalidades del plano de transporte, como conmutación, cortafuegos, etc. Dichos dominios podrán diferir en aspectos como las políticas de QoS, los mecanismos de administración empleados, o los mecanismos de control de QoS (RSVP, *DiffServ*, MPLS...), de acceso, de cómputo, de direccionamiento (global, local) e incluso en los protocolos de transporte (IPv4, IPv6...).

Para poder ofrecer un servicio global será necesario definir entidades encargadas de hacer posible la interoperabilidad entre estos dominios, en forma de funciones de interconexión. El aspecto genérico de un despliegue TIPHON se muestra en la figura 3.9.

El control de QoS de extremo a extremo en un esquema como el anterior, formado por múltiples dominios, se puede conseguir de dos formas distintas:

- **Control mediante plano de aplicación:** En este primer caso, el encaminamiento de la llamada es controlado por los proveedores de servicio de aplicación. Por ello, cada dominio de transporte podría aplicar diferentes políticas de QoS, o utilizar diferentes mecanismos para su gestión.

En la figura 3.9 se contempla un caso general en el que múltiples dominios, tanto de servicio como de transporte, se ven involucrados en una llamada. La señalización de control de la llamada tiene lugar no sólo entre usuario final y proveedor de servicio, sino también entre diferentes proveedores. Del mismo modo, la señalización de QoS y los acuerdos de grado de servicio (SLA) también tienen lugar en estas mismas interfaces. Los flujos de datos, por el contrario, atraviesan únicamente el plano de transporte, estableciéndose entre usuario final y dominio de transporte, y entre diferentes dominios de transporte.

Entre cada proveedor de servicio, y el dominio de transporte asociado, es necesaria la existencia de SLA que aseguren que los requisitos de QoS son cubiertos por cada dominio de transporte involucrado en la llamada.

- **Control mediante plano de transporte:**

El control de QoS de la llamada es llevado a cabo ahora por el dominio de transporte local (en relación con el proveedor de servicio que atiende la llamada) y mediante acuerdos entre diferentes operadores de red. Es necesaria la existencia de SLA's tanto entre usuarios finales y proveedores de servicio, como entre los diversos operadores de red. Los usuarios finales, antes de establecer un flujo de datos a través del operador de red local, deben recibir autorización para realizar la llamada por parte de su proveedor de servicio de VoIP.

Este esquema, representado en la figura 3.10, sólo será una opción viable en el caso de que el plano de transporte comprenda un espacio de política homogéneo único, es decir, direccionamiento, acceso, mecanismos y políticas de QoS han de ser uniformes a través de los múltiples dominios de transporte.

Es posible la existencia de situaciones híbridas entre las dos anteriores, donde un dominio de servicio controle múltiples dominios de transporte, etc.

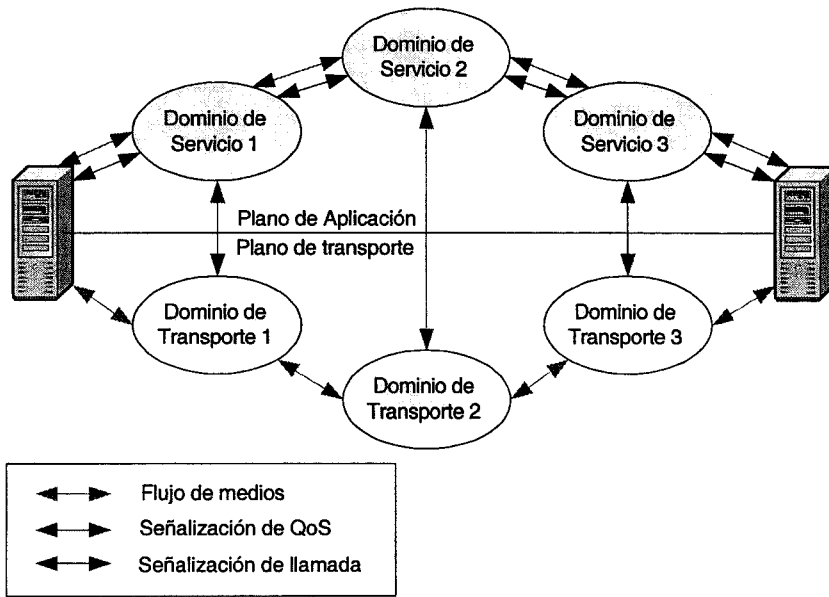


Figura 3.9: Arquitectura general TIPHON con control de QoS extremo a extremo por plano de servicio

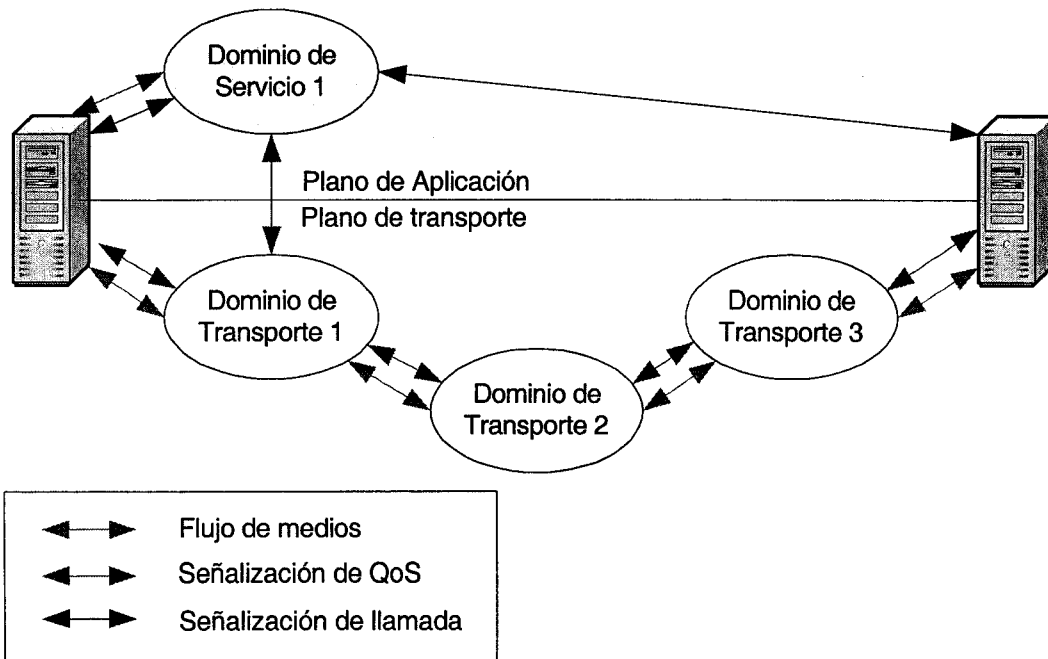


Figura 3.10: Arquitectura general TIPHON con control de QoS extremo a extremo por plano de transporte

3.3.2. Caracterización de la QoS

Parámetros de QoS de nivel de usuario, aplicación y transporte

En [36] se definen cuatro clases de QoS de extremo a extremo, expresadas en términos de la percepción de la calidad de la comunicación por parte de los usuarios finales. Dichas clases de QoS serán tomadas como base para la realización de acuerdos de QoS entre IPTSP's y usuarios finales. La especificación de una clase equivale a la solicitud de un determinado nivel de calidad (MOS) por parte de un usuario.

Dentro del plano de aplicación de telefonía IP, la clase de QoS percibida subjetivamente en el ámbito de usuario está determinada por un conjunto de parámetros de ingeniería, localizados tanto en el equipamiento de usuario como en los equipos de red y en el propio rendimiento de la red. Como ejemplo, se puede citar el tipo de códec, cualquier tipo de técnica de corrección de errores empleada, el algoritmo de empaquetado, los algoritmos ejecutados por el receptor para compensar la variación de retardo, técnicas de cancelación de errores empleadas en el decodificador, o retardo de procesamiento de los equipos.

En la práctica, una gran parte de estos parámetros estarán determinados por el diseño del equipamiento de usuario, y el control de la QoS extremo a extremo de una comunicación se verá reducido a la gestión de un cierto número de parámetros básicos de transporte, específicamente:

- Retardo máximo de extremo a extremo
- Máxima variación del retardo extremo a extremo
- Porcentaje de pérdida de paquetes.

Cuando en una llamada están involucrados múltiples dominios de transporte, estos parámetros han de ser especificados y controlados en todos ellos.

El grado de QoS alcanzable en los niveles de aplicación y usuario, dependerá en última instancia del rendimiento de las redes de transporte subyacentes que soportan el servicio de telefonía IP. Por tanto, es preciso el control en el plano de transporte de los tres parámetros anteriores.

La figura 3.11 describe los parámetros relevantes a cada nivel.

Interfaz con el Plano de Transporte

A pesar de que el nivel de QoS en la llamada es fijado en las aplicaciones de telefonía IP, su consecución depende del control del plano de transporte, con independencia de los mecanismos empleados en el plano de transporte para el control de la QoS. De este modo, ATM, RSVP, *DiffServ*, MPLS o incluso el dimensionamiento estático de la red son todos métodos válidos para controlar los parámetros de QoS requeridos, o incluso podría darse una mezcla de estos mecanismos en el caso de intervención de múltiples dominios de transporte. De cualquier forma, los

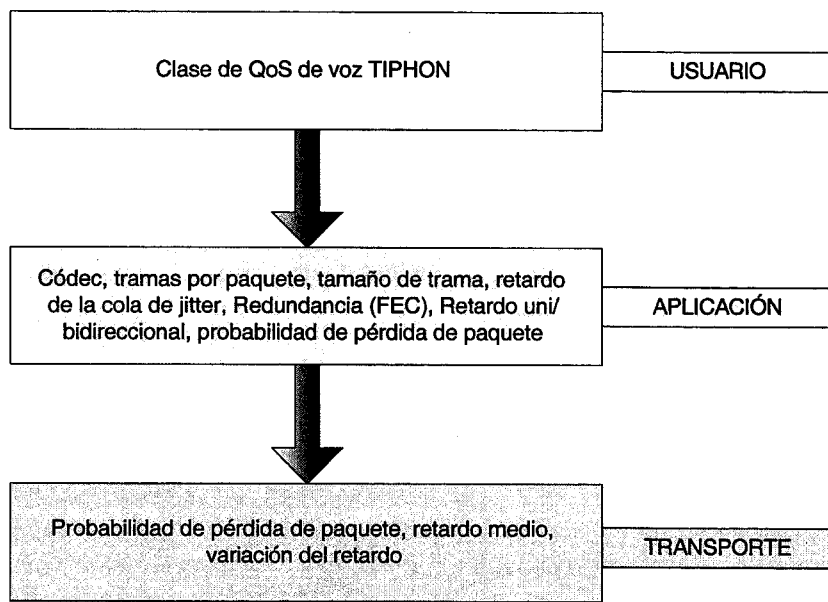


Figura 3.11: Parámetros de QoS de usuario, aplicación y transporte

requisitos de QoS para un cierto flujo de medios han de ser conocidos por el plano de transporte, de forma que sea posible controlar los parámetros de la red hasta alcanzar los niveles deseados.

En la conexión con el plano de transporte debe hacerse distinción entre parámetros de QoS de transporte, que especifican el nivel de QoS requerido, y descriptor de tráfico, que caracteriza al flujo de datos de forma suficiente para permitir a la red gestionar de forma óptica los recursos empleados por el flujo.

■ Parámetros de QoS de transporte

Los parámetros de QoS de transporte (máximo retardo de paquete, máxima variación de retardo, máxima probabilidad de pérdida) especifican de forma completa, en el nivel de transporte, los requisitos de QoS del flujo de transporte que acarrea la portadora.

La manera en que estos parámetros son especificados puede variar en función de diversas circunstancias. En general, existen tres posibilidades:

1. Especificación sobre el control de QoS en la portadora. En ciertas comunicaciones, como llamadas de voz *Best Effort* (lo mejor posible), no se requiere control de QoS.
2. Especificación de que alguno (o todos) de estos tres parámetros han de ser controlados, pero el valor para estos parámetros está especificado en algún otro lugar (por ejemplo, mediante acuerdos de servicio).
3. Especificación de los valores absolutos de los tres parámetros.

En [37] se puede encontrar una descripción más formal de los parámetros de caracterización de tráfico.

- **Descriptor de tráfico**

Este descriptor caracteriza las necesidades de recursos de un flujo de medios. La garantía relativa al nivel de QoS ofrecido a un flujo rige únicamente si el flujo permanece conforme a su descriptor.

Un descriptor de tráfico incluirá los siguientes parámetros:

- **Tamaño máximo de paquete,**

Este parámetro se define como el tamaño máximo de los paquetes de datos enviados, incluyendo la cabecera RTP o de cualquier otro protocolo de transporte de medios empleado.

- **Tasa de bit de pico,**

Este parámetro se define como la máxima tasa del flujo de medios a la cual el dominio de transporte está obligado a mantener las garantías de QoS. (la sobrecarga introducida por cabeceras de protocolos como RTP y similares debe estar contemplada en este valor).

Si el flujo tiene características de tasa constante de bit (CBR), estos parámetros son suficientes para habilitar de forma óptima la reserva de recursos de transporte. Sin embargo, en el caso de flujos con tasa de bit variables (VBR), se necesitará conocer otro tipo de parámetros, como la tasa media de bit o algún otro que caracterice sus ráfagas de transmisión. Continúa en estudio la posibilidad de ampliar la definición de este campo para permitir caracterizar el tráfico VBR, típico cuando se utilizan codecs con supresión de silencios.

3.3.3. Entidades funcionales e Interfaces normalizadas de QoS

Los mecanismos de QoS de TIPHON requieren la existencia de ciertas entidades, tanto en el plano de aplicación, como en el de transporte. Los elementos involucrados en la arquitectura de control QoS se describen a continuación:

Entidades Funcionales

- **Gestor de Calidad de servicio QoS (QoSM)**

Entidad funcional que media en peticiones de QoS controlada extremo a extremo, de acuerdo con políticas determinadas por la entidad de política de transporte (QoSPE). Se comunica con otros QoSM y con entidades de gestión de recursos de transporte (TRM) para determinar, establecer y controlar la QoS ofrecida o a ofrecer.

- **Entidad de política de QoS (QoSPE)**

Entidad funcional que gestiona las políticas de QoS en el dominio de telefonía IP, y proporciona autorización para niveles permitidos y por defecto de QoS. Recibe peticiones y envía respuestas a las entidades de gestión de QoS (QoSM) para determinar los niveles de QoS extremo a extremo autorizados.

- **Gestor de recursos de transporte (TRM)**

Entidad funcional que aplica mecanismos y políticas de QoS a los recursos de transporte existentes, con el objetivo de asegurar que estos recursos son reservados de tal forma que son suficientes para garantizar las características de QoS exigidas a lo largo del dominio de transporte bajo el control del TRM.

- **Entidad de política de transporte (TPE)**

Entidad funcional que mantiene las políticas de gestión de un dominio de transporte.

- **Función de interconexión (ICF)**

Entidad funcional que permite la conexión de un dominio de transporte, con elementos ajenos a él. Aplica las políticas adecuadas a los flujos que atraviesan el dominio de transporte, para asegurar que éstas son consistentes con las establecidas en el dominio local por el gestor relevante. Realiza la conversión de políticas para flujos que han de atravesar múltiples dominios de transporte.

- **Funcionalidad de transporte (TF)**

Es una entidad funcional que representa el conjunto de recursos de transporte pertenecientes a un dominio, que son capaces de controlar y gestionar objetivos de QoS.

La figura 3.12 muestra la ubicación de estas entidades en sus respectivos planos, así como las interrelaciones que se establecen entre ellas.

Primitivas y Puntos de referencia

Los puntos de referencia indicados en la figura 3.12 se derivan de un conjunto de puntos de referencia genéricos recogidos en la especificación [37], que en este caso son usados únicamente para señalización de QoS. Por este motivo la especificación [37] precede los nombres genéricos de estos puntos de referencia con la letra Q.

Para el control extremo a extremo de la QoS los flujos de información serán intercambiados entre los dominios de aplicación y transporte a lo largo del trayecto de establecimiento de llamada. En cada uno de los puntos de referencia definidos en la sección 3.3.3 se intercambiará información mediante primitivas de QoS.

En la figura 3.13, se ilustran algunas de las primitivas definidas, involucradas en el caso particular de un establecimiento de llamada en una única dirección, extremo a extremo. Para el caso bidireccional, se aplica un modelo análogo.

- **Punto de referencia QC1**

A través de este punto de referencia circulan los flujos de señalización establecidos entre la entidad QoSM del plano de aplicación de telefonía IP de un proveedor de servicio y su entidad par en el equipamiento de un usuario final. Esta señalización transporta la información de la portadora concerniente a la calidad de servicio requerida.

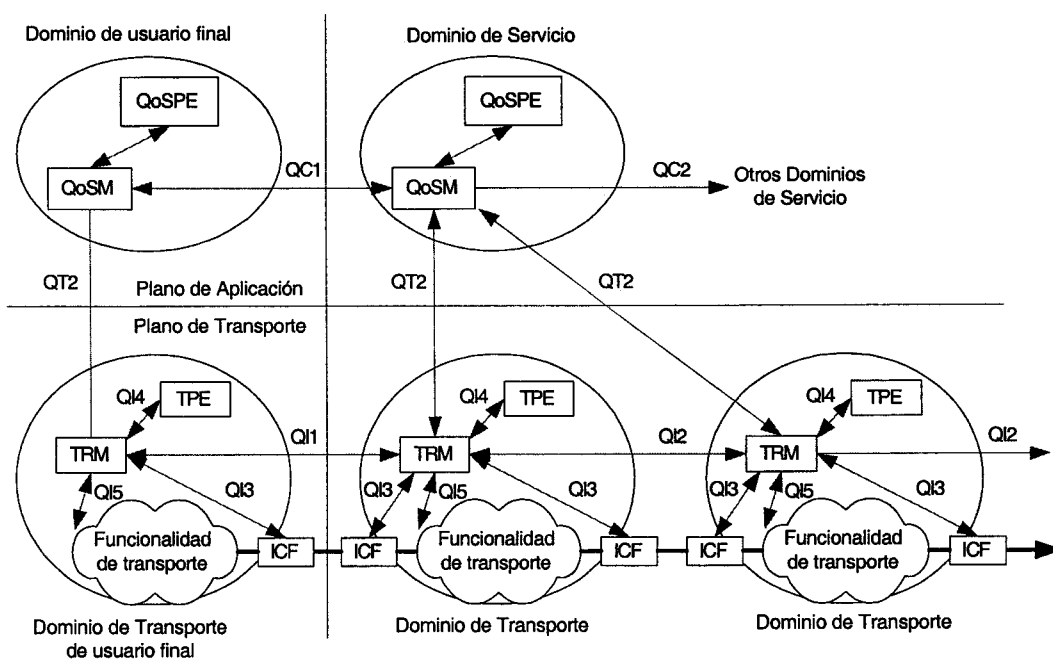


Figura 3.12: Arquitectura general TIPHON con control de QoS extremo a extremo por plano de transporte

■ Punto de referencia QC2

Los mensajes que atraviesan este punto son aquellos intercambiados entre entidades QoSM pertenecientes a distintos dominios dentro del plano de aplicación de telefonía IP. En ellos se transporta información de portadora concerniente a QoS.

Los mensajes intercambiados (en forma de primitivas) a través de esta interfaz comunican información de QoS entre diferentes dominios en el plano de aplicación de telefonía IP (QC2) o entre el equipamiento de usuario y su dominio de servicio (QC1). Para ambos puntos de referencia, se definen las siguientes primitivas:

QoSM request (QC1/2. QoSMreq) solicita el establecimiento de una portadora conforme a una determinada Clase de Servicio TIPHON, o con unas ciertas garantías de QoS.

QoSM confirm (QC1/2. QoSMconf) confirma la creación de una portadora conforme a la Clase de Servicio o características de QoS solicitadas.

QoSM reject (QC1/2. QoSMrej) rechaza el establecimiento de una portadora conforme a una Clase de Servicio o a ciertas características de QoS.

QoSM release request (QC1/2. QoSMrelreq) solicita la liberación de una portadora.

QoSM release confirm (QC1/2. QoSMrelconf) confirma la liberación de una portadora.

■ Punto de referencia QS4

Localizado entre una entidad QoSM y la entidad QoSPE asociada. Los mensajes intercambiados a través de esta interfaz comunican información relativa a la política de QoS a aplicar al establecimiento de una portadora con unos niveles de QoS especificados. Están definidas las siguientes primitivas:

QoSPE request (QS4. QoSPEreq) solicita autorización para el establecimiento de una portadora con ciertas características de QoS.

QoSPE confirm (QS4. QoSPEconf) autoriza el establecimiento de una portadora con ciertas características de QoS

QoSPE reject (QS4. QoSPErej) rechaza el establecimiento de una portadora con características de QoS definidas.

■ Punto de referencia QT2

Localizado entre QoSM y su(s) TRM(s) asociado(s), por este punto circulan mensajes que comunican información de QoS relativa a flujos de transporte, entre un dominio de servicio y el dominio de transporte asociado. Dicha información especifica datos como el nivel de QoS, direccionamiento, y caracterización de los flujos de medios soportados. Las primitivas definidas son:

TRM QoS request (QT2. TRMQreq) solicita el establecimiento de un flujo de transporte con características de QoS definidas a través de un dominio de transporte, o bien la reserva en general de recursos de este dominio.

TRM QoS confirm (QT2. TRMQconf) confirma la creación del flujo de transporte o reserva de recurso solicitada.

TRM QoS reject (QT2. TRMQrej) rechaza la creación del flujo de transporte o reserva de recurso solicitada.

TRM QoS release request (QT2. TRMQrelreq) solicita la liberación de un flujo de transporte o reserva de recurso.

TRM QoS release confirm (QT2. TRMQrelconf) confirma la liberación de un flujo de transporte o reserva de recurso.

TRM QoS performance notificacion (QT2. TRMQperfnofif) notifica al dominio de servicio el rendimiento del dominio de transporte, en cuanto al cumplimiento de los niveles de QoS solicitados. Esta primitiva podría ser utilizada para una notificación urgente. Su definición y uso permanecen en estudio.

■ Punto de referencia QT1

Los mensajes intercambiados a través de esta interfaz comunican información de QoS relativa a flujos de transporte, transmitida entre el gestor de QoS (QoSM) integrado en el equipamiento de usuario, y el dominio de transporte asociado.

Esta información en concreto describe los requisitos de QoS exigidos a los flujos de transporte que soportarán la transmisión de medios, aporta información de direccionamiento referente a los mismos, y caracteriza los flujos de medios transportados.

Las primitivas definidas son las siguientes:

QoS request (QT1. TRMQreq) solicita el establecimiento de un flujo de transporte con características de QoS definidas a través de un dominio de transporte, o bien la reserva en general de recursos de este dominio.

QoS confirm (QT1. TRMQconf) confirma la creación del flujo de transporte o reserva de recurso solicitada.

QoS reject (QT1. TRMQrej) rechaza la creación del flujo de transporte o reserva de recurso solicitada.

QoS release request (QT1. TRMQrelreq) solicita la liberación de un flujo de transporte o reserva de recurso.

QoS release confirm (QT1. TRMQrelconf) confirma la liberación de un flujo de transporte o reserva de recurso.

■ **Punto de referencia QI1**

Localizado entre la entidad TRM del dominio de transporte de usuario y su entidad par en el dominio de transporte local. El flujo de información a través de este punto especifica las características de QoS exigidas a los flujos de transporte del bucle local que soportarán el flujo de medios, sus propiedades, e información de direccionamiento de los flujos de transporte.

■ **Punto de referencia QI2**

Los mensajes intercambiados a través de esta interfaz comunican información de QoS relativa a flujos de transporte, transmitida entre una entidad TRM en el equipamiento de usuario y su entidad par en el dominio de transporte (QI1), o bien entre dos entidades TRM de dominios diferentes (QI2).

Esta información se refiere de nuevo a los requisitos de QoS exigidos a los flujos de transporte que soportarán la transmisión de medios, la información de direccionamiento referente a los mismos, y la caracterización de los flujos de medios transportados.

Para los puntos QI1 y QI2, se encuentran definidas las siguientes primitivas:

QoS request (QI1/2. TRMQreq) solicita el establecimiento de un flujo de transporte con características de QoS definidas a través de un dominio de transporte, o bien la reserva en general de recursos de este dominio.

QoS confirm (QI1/2. TRMQconf) confirma la creación del flujo de transporte o reserva de recurso solicitada.

QoS reject (QI1/2. TRMQrej) rechaza la creación del flujo de transporte o reserva de recurso solicitada.

QoS release request (QI12/2. TRMQrelreq) solicita la liberación de un flujo de transporte o reserva de recurso.

QoS release confirm (QI1/2. TRMQrelconf) confirma la liberación de un flujo de transporte o reserva de recurso.

■ Punto de referencia QI3

Localizado entre TRM e ICF, los mensajes que cruzan este punto controlan la función de interconexión, permitiendo que realice sus funciones de interoperabilidad de redes y administración de políticas.

Los mensajes intercambiados a través de esta interfaz comunican información de QoS relativa a flujos de transporte, transmitida entre una entidad TRM y una entidad ICF dentro de un dominio de transporte. Esta información indicará las características de QoS requeridas a los flujos de transporte que portan los medios, las propiedades de este último, información de direccionamiento de transporte, y datos relativos a los mecanismos de QoS empleados.

Están definidas las siguientes primitivas:

QoS request (QI3. ICFQreq) solicita el establecimiento de un flujo de transporte con unas características y mecanismos de QoS definidas, entrante o saliente al dominio de transporte.

QoS confirm (QI3. ICFQconf) confirma la creación del flujo de transporte solicitado.

QoS reject (QI3. ICFQrej) rechaza la creación del flujo de transporte solicitado.

QoS release request (QI3. ICFQrelreq) solicita la liberación de un flujo de transporte.

QoS release confirm (QI3. ICFQrelconf) confirma la liberación de un flujo de transporte.

■ Punto de referencia QI4

Localizado entre TRM y TPE. En la versión de la norma vigente durante la realización de la presente Tesis, aún permanece en estudio.

Los mensajes intercambiados a través de esta interfaz contienen solicitudes y admisiones para el establecimiento de flujos de transporte con QoS garantizada a lo largo de un dominio de transporte, así como información de política específica a los mecanismos de QoS involucrados. Están definidas las primitivas:

QoS policy request (QI4. PQreq) solicita autorización para el establecimiento de un flujo de transporte con características definidas de QoS, y se contempla la posibilidad de que el TRM solicite la especificación de un mecanismo de QoS y la dirección de una entidad ICF.

QoS policy confirm (QI4. PQconf) proporciona la autorización necesaria para el establecimiento del flujo de transporte, y define el mecanismo de QoS y la dirección de la entidad ICF requerida.

QoS policy reject (QI4. PQrej) rechaza la creación del flujo de transporte solicitado.

■ Punto de referencia QI5

Localizado entre una entidad TRM y el conjunto de recursos que forman la funcionalidad de transporte, en la versión de la norma vigente a la hora de efectuar el presente trabajo, este punto permanece en estudio.

Los mensajes intercambiados a través de esta interfaz comunican información de QoS relativa a flujos de transporte, transmitida entre una entidad TRM y la Funcionalidad de Transporte gestionada por él. Esta información indicará las características de QoS requeridas al flujo, y los datos relacionados con el mecanismo de QoS necesarios para conseguir estos objetivos. Las primitivas definidas son:

QoS request (QI5. TFQreq) solicita el establecimiento de un flujo de transporte mediante un mecanismo y con unas características de QoS definidas.

QoS confirm (QI5. TFQconf) admite la creación del flujo de transporte solicitado.

QoS reject (QI5. TFQrej) rechaza la creación del flujo de transporte solicitado.

QoS release request (QI5. TFQrelreq) solicita la liberación de un flujo de transporte.

QoS release confirm (QI5. TFQrelconf) confirma la liberación de un flujo de transporte.

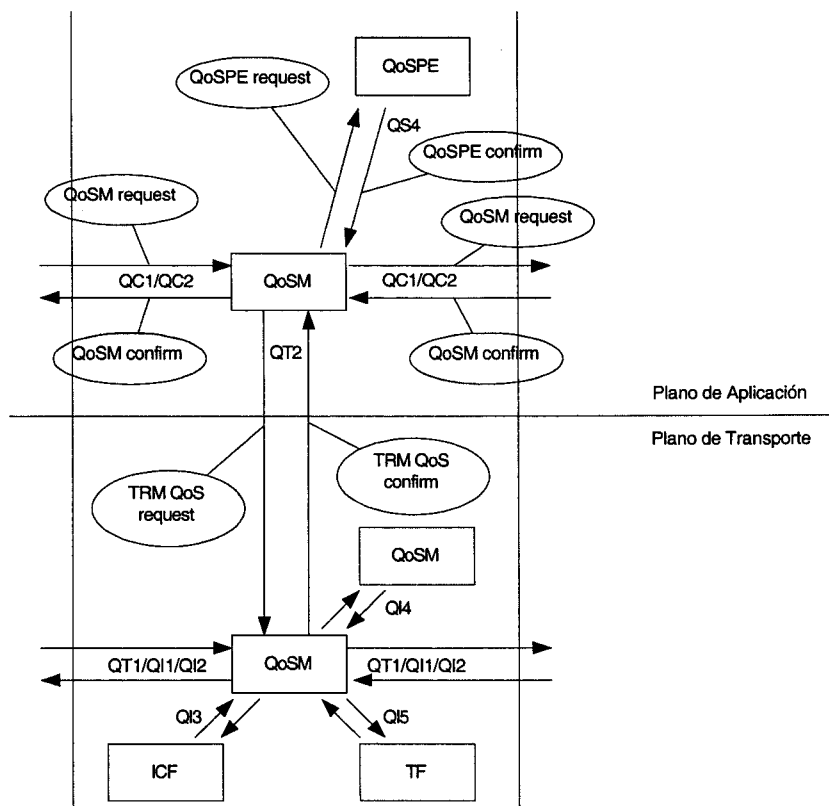


Figura 3.13: Modelo de flujos de información de QoS

Grupos de Parámetros de QoS

Cada uno de las primitivas definidas en cada interfaz, contiene uno o varios elementos de información, conocidos como *grupos de parámetros*. Algunos de estos grupos tienen carácter

obligatorio, y otros opcional. Una descripción de los más importantes se muestra en el cuadro 3.1.

Grupo	Descripción	Parámetros
Clase de Servicio QoS	Describe la clase QoS extremo a extremo TIPHON de la portadora	<i>Best, High, Medium o Best Effort</i>
Tipo de Códec y Empaquetado	Describe el tipo de códec empleado en una portadora y el modo en que los datos son distribuidos en paquetes	<i>Tipo de Códec (Opcionalmente, una lista con todos los tipos de códecs admisibles), Tramas por paquete (Opcionalmente, una lista)</i>
Parámetros de QoS de Transporte	Especifica las características de QoS requeridas al flujo de transporte que soportará la conexión de medios	<i>Retardo Máximo, Máxima Variación del Retardo, Máxima Pérdida de Paquetes</i>
Descriptor de Tráfico	Caracteriza los requisitos de una flujo de datos de aplicación (excluyendo los requisitos del flujo de transporte portador)	<i>Tasa de Pico, Tamaño Máximo de Paquete</i>
Direcciones de Transporte	Especifica información de direccionamiento suficiente para definir el flujo de transporte correspondiente a una portadora de medios	<i>Dirección de Transporte Origen, Dirección de Transporte Destino</i>
Mecanismo de QoS	Describe el mecanismo empleado en el plano de transporte	<i>Tipo: Ninguno, RSVP, diff-Serv o MPLS</i>

Cuadro 3.1: Grupos de parámetros QoS

3.3.4. Reserva de QoS a través de dominios de servicio y transporte

El modelo genérico representado en la figura 3.9 asume que el sistema TIPHON está dividido en una serie de dominios de servicio y transporte, de tal forma que una llamada atravesará en general un cierto número de esos dominios.

Se pueden considerar los requisitos de QoS extremo a extremo como presupuestos de calidad extremo a extremo relacionados con los flujos de medios. Para conseguir la QoS deseada, estos presupuestos han de ser distribuidos entre los dominios, incluido el equipamiento de usuario, como se ilustra en la figura 3.14. Los parámetros de QoS de transporte especifican los presupuestos asignados a cada dominio de transporte.

Asumiendo que el rendimiento de cada dominio es estadísticamente independiente del resto de dominios, la acumulación global de los parámetros de QoS de transporte puede calcularse como sigue:

- El retardo es aditivo, es decir, $D_{tot} = D_1 + D_2 + \dots D_n$

- La probabilidad de pérdida se acumula de forma probabilística, es decir,

$$P_{tot} = 1 - [(1 - P_1)(1 - P_2) \dots (1 - P_n)]$$

- La variación del retardo se acumula de manera cuadrática media, es decir:

$$V_{tot} = \sqrt{V_1^2 + V_2^2 + \dots + V_n^2}$$

Donde D_n es el retardo medio en un sólo sentido, P_n es la probabilidad de pérdida de paquete, y V_n es la desviación estándar de la variación del retardo, todas relativas al dominio n .

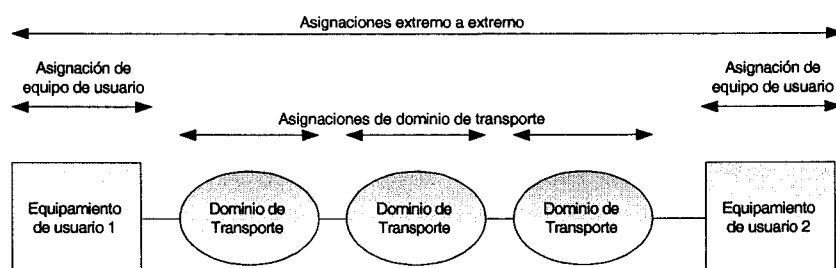


Figura 3.14: Reserva de los presupuestos de QoS TIPHON

La norma [37] describe tres formas de distribuir la responsabilidad de soportar el nivel de calidad a proporcionar entre los dominios involucrados:

- Señalización dinámica de los parámetros de QoS de transporte extremo a extremo entre IPTSPs por cada llamada.
- SLA's estáticos entre IPTSPs que definen el valor de los parámetros de QoS de transporte aplicados a llamadas que atraviesen su dominio de control. Este método requiere además de disposiciones en los SLA's especificando el máximo número de dominios extremo a extremo que pueden verse involucrados en la llamada.
- Agregación de recursos de dominio de transporte y mantenimiento de información de su disponibilidad, de forma que una señalización dinámica de los parámetros de QoS de transporte permita conocer qué valores de estos parámetros pueden ser proporcionados en cualquier punto del tiempo.

Los acuerdos (SLA) estáticos se realizan por procedimientos administrativos, lo que cae fuera del alcance del presente estudio. A continuación se detallan los restantes métodos ¹.

¹La distribución del grado de calidad entre red y equipamiento de usuario debe ser establecida previa a la llamada bien mediante SLA's, bien mediante señalización dinámica, o con una combinación de ambos métodos.

Señalización dinámica de los parámetros de QoS de transporte

La negociación dinámica de estos parámetros permitirá a los IPTSP's la señalización de los requisitos de QoS de forma individual para cada llamada.

La señalización de control de llamada tiene lugar en el plano de aplicación entre diferentes IPTSPs, y entre estos últimos y los usuarios finales. De forma similar, la señalización de QoS tiene lugar entre las mismas partes, y sigue el camino de establecimiento de la llamada. Entre cada IPTSP involucrado en la llamada y su dominio de transporte asociado, tiene lugar la señalización de los parámetros de QoS de transporte, posiblemente mediante SLA's, para asegurar que los requisitos sobre estos parámetros son satisfechos por el dominio. Alguno de estos parámetros pueden estar recogidos en SLA's, otros se negociarán dinámicamente.

Especificación de los parámetros de QoS de transporte en Acuerdos de Nivel de Servicio

Los SLAs son parte de los acuerdos de colaboración entre IPTSPs. Éstos pueden especificar los parámetros de QoS de transporte para las llamadas que atraviesan los dominios de transporte relacionados, obviando así la necesidad de señalar estos parámetros de manera individual por cada llamada.

En el caso de que los parámetros extremo a extremo se especifiquen de esta forma, la necesidad de señalización de QoS puede reducirse a la disponibilidad de recursos para ofrecer la clase de servicio acordada.

Agregación

Con la reserva agregada o en bloque de recursos de transporte, el plano de aplicación de telefonía IP reserva una cantidad de recursos de transporte que es suficiente para acarrear un cierto número de flujos de medios. De esta manera, el plano de aplicación puede utilizar de una forma más eficiente los recursos reservados, usando multiplexión estadística de flujos VBR. La agregación de recursos también supone un ahorro en cuanto a señalización, puesto que no es necesario contactar con el plano de transporte para cada llamada.

La agregación de recursos puede ser llevada a cabo de la misma forma en el dominio de transporte, bajo control de los operadores de red o de los propios IPTSPs. Una cierta cantidad de recursos de transporte con QoS garantizada son reservados con anterioridad al establecimiento de un flujo; flujos individuales establecidos posteriormente consumirán una porción de estos recursos, evitando la necesidad de controlar en la red parámetros de QoS de forma individual para cada flujo. En función de quién controle el proceso de agregación se puede distinguir:

- *Agregación bajo control del TRM.*

En este caso la agregación es llevada a cabo por los operadores de red. El papel del TRM en este caso es el de monitorizar la reserva de estos recursos, empleando mecanismos como MPLS o *IntServ*. El plano de aplicación de telefonía IP no es consciente de la agregación que se está llevando a cabo en el plano de transporte, y por lo tanto es algo que queda fuera del ámbito de [37].

- *Agregación bajo control del QoSM.*

Cuando el control de los recursos agregados está bajo control del IPTSP, una entidad de gestión de QoS (QoSM) reservará una cantidad determinada de recursos de transporte, con características de QoS garantizadas, previamente al establecimiento de ningún flujo de medios. Esto se llevaría a cabo mediante acuerdos con el operador de red. A partir de ahí, el QoSM controlaría la disponibilidad de recursos de QoS reservados calculando la utilización actual del agregado.

La agregación de recursos requiere la capacidad de gestionar el ancho de banda del agregado, y la funcionalidad de control de admisión de conexiones.

Un recurso agregado tiene una dirección IP origen y una dirección IP destino dentro de cada dominio de transporte. Múltiples portadoras serían multiplexadas, basándose por ejemplo en número de puerto. A la hora de reservar un agregado, el plano de aplicación de telefonía IP enviaría al dominio de transporte, por ejemplo, un rango de número de puertos. Es deseable que todos estos puertos sean admitidos por el ICF, de manera que no sea necesario señalización entre los planos de aplicación y transporte por cada portadora del agregado que pase a acarrear un flujo de medios.

Para la creación de recursos agregados, existen dos posibilidades:

- Preasignación (provisión de recursos en forma semipermanente).
- Establecimiento y liberación dinámica de agregados. Opcionalmente, puede existir negociación del ancho de banda de estos agregados.

El control de admisión de conexiones en el recurso agregado ha de ser llevado a cabo de forma individual para cada flujo. La información de uso de recursos será utilizada para este control de admisión. El uso actual del agregado puede calcularse de varias formas, por ejemplo:

- Cálculos de ocupación basándose en el valor de los descriptores de tráfico de los flujos ya admitidos.
- Medidas de uso del recurso.

La renegociación de la reserva de recursos debe poder tener lugar en cualquier instante. Por ejemplo, si se excede un cierto porcentaje de utilización de los recursos reservados debe ser posible ampliar la reserva o crear un nuevo agregado. De forma similar puede ser deseable reducir los recursos reservados si la demanda de tráfico decrece.

3.3.5. Procedimientos de señalización de QoS

Las primitivas definidas en la sección 3.3.3 pueden ser utilizadas en un cierto número de procedimientos de señalización de QoS. A modo de anexo informativo, la especificación [37] incluye un conjunto de procedimientos de establecimiento de portadora con QoS controlada. A continuación se describe uno de ellos, que será posteriormente utilizado en el nuevo modelo propuesto en el capítulo 4.

Establecimiento mediante intermediario de una portadora con QoS controlada

En esta situación una tercera entidad, típicamente un proveedor de servicio, es el encargado de establecer una portadora con QoS controlada en representación del llamante, transmitiendo al plano de transporte las características de QoS solicitadas.

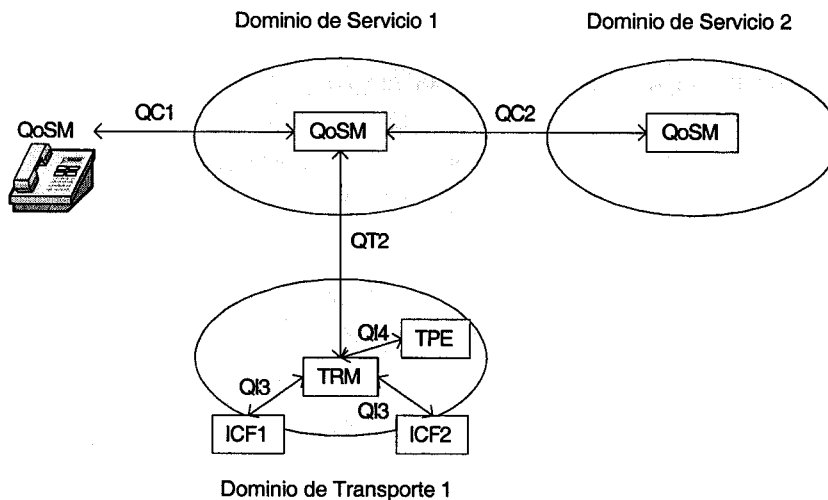


Figura 3.15: Elementos funcionales involucrados en un establecimiento de llamada mediante intermediario

Los elementos funcionales implicados en la llamada aparecen en la figura 3.15. El gestor de QoS localizado en el terminal solicita al QoSM del dominio 1 la prestación de un servicio. Este QoSM establece una portadora con QoS controlada a través de señalización con el TRM del dominio de transporte 1, y con el QoSM del dominio de servicio 2. Los flujos de información que circulan en este escenario se detallan en la figura 3.16

3.4. Conclusiones

En el presente capítulo se han introducido los diferentes esquemas de señalización existentes para el soporte de VoIP, esto es, H.323 y SIP. Ambos modelos delegan el tratamiento de la calidad en las llamadas a los diferentes mecanismos de QoS existentes en la red IP, presentados en la sección 3.2.1.

Según dicho enfoque, los terminales de usuario deben ser capaces de soportar la señalización del protocolo de QoS utilizado en la red de acceso; normalmente RSVP. Esto provoca una mayor complejidad en el terminal, lo que se traduce en mayores retardos, mayor coste e incremento del tiempo de establecimiento de la llamada. Además, la calidad ofrecida por la red de transporte es una parte de los factores que afectan a la calidad total, ya que los terminales tienen gran influencia en el retardo, eco o el factor de equipo (*Ie*). Así, es posible obtener una calidad

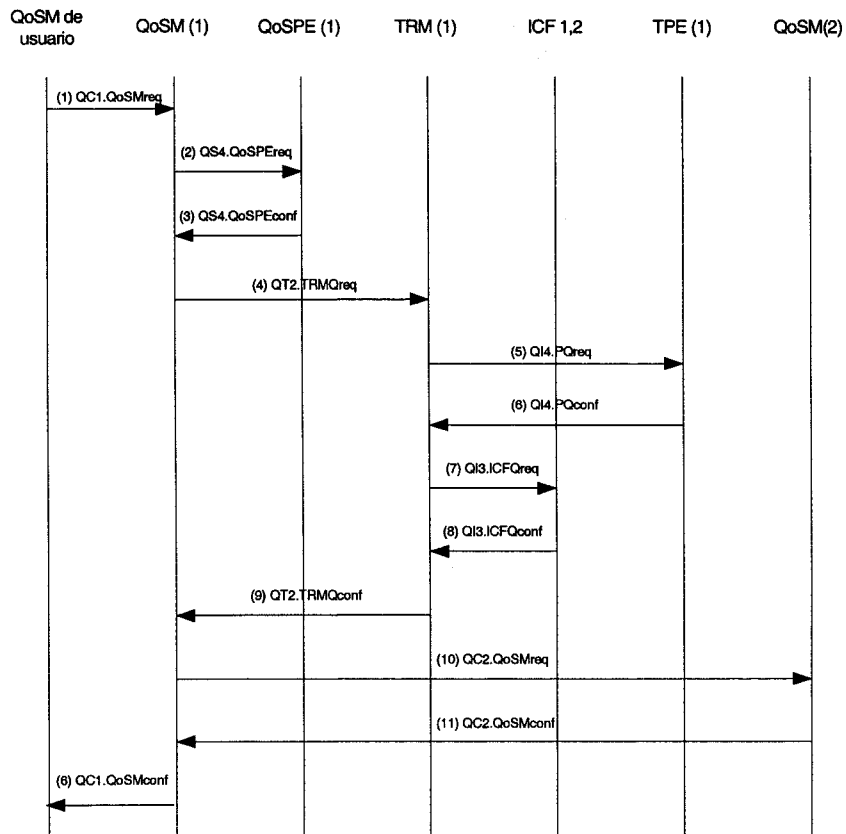


Figura 3.16: Flujos de información en un establecimiento de portadora mediante intermediario

insatisfactoria en la conversación a pesar de tener gran calidad de la red de transporte (lo que se traduce en desperdicio de recursos).

La necesidad de una entidad capaz de traducir el nivel de QoS deseado para una conversación a parámetros de red y terminales, ha sido atendida en el esquema de calidad propuesto por la ETSI en el marco de TIPHON. En dicho modelo se propone el uso de una señalización de calidad propia para las redes de VoIP, que permita a usuarios, proveedores de servicio de VoIP y proveedores de recursos de transporte, negociar dinámicamente la calidad de la conversación, así como los parámetros de funcionamiento en redes y terminales.

La aplicación de este modelo exige la toma de una serie de decisiones relativas a implementación no especificadas en la norma, que pueden afectar al rendimiento del mismo. En concreto, se deja libertad en la implementación de los siguientes aspectos:

- La estructura completa de las primitivas de cada interfaz, pues solo se especifican los elementos de información que deben contener.
- La ubicación física de las entidades funcionales.

A su vez, la versión de dicha norma vigente en el momento de realizar este trabajo, se encuentra incompleta en los siguientes puntos:

- Definiciones ASN.1 de los elementos de información a utilizar. Tan sólo se ofrece una descripción de los mismos.
- Elementos de información de algunas primitivas de servicio (se deja como trabajo futuro).

En el modelo de TIPHON se definen tan sólo 3 niveles de calidad seleccionables por el usuario. Este hecho resta flexibilidad a una de las principales ventajas de VoIP: la selección del nivel de QoS por parte del usuario. La especificación más adecuada para definir el nivel de calidad sería en forma de MOS, presentada en el capítulo 2 de esta memoria.

Por otra parte, el esquema no tiene en cuenta el coste de los recursos de transporte. Sin embargo, debido a la influencia de los terminales en la calidad total, es posible minimizar la utilización de los recursos de transporte para conseguir un determinado nivel de QoS, especificando los parámetros relativos a los terminales.

El modelo propuesto por la ETSI servirá como punto de partida para el desarrollo del nuevo modelo que define el capítulo 4 de la presente memoria. No obstante, para su uso es requisito indispensable la toma de una serie de decisiones relativas a implementación, así como completar el modelo en aquellos puntos en los que aún se encuentra inconcluso. En los Apéndices A y B, así como en el capítulo 4 se puede encontrar las modificaciones y decisiones de diseño tomadas en la implementación realizada.

Parte II

Contribuciones

Capítulo 4

Modelo para el soporte de QoS con mínimo coste

En el capítulo previo se ha señalado la influencia de los terminales sobre la calidad de la conversación para las redes de VoIP. Éstos determinan factores como el eco, parte del retardo o la distorsión. El resto de los factores que influyen en la calidad, principalmente el retardo y la pérdida de paquetes, se sitúan en la red. En redes TCP/IP sin soporte de QoS no es posible mantener una calidad aceptable siempre, ya que se depende del tráfico existente en la red.

En redes TCP/IP con soporte QoS, la influencia de la red se encuentra acotada y bajo control, por lo que normalmente se podrá asegurar a los usuarios el nivel de QoS que deseen. Para traducir el nivel de QoS deseado (en forma de MOS) a parámetros de funcionamiento en los terminales (codec y empaquetado) y en la red de transporte (retardo y tasa de pérdida de paquetes) es necesario emplear una herramienta de medida de calidad basada en parámetros. El Modelo-E, descrito en el capítulo 2 se adecua perfectamente a este cometido. Así, el nivel de calidad final que tendrá una llamada se puede descomponer como sigue:

$$MOS = f(R) = f(\text{Codec}, \text{Empaquetado}, \text{Retardo}, \text{Pérdida Paquetes}, \text{Variación Retardo})$$

Encontrar la configuración de los parámetros de los terminales que minimiza el consumo de recursos en la red de transporte será el objetivo del algoritmo de optimización que se describe en la sección 4.6. La figura 4.1 muestra el esquema general de funcionamiento del algoritmo de optimización, que constituye el núcleo del modelo propuesto.

Este nuevo modelo permite a los proveedores de VoIP ofrecer el nivel de QoS seleccionado por los usuarios, minimizando el consumo de los recursos de transporte a través de la selección adecuada de los parámetros en los terminales. En los próximos apartados se detallarán los aspectos esenciales que describen dicho modelo. A fin de facilitar la lectura y comprensión del modelo en esta memoria, se han separado en Apéndices todas las cuestiones relativas a la implementación del mismo.

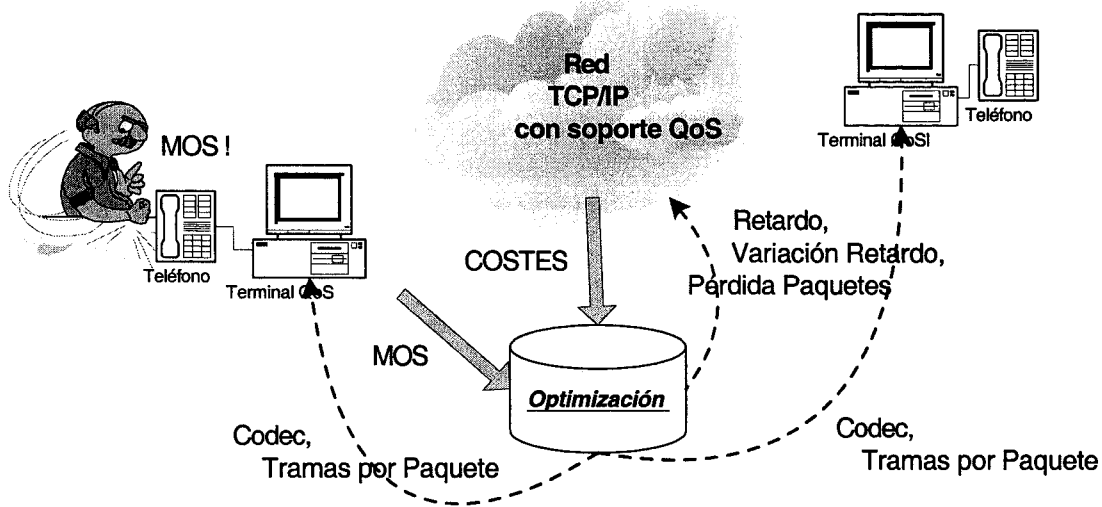


Figura 4.1: Funcionamiento de la optimización en el modelo

4.1. Requisitos del modelo

Para seleccionar los parámetros de calidad en la red IP de forma óptima, es necesario tener conocimiento de los parámetros relativos a los terminales de los clientes, ya que ambos afectan a la calidad que ha sido solicitada para la conversación.

Se requieren pues dos partes bien diferenciadas para dar soporte al esquema que se busca:

- Soporte de señalización: necesaria para llevar los parámetros y primitivas de servicio correspondientes al plano de calidad.
- Optimización los recursos exigidos a la red de transporte en base a los parámetros que intervienen en la calidad de las conversaciones.

Para soportar el esquema de señalización se tomará como partida el modelo propuesto por la ETSI en [37], descrito en el capítulo 3. *Debido a que la versión vigente de dicha norma, en el momento de realizar el presente trabajo, se encuentra aún incompleta en algunos aspectos, es requisito indispensable completarla en los siguientes puntos:*

- especificación formal de parámetros de calidad.
- completar los elementos de información de algunas primitivas de servicio.

Así mismo, la norma deja libertad de implementación en aspectos como:

- definición completa de las primitivas de calidad.
- ubicación física de las entidades funcionales.

En el Apéndice A de la presente memoria se puede encontrar la definición en ASN.1 de parámetros y primitivas del modelo utilizado.

Las necesidades del nuevo modelo en la relación entre las diversas interfaces son:

1. Relación entre los planos de transporte y aplicación

- El servicio de VoIP exige la utilización de un servicio de transporte IP. Ambos servicios pueden ser proporcionados por el mismo o por distinto proveedor. Como caso más general, se asume que ambos servicios son proporcionados por diferentes proveedores
- El proveedor de servicio de transporte IP deberá proporcionar una función de coste al proveedor de servicio de voz sobre IP. Es por tanto responsabilidad del proveedor de servicio de transporte, el dimensionado de su red para dar soporte de calidad a los distintos flujos de voz. La definición de dichas funciones de coste, así como la especificación de los parámetros de calidad y de caracterización de flujo se detallan en la sección 4.6.

2. Restricciones del plano de transporte

- El dominio de transporte será único. Esto supone una simplificación del problema real, en el que pueden existir diversos dominios de transporte con políticas de calidad de servicio que pueden o no diferir. La ampliación a varios dominios de transporte añade complejidad a la señalización, sin aportar nada nuevo al modelo en sí (mismo algoritmo de optimización).
- La estructura del dominio de aplicación deberá ajustarse a la especificada en [37].
- Caen fuera del alcance del presente trabajo el funcionamiento interno de la entidad funcional del transporte (TF), así como los mecanismos utilizados por esta para asegurar a un cierto flujo el nivel de calidad solicitado.

3. Restricciones del plano de aplicación

- El dominio de servicio será único. Esto es una simplificación del caso general. En el caso de varios dominios, deben existir acuerdos (SLA) entre los proveedores del servicio de telefonía IP que permitan un control de la calidad extremo a extremo. La extensión a varios dominios viene especificada en la norma de la ETSI anteriormente mencionada, por lo que su implementación no aporta nada al nuevo modelo.
- El nivel de calidad deseado por los clientes se especificará en forma de MOS (Mean Opinion Score), ya que es la forma natural de expresar el nivel de calidad en conversaciones telefónicas. Para una mayor precisión en el grado de calidad solicitado, se utilizará un número entero (entre 1 y 5) y un decimal, así como un margen de variación en torno a esa cantidad.

- Si el Gestor de Calidad de Servicio comprueba que no se puede ofrecer la calidad solicitada, se abortará el proceso de llamada. El motivo de ello radica en el paralelismo que debe existir entre el funcionamiento del servicio de VoIP y el ofrecido por las redes telefónicas tradicionales.
- La tarificación de la calidad a los usuarios queda fuera del presente trabajo, aunque debido a la estructura del esquema de soporte de calidad, la tarificación sería de fácil implementación.
- El modelo de control de calidad extremo a extremo estará gobernado por el plano de aplicación, por lo que el encaminamiento de la señalización de llamada y calidad de servicio se llevará a cabo únicamente por este plano.
- Sólo se permite la realización de llamadas a clientes VoIP, sin salida hacia otro tipo de redes. Dicha simplificación se basa en que la calidad en las redes RTC es fija, por lo que el algoritmo de optimización seguirá manteniendo su validez.

4. Requisitos de los clientes (QoSM del dominio de usuario)

- Siguen la Recomendación H.323 versión 3 y H.225.0 versión 4, pues la señalización de estas normas será utilizada para integrarla con la señalización de calidad.
- Deben estar registrados en el Gestor de Calidad de Servicio, ya que es la única forma posible para garantizar que el terminal llamado soporta el esquema de QoS propuesto. Además, al registrarse deberán especificar sus características de retardo en función del codec, y de retardo medio en la subred de acceso. Dichos retardos no varían de una llamada a otra, por lo que se pueden especificar de forma estática en el momento del registro. Así mismo los clientes especificarán un retardo general para cubrir el caso de no disponer de datos de retardo anteriores. En el Apéndice C se proporciona un método para conseguir los retardos debidos al cliente en función del codec.
- Los clientes establecerán los codecs, número de tramas por paquetes y tamaño de la cola para compensar la variación del retardo que le sean indicados por el Gestor de calidad de servicio del plano de aplicación (QoSM).
- Los clientes podrán especificar los codecs disponibles de forma dinámica en cada una de las llamadas, lo que permite adaptarse a las condiciones cambiantes del tráfico en la red.
- Los clientes H.323 pedirán siempre autorización (utilizando un mensaje *ARQ*) al Gatekeeper antes de responder a una llamada o iniciarla mediante señalización Q.931. Esto está motivado por el encapsulamiento de los mensajes de calidad intercambiados entre clientes y Gestor de Calidad de Servicio en el canal RAS de H.323. (Ver 4.3).
- Los clientes utilizarán una cola de tamaño constante (configurable mediante señalización de calidad de servicio) para compensar la fluctuación del retardo. De esta forma se asegura la calidad solicitada, ya que los esquemas dinámicos de dimensionamiento para la compensación de la variación del retardo permiten ciertos niveles de pérdidas de tramas.

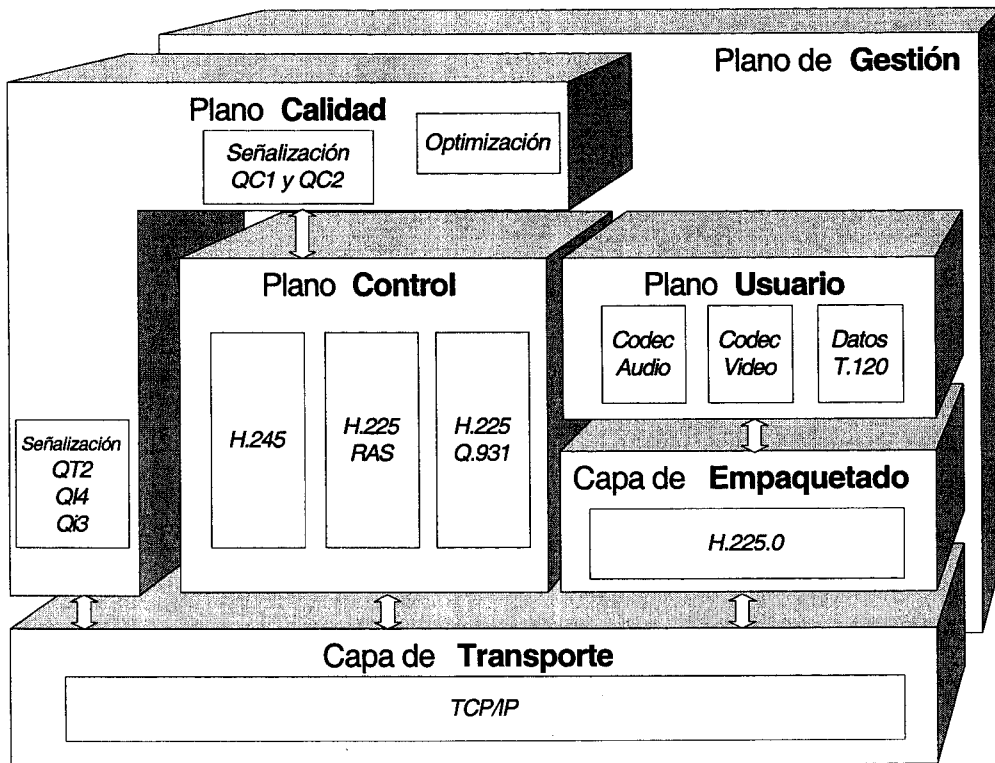


Figura 4.2: Arquitectura del Modelo

Las consideraciones anteriormente expuestas sirven como punto de partida para el modelo que se describe a continuación.

4.2. Arquitectura propuesta

La arquitectura propuesta para el modelo se puede apreciar en la figura 4.2. En ella, existe un Plano de Usuario, uno de Control, otro de Calidad y rodeando a todos uno de Gestión. Así mismo existen dos capas, una de Transporte y otra de Empaquetado.

El usuario selecciona el nivel de calidad para la llamada haciendo uso del Plano de Gestión. El Plano de Calidad se encarga de garantizar dicho nivel durante la llamada con el menor coste.

El Plano de Calidad deberá para ello seleccionar de forma óptima los parámetros de funcionamiento de terminales y red de transporte. Para establecer los valores seleccionados en el terminal, el Plano de Calidad deberá interactuar con el Plano de Usuario por medio del Plano de Gestión, configurando en ambos terminales el codec, tamaño de la cola para compensar la

variación del retardo y el empaquetado.

Los parámetros de funcionamiento exigidos a la red se indicarán por procedimientos del Plano de Calidad. Los mensajes correspondientes serán transportados en parte por el canal RAS (en las interfaces QC1 y QC2) y en parte directamente por la Capa de Transporte (en las demás interfaces).

En los clientes no existirá la entidad de optimización dentro del plano de calidad, ya que esta se realiza de forma centralizada en el Gestor de Calidad de Servicio. Así mismo, en los Gestores de Calidad no existirá el Plano de Usuario. En el resto de las entidades funcionales, los únicos planos existentes serán los de Calidad (sin optimización) y Gestión (además de la capa de Transporte).

El esquema propuesto se centra en el Plano de Aplicación, por lo que las entidades funcionales del Plano de Transporte serán tratadas como procesos en equipos independientes ya que su implementación no es objeto del presente trabajo.

A continuación se describen las principales decisiones de diseño utilizadas en el modelo propuesto. Posteriormente se realizará una introducción al funcionamiento básico de dicho modelo.

4.3. Especificaciones de Diseño

El desarrollo e implementación de un modelo de calidad exige la toma de una serie de decisiones relativas al diseño, que se resumen a continuación.

4.3.1. Ubicación de las Entidades Funcionales

El modelo propuesto en este capítulo se aplica sobre redes de VoIP que responde al modelo descrito por la ITU-T en su Recomendación H.323. Debido a ello la ubicación física de las entidades será:

- QoS del dominio de Usuario (Cliente): Se implementará en el mismo dispositivo que el cliente H.323, modificando éstos para que adquieran las funcionalidades adicionales necesarias. Los motivos para ello son:
 - En [37] no se define una interfaz normalizada entre las dos entidades, ni es objeto de la presente Tesis la definición de dicha interfaz.
 - Cliente H.323 y QoS del dominio de usuario deben compartir información de configuración, como codec a utilizar, momento de inicio y fin de la llamada, tamaño de la cola para compensar la variación del retardo y número de tramas por paquete.

Por tanto, la relación entre ambas entidades se realizará por medio del Plano de Gestión y de forma propietaria.

- QoS del dominio de Aplicación (Gestor de Calidad): Se implementará sobre el mismo dispositivo que el GateKeeper. El principal motivo para ello es también que ambas entidades deben compartir información como: clientes implicados en la conversación, estado de una llamada, lista de conexiones pendientes, inicio de una llamada ... Debido que en [37] no se define una interfaz normalizada entre estas dos entidades y que dicha definición no es objeto del presente trabajo, se ha optado por esta solución.

4.3.2. Ubicación de la señalización de calidad

Las PDU's de señalización de calidad de servicio, según se especifica en [37], deben llevar ciertos elementos de información. No obstante, el formato completo de la PDU de señalización no se especifica, con el objetivo de permitir la libertad de diseño. Existen dos posibilidades para ubicar la señalización:

1. Definir nuevas PDU's que transporten los elementos de información correspondientes a la calidad.
2. Enviar los elementos de información de calidad encapsuladas dentro de otras PDU's de señalización de la Recomendación H.323.

Debido a que uno de los problemas respecto a la calidad de la voz sobre IP reside en el excesivo tiempo de establecimiento de conexión. La primera opción produce un incremento en el tiempo de establecimiento de conexión. Por ello, la solución ideal sería enviar dichos elementos de información encapsulados dentro de unidades de datos de los protocolos que exige obligatoriamente H.323, a fin de no incrementar el tiempo de establecimiento de conexión.

Partiendo de que la utilización del GateKeeper es obligatoria, al inicio de la llamada los clientes deberán enviar un mensaje *ARQ* del canal RAS al Gatekeeper, y tras ello esperan recibir un mensaje *ACF* (o *ARJ*) del Gatekeeper con información para iniciar la señalización Q.931 de llamada. Por tanto, las PDU's de Solicitud de calidad (QoSrequest) que envía el cliente al Gestor de Calidad de servicio (ubicado en el GateKeeper) podrían encapsularse dentro del mensaje *ARQ*.

Así mismo, la respuesta a la solicitud de calidad (QoSconfirm o QoSreject) se encapsulará dentro de la respuesta del *ARQ*, es decir, en el *ACF* que devuelve el Gatekeeper al cliente.

Para encapsular dichas primitivas existen varias posibilidades:

- Definir en el código ASN.1 del canal RAS una nueva primitiva *ARQ* modificada que incluya la señalización de calidad
- Aprovechar el campo ASN.1 Generic Data que incluyen los mensajes del canal RAS (entre ellas *ARQ* y *ACF*) para introducir en él la señalización de calidad.

La primera posibilidad exige la modificación del código ASN.1 de la norma H.225.0, mientras que la segunda supondría utilizar uno de los tipos de datos que componen un tipo 'Generic

Data' (tipo *raw*, cadena de octetos), y por tanto seguiría siendo compatible con un esquema de señalización sin calidad. La opción empleada es, por tanto, la segunda. La definición ASN.1 de los nuevos tipos de datos para la señalización de calidad se especifica en el Apéndice A.

Toda la señalización de QoS del plano de transporte utilizará el protocolo UDP, lo que mejora el tiempo de envío de los mensajes de QoS. Como contrapartida los mensajes se podrán perder, ya que UDP no utiliza mecanismos de detección de errores. Por ello es necesario emplear mensajes de confirmación y temporizadores que permitan solucionar el problema de las pérdidas de mensajes en la red. Es de suponer que la tasa de pérdidas de paquetes en la red de transporte no debe ser muy grande (ya que la calidad de la conversación se degrada muy rápidamente con la pérdida de paquetes). Además si la red admite soporte calidad, los mensajes de QoS pueden ir siempre por un canal con alto nivel de calidad.

4.3.3. Comienzo y Fin de las llamadas

■ Determinación del comienzo de las llamadas

El Gestor de Calidad de Servicio deberá solicitar al TRM el grado de calidad de servicio en la red cuando comience la conversación. El punto en el cual se inicia la conversación (apertura de canales de medios) deberá por tanto ser conocido por el GK.

Para ello existen varias posibilidades:

1. Suponer que la conversación comienza cuando algún cliente recibe un mensaje Q.931 de *CONNECT*. En tal caso, el Gestor de Calidad de Servicio deberá conocer cuándo los clientes intercambian dicho mensaje. Para ello existen dos formas:
 - Que el GateKeeper encamine la señalización Q.931
 - Que el GateKeeper indique al cliente que al envío o recepción de mensajes Q.931 le mande una indicación. (esto se puede realizar con mensajes del canal RAS, como *ARQ* (campo *will supply UUIE*)).

Las ventajas de la primera alternativa son que no requiere intervención del cliente y que el cliente no necesita modificaciones para soportar este formato, ya que el encaminamiento de la señalización Q.931 por parte del GK es una decisión tomada por el GK, y todos los clientes lo aceptan sin necesidad de ningún mecanismo extraordinario para ello. El inconveniente que puede presentar es que se incrementa el tiempo de establecimiento de la conexión cuando el GateKeeper no esté en el camino entre origen y destino.

2. Suponer que la conversación comienza cuando se abren (mediante H.245 *Open logical Channel ACK*) los canales de medios. Para que el GK sepa cuándo ocurre esto, deberá enrutar la señalización H.245.

La primera alternativa adolece de dos problemas principalmente:

- A pesar de recibir *CONNECT*, podrían no abrirse los canales lógicos por algún problema y sin embargo el Gestor de Calidad de Servicio solicitaría calidad al dominio de transporte
- Hasta la apertura de canales lógicos con H.245 no se conocen los puertos que utilizan los clientes para el transporte de medios con RTP. En este caso, el Gestor de Calidad de Servicio no puede conocer el puerto al que envían los distintos medios ambos clientes, y por tanto, el TRM deberá autorizar en la ICF únicamente las direcciones de red. Esto presentaría problemas si ambos clientes además de la voz intercambian otro tipo de medios, ya que dicha información disfrutaría de la misma calidad de servicio que los medios de voz, decrementando así la calidad solicitada.

Por esto se opta por emplear la segunda alternativa descrita como método para determinar el comienzo de las llamadas

■ Determinación del final de las llamadas

Tras finalizar la conversación, el Gestor de Calidad de Servicio deberá solicitar al TRM que libere la reserva de calidad realizada en el dominio de transporte.

Para ello el Gestor de Calidad de Servicio deberá saber cuándo termina una conversación. Las alternativas posibles para realizar dicha tarea son tres:

1. Suponer que la conversación termina cuando un cliente recibe un mensaje Q.931 de *DISCONNECT*. En tal caso, el Gestor de Calidad de Servicio deberá conocer cuándo los clientes intercambian dicho mensaje. Para ello existen dos formas:
 - Que el GK encamine la señalización Q.931
 - Que el GK indique al cliente que cuando envíe o reciba mensajes Q.931 se lo indique al GK (esto se puede realizar con algunos mensajes del canal RAS, como *ARQ* (campo *will supply UUIE*)).

Las ventajas de la primera alternativa son:

- no requiere intervención del cliente
- el cliente no necesita modificaciones para soportar este formato, ya que el encaminamiento de la señalización Q.931 por parte del GK es una decisión tomada por el GateKeeper, y es aceptada por los clientes sin necesidad de ningún mecanismo extraordinario para ello.

El inconveniente que puede presentar es que se incremente levemente el tiempo de establecimiento de la conexión cuando el GK no esté en el camino entre origen y destino.

Por otra parte, la segunda alternativa elimina parte de esta última desventaja, sin embargo requiere que todos los clientes soporten esa característica, así como una colaboración por parte de los mismos. Por ello, es preferible la primera alternativa.

2. Suponer que la conversación termina cuando se cierran (mediante H.245 *Close logical Channel ACK*) los canales de medios. Para que el GK sepa cuándo ocurre esto, deberá enrutar la señalización H.245.
3. Suponer que la conversación termina cuando se envía el mensaje H.245 *endSessionCommand* (tras haber cerrado todos los canales lógicos). En este caso, el GK deberá encaminar la señalización H.245.

Teniendo en cuenta que según la Recomendación H.323 los clientes pueden cerrar el canal Q.931 tras abrir el canal H.245, la primera alternativa quedará descartada.

La alternativa segunda sería la más fiable, pues indica exactamente el momento en que se cierran los canales de medios. Tras del análisis de los clientes H.323 de distintos fabricantes, se deduce que un gran porcentaje de los mismos no emplea este mensaje para cerrar los canales de medios, sino directamente el mensaje H.245 *endSessionCommand*, por lo que es la tercera alternativa la que se tomará para el cierre de la conversación (y por tanto para el envío de la señalización de liberación de la calidad reservada).

4.4. Funcionamiento Básico

La figura 4.3 representa el funcionamiento básico del esquema propuesto en el caso de una llamada con calidad de servicio.

Se parte de un sistema que cumple con los requerimientos expuestos en el apartado anterior. Por tanto los dos clientes se encuentran registrados en el Gestor de Calidad de Servicio (ubicado en el Gatekeeper) con los siguientes datos:

- Alias del cliente
- Nivel de Calidad por omisión (formato MOS)
- Nivel de Calidad máximo alcanzable
- Tabla de retardos del cliente y subred de acceso

Todos los mensajes de señalización de Calidad enviados entre clientes y Gestor de Calidad de servicio, serán encapsulados en mensajes de señalización H.225.0 RAS a fin de no incrementar el tiempo de establecimiento de la conexión. El resto de los mensajes de Calidad son enviados directamente sobre UDP.

1. El cliente A quiere efectuar una llamada con Calidad de Servicio contra el cliente B. Para ello el cliente A envía un mensaje de servicio de Calidad **QoSMrequest**, que se encapsula dentro del mensaje inicial *ARQ* del canal RAS. Entre los elementos de información que contiene figuran el nivel de calidad exigido por el cliente (MOS) y los codecs que éste puede utilizar en la presente llamada.

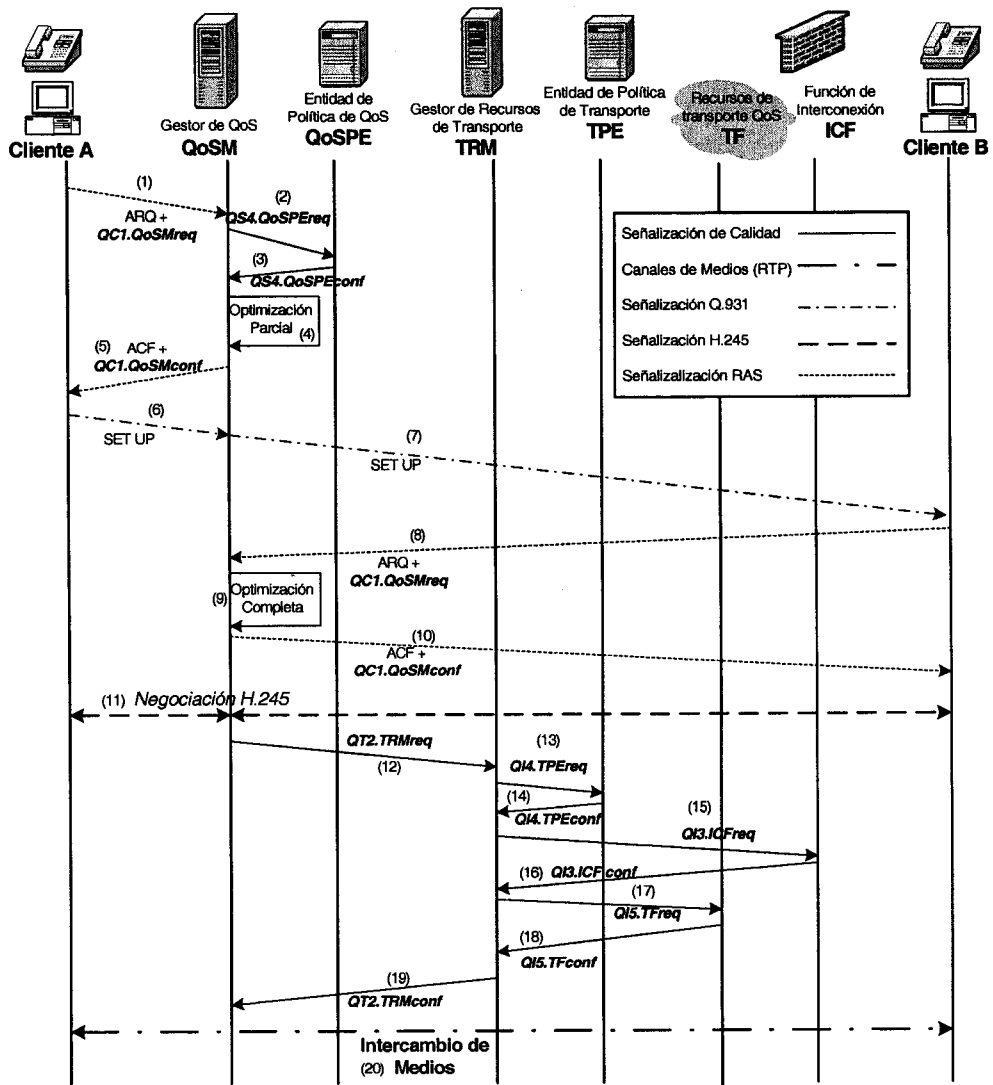


Figura 4.3: Diagrama del Modelo Propuesto

2. El Gestor de Calidad de Servicio solicita autorización de la calidad solicitada por dicho cliente al QoSPE mediante el envío de un mensaje **QoSPErequest** con los grupos de parámetros especificados en la norma [37].
3. El QoSPE debe comprobar que tanto llamante como llamado cumplen el esquema de calidad de servicio (ambos deberán estar registrados como aptos para utilizar calidad). En caso afirmativo devuelve un mensaje **QoSPEconfirm** al Gestor de Calidad de Servicio autorizando la llamada con calidad. Si alguno de los dos clientes no estuviera registrado para utilizar la calidad o bien no se encuentra autorizado por algún otro motivo, se enviará de vuelta un mensaje **QoSPEreject** y se abortará el proceso normal de la llamada.
4. Una vez se ha comprobado que ambos clientes están autorizados a efectuar la llamada con calidad, el Gestor de Calidad de Servicio realiza una optimización en virtud de la cual se proporciona, basándose en las funciones de coste del proveedor de Transporte y en las características de los clientes (retardos y codecs posibles), la configuración de menor coste que permita alcanzar la calidad solicitada por el cliente llamante (ver figura 4.1). Esta optimización se detallará en el apartado 4.6.

Como resultado de dicho proceso se obtendrá el punto de funcionamiento óptimo (7), que se compone de:

- Parámetros óptimos de los clientes: Codec, Tramas por paquete y Tamaño de la cola para compensar la variación del retardo.
- Parámetros de la red de transporte: Retardo medio, Variación del retardo máxima y Tasa de pérdida de paquetes.

Este planteamiento presupone que el cliente destino soportará el codec seleccionado como óptimo, cosa que no tiene por qué ser siempre cierta. Por lo tanto, esta optimización podrá ser refinada por otra posterior que se realizará cuando se tenga información sobre los codecs que soporta el cliente destino. Por ello, esta fase se denomina, **Optimización Parcial**

5. El Gestor de Calidad de Servicio responde al *ARQ* recibido del cliente llamante con un mensaje **QoSMconfirm** encapsulado en un *ACF* del canal RAS. En dicho mensaje *ACF* se establecerá que el encaminamiento de la señalización Q.931 se realice a través del Gatekeeper. Entre los elementos de información que figuran en la primitiva de calidad se encuentran:
 - Nivel de calidad que le ha sido asignado (siempre igual al solicitado a menos que no tenga autorización, en cuyo caso se le asigna el más próximo posible).
 - Parámetros a configurar en el cliente: Codec, Tramas por paquete y Cola de compensación de la variación del retardo.

Una vez recibidos estos parámetros el cliente se configura en función de ellos, ordenando la lista de codecs para la negociación H.245, estableciendo las tramas por paquete y ajustando el tamaño de la cola para compensar la variación del retardo.

6. El cliente llamante envía un mensaje Q.931 de solicitud de establecimiento de conexión (mensaje SET UP especificado en la norma H.225.0) para indicar el destino con el que desea establecer una conexión.
7. El GateKeeper encamina dicho mensaje hasta el cliente llamado.
8. El cliente llamado recibe el mensaje de solicitud de establecimiento de conexión y solicita autorización al Gestor de Calidad de servicio para responder a la llamada con calidad. Para ello, el cliente llamado envía un mensaje de calidad de servicio **QoSMrequest** encapsulado en un *ARQ* del canal RAS al Gestor de Calidad de Servicio. En este caso, la calidad solicitada por el cliente llamado es ignorada, pues la calidad se fija en el llamante (como ocurre en las redes RTC). En dicho mensaje de calidad figura una lista con los codecs que el cliente llamado puede utilizar.
9. El Gestor de Calidad de Servicio ya dispone de los codecs que ambos cliente pueden aplicar, por lo que existen dos situaciones posibles:
 - Que el cliente llamado disponga del codec seleccionado en la fase de optimización parcial, en cuyo caso se adopta como válido el punto seleccionado en aquella fase.
 - Que el cliente llamado no disponga del codec seleccionado en la fase de optimización parcial, en cuyo caso deberá realizarse una nueva optimización utilizando para ello los codecs que ambos clientes tienen en común y fijando el tamaño de la cola para compensar la variación del retardo al valor seleccionado en la optimización parcial (pues ya quedó establecido en el cliente llamado).

En cualquier caso a esta fase se le llamará **Optimización Total**, y de ella saldrá el punto de funcionamiento definitivo.

10. El Gestor de Calidad de Servicio envía un mensaje **QoSMconfirm** encapsulado en un *ACF* del canal RAS de vuelta al cliente llamado, autorizando el uso de calidad de servicio. En dicho *ACF* se fija el encaminamiento de la señalización a través del Gatekeeper. Entre los elementos de información que contiene se encuentra el codec seleccionado en la Optimización Total, así como el número de tramas por paquete y el tamaño de la cola para compensar la variación del retardo. El cliente llamado tras recibir este mensaje establecerá como único codec para la negociación H.245 el que le ha sido asignado mediante la Optimización Total, y con el correspondiente número de tramas por paquete. Además, fijará el tamaño de la cola para compensar la variación del retardo al valor especificado en el punto óptimo.
11. Ambos clientes inician la negociación H.245 en la que será seleccionado el codec a utilizar. Como el cliente llamado sólo presenta un codec como válido, se fuerza en la negociación H.245 a que el cliente llamante seleccione el codec que el llamado le indica (fijado en la optimización total).

A partir de este punto, el funcionamiento es el propuesto por [37]:

12. El Gestor de Calidad de Servicio envía al TRM un mensaje **QT2. TRM request** para solicitar a la red la calidad (Tasa de Pérdida de paquetes, Máxima variación del retardo y

retardo medio) que ha sido calculada en la fase de optimización. En dicho mensaje figuran los parámetros descriptores de tráfico y las direcciones de transporte de ambos clientes.

13. El TRM consulta a su servidor de autorizaciones TSE (mediante el envío de un mensaje **QI4. TPE request**) para comprobar si la solicitud anterior puede ser autorizada en el plano de transporte.
14. El TRM recibe autorización del TSE (recibe el mensaje **QI4. TPE confirm**) para proveer la calidad solicitada a el flujo de medios.
15. El TRM solicita a la función ICF mediante el mensaje **QI3. ICFrequest** que permita el paso de el flujo de medios dentro del dominio de transporte del TRM.
16. La función ICF responde afirmativamente al TRM (**QI3. ICFconfirm**), confirmando así que el flujo de medios podrá pasar por el presente dominio de transporte.
17. El TRM pide a la funcionalidad de transporte (TF), que ofrezca al flujo de medios una tasa de pérdida de paquetes, variación del retardo y retardo medio determinados. Para ello, envía el mensaje **QI5. TF request** aun no definido en la versión vigente de norma [37].
18. La funcionalidad de transporte le confirma al TRM que dicho flujo de medios será afectado por el retardo, variación del mismo y tasa de pérdida de paquetes solicitados mediante el mensaje **QI5. TF confirm**.
19. El TRM confirma al QoSM que la calidad solicitada para el flujo de medios ha sido aceptada por parte del dominio de transporte (**QT2. TRM confirm**).
20. Ambos clientes, tras haber realizado la negociación H.245 y abrir sus canales RTP, comienzan el intercambio de medios que contiene voz digitalizada.

La finalización de la conversación sería provocada por el cierre en alguno de los canales de medios, y sería gobernada por el Gestor de Calidad de servicio, ya que el Gatekeeper enruta la señalización.

A continuación se pasa a describir la estructura funcional de cada una de las entidades que intervienen en este esquema.

4.5. Entidades Funcionales del modelo

Las entidades funcionales que se han implementado en el presente esquema serán descritas mediante una breve explicación y un diagrama SDL que especifica su comportamiento.

Ciertas entidades funcionales han sido implementadas en el mismo dispositivo físico que implementa otra entidad de H.323, como clientes y Gatekeeper. En dichos casos, la descripción del comportamiento hará referencia sólo a la parte del funcionamiento que afecta a la QoS, por lo que no se especificará el resto del comportamiento de las entidades H.323, pudiéndose éste tomar de dicha Recomendación.

4.5.1. Clientes con Soporte de Calidad (QoSM del Dominio de Usuario)

El funcionamiento del cliente con calidad de servicio será el siguiente:

- Cuando dicho cliente quiera efectuar una llamada con QoS:

Deberá enviar al QoSM del dominio de aplicación el mensaje **QC1. QoSMrequest**.

Dicho mensaje se encapsula en un *ARQ* del canal H.225.0 RAS, a fin de no incrementar el tiempo de establecimiento de llamada con una primitiva de servicio independiente. Entre los elementos de información que contiene están:

- Calidad solicitada por el cliente (MOS) (parámetro: QoS Service Class)
- Lista con los codec que se pueden utilizar y las tramas por paquete (parámetros: *Codec Type* y *Packetisation*)

El resto de los grupos de parámetros que especifica la norma se encuentran presentes con valor nulo. La MOS solicitada, así como la lista de codecs utilizables, serán configurables por el usuario en cada llamada.

Una vez enviado el *ARQ* con el mensaje **QC1. QoSMrequest** el cliente deberá esperar la llegada de un *ACF* que encapsule un **QC1. QoSM confirm**. Si este mensaje de QoS no llegara, continuará su funcionamiento normal, asumiendo que la llamada no tendrá calidad de servicio (establecimiento de los canales Q.931, H.245 y canales de medios).

Cuando se recibe el mensaje **QC1. QoSM confirm**, ésta contendrá los siguientes elementos de información:

- Calidad solicitada por el cliente (MOS) (parámetro: QoS Service Class)
- Lista ordenada de los codec que debe utilizar y su empaquetado (parámetros: *Codec Type and Packetisation*)
- Retardo Medio, variación del retardo y tasa de pérdida de paquetes que se solicitará a la red (parámetros: *Transport QoS Parameters*)
- Régimen binario y máximo tamaño de la PDU (parámetro: *Traffic Descriptor*)

El resto de los grupos de parámetros que especifica la norma se encuentran presentes pero con valor nulo.

El cliente, tras recibir el **QC1. QoSM confirm**, establecerá los codecs para la negociación H.245 en el orden que le han sido asignados, y con los correspondientes empaquetados. Además, fijará la cola para compensar la variación del retardo al valor especificado como máxima variación del retardo que se solicitará a la red.

Después, continuará el procedimiento normal de una llamada H.323, es decir, establecimiento de los canales Q.931, H.245 y canales de medios.

- Cuando dicho cliente reciba una llamada con QoS:

Tras recibir la primitiva de H.225.0 *SET UP*, enviará un *ARQ* al GK en dónde se encapsulará una primitiva de calidad de servicio **QC1. QoS request**, procediendo exactamente igual que en caso anterior. La única diferencia respecto al caso anterior es que al recibir un **QC1. QoS confirm** encapsulado en el *ACF*, el cliente deberá seleccionar sólo el primer codec de la lista ordenada, de tal forma que en la posterior negociación H.245 sea este el único codec que se presente (método para forzar la negociación).

En la figura 4.4 se detalla un diagrama SDL con el comportamiento del cliente en el que sólo se contemplan los procedimientos de señalización relativos a QoSM.

4.5.2. Gestor de Calidad de Servicio (QoSM del Dominio de Aplicación)

El funcionamiento del QoSM será el siguiente:

Cuando recibe una petición de llamada (*ARQ*) realiza en un chequeo para averiguar si dicha petición corresponde a una conexión nueva (en cuyo caso proviene del llamante) o a una conexión ya solicitada (en cuyo caso proviene del llamado). El funcionamiento en ambos casos es el siguiente:

■ *ARQ* del llamante

Se comprueba si la primitiva lleva encapsulada la PDU de calidad de servicio **QC1. QoS request**. En caso negativo, se configura esta conexión como sin calidad de servicio (No QoS) y se continúa con el funcionamiento normal de GK según la Recomendación H.323.

Si existiera el mensaje **QC1. QoS request**, se extraen los datos relevantes, que son MOS solicitada y lista de codecs utilizables por dicho cliente. Tras esto, se solicita autorización para que dicho cliente efectúe una llamada con QoS.

Para ello se envía un mensaje **QS4. QoSPerequest** al QoSPE, en la que figuran datos como identidad de llamante y llamado, así como calidad solicitada. Si no se recibe ninguna respuesta del QoSPE, o bien se recibe respuesta negativa (mensaje **QoSPereject**) se configura esta conexión como sin calidad de servicio (No QoS) y se continúa con el funcionamiento normal del GK según la Recomendación H.323.

En caso de recibir autorización del servidor QoSPE, se realiza la fase de optimización, detallada en el apartado 4.6. En dicha fase se obtiene el punto de mínimo coste que permite obtener la MOS solicitada.

Este punto óptimo tiene los valores de funcionamiento tanto para el cliente (*Codec, tramas por paquete y tamaño de la cola para compensar la variación del retardo*), como para la red (*retardo medio, variación del retardo y tasa de pérdidas de paquetes*). Estos valores se almacenarán como parámetros de funcionamiento de la conexión QoS, y se enviarán de vuelta al cliente en el mensaje **QC1. QoSconfirm encapsulado en el *ACF* generado como respuesta al *ARQ* recibido. Dicho *ACF* transportará el número de los puertos empleados para el intercambio de medios, junto con la dirección del propio GK como dirección destino (lo que permite enrutar la señalización Q.931 y H.245).**

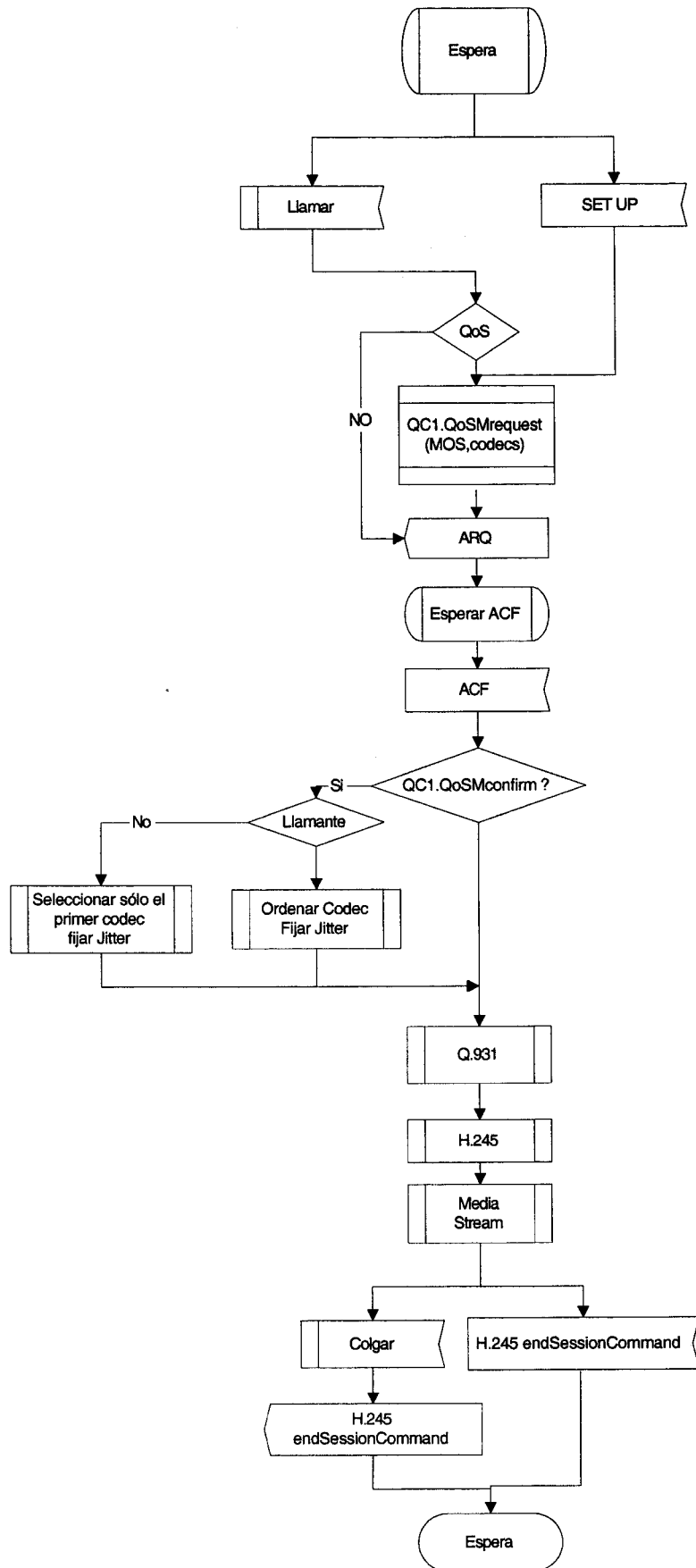


Figura 4.4: Diagrama SDL del cliente QoS

la dirección del propio GateKeeper como dirección destino, para enrutar de esta manera la señalización Q.931 y H.245, requisito necesario para saber el comienzo y final de las comunicaciones, así como los puertos utilizados en el intercambio de medios.

■ **ARQ del llamado**

Tras recibir el **ARQ** se realiza un chequeo para verificar si corresponde a una conexión con QoS o no. En este último caso, se continúa con el funcionamiento normal del GK como se especifica en la Recomendación H.323.

Si pertenece a una conexión con QoS, se comprueba si este **ARQ** contiene el mensaje **QC1. QoSMrequest**. En caso negativo se elimina la característica QoS de la conexión y rechaza la petición **ARQ** conexión enviando un mensaje **ARJ** del canal RAS. El motivo para rechazar la conexión es que un cliente ya tiene fijado el tamaño de la cola para compensar la variación del retardo. Como no existe comunicación de calidad con el llamado, no es posible asegurar el nivel de QoS requerido, por lo tanto no se puede cursar la llamada solicitada.

Si el **ARQ** encapsula una PDU **QC1. QoSMrequest**, se extrae de ésta la lista de codecs que soporta el cliente destino. Si entre esos codecs se encuentra el que fue seleccionado como parte del punto óptimo en las características de la conexión QoS, dichas características se envían de vuelta al cliente destino en un mensaje **QC1. QoSMconfirm**, encapsulado en el **ACF**.

Si en la lista de codecs que soporta el cliente destino no se encuentra el codec del punto óptimo, deberá realizarse una nueva optimización teniendo en cuenta que la cola para compensar la variación del retardo debe permanecer al valor establecido en el antiguo punto óptimo, pues ya está fijado en el cliente llamante y no hay manera de cambiarlo. Los parámetros del nuevo punto óptimo se envían de vuelta al cliente en una PDU **QC1. QoSMconfirm**, que se encapsulará en el **ACF** que el GK devuelve al cliente como respuesta al **ARQ**. Dicho **ACF** estará preparado para enrutar la señalización del cliente (Q.931 y H.245).

Una vez recibidos y procesados ambos **ARQ**, los clientes establecen su canal de señalización H.245 para negociar la capacidad de los canales de medios que se abrirán. Esta señalización será enrutada por el GateKeeper para conocer el instante en que se abren y cierran los canales de medios a fin de solicitar y liberar la capacidad de transporte de la red IP. Si no existe H.245 *tunneling*, en la negociación de Q.931 se indicará que el GK enrutará el H.245. Si existe H.245 *tunneling*, el GK deberá decodificar e interpretar los mensajes H.245 encapsulados dentro del canal Q.931.

Cuando el GK reciba un mensaje H.245 *Open Logical Channel ACK* interpretará que se va a abrir un canal de medios para una conversación.

En primer lugar deberá verificar si dicha conversación tiene QoS solicitada. En caso negativo, re-enruta el mensaje hasta el destino sin realizar ninguna operación.

Si dicha primitiva corresponde a una conexión con soporte QoS, el GK extraerá las características de dicha conexión. Con estos datos y las direcciones origen y destino, genera la PDU **QT2. TRM request**, que se envía al TRM solicitando un cierto nivel de QoS en la red de transporte para un flujo de medios (la dirección IP se obtiene de los datos de la conexión, y el puerto

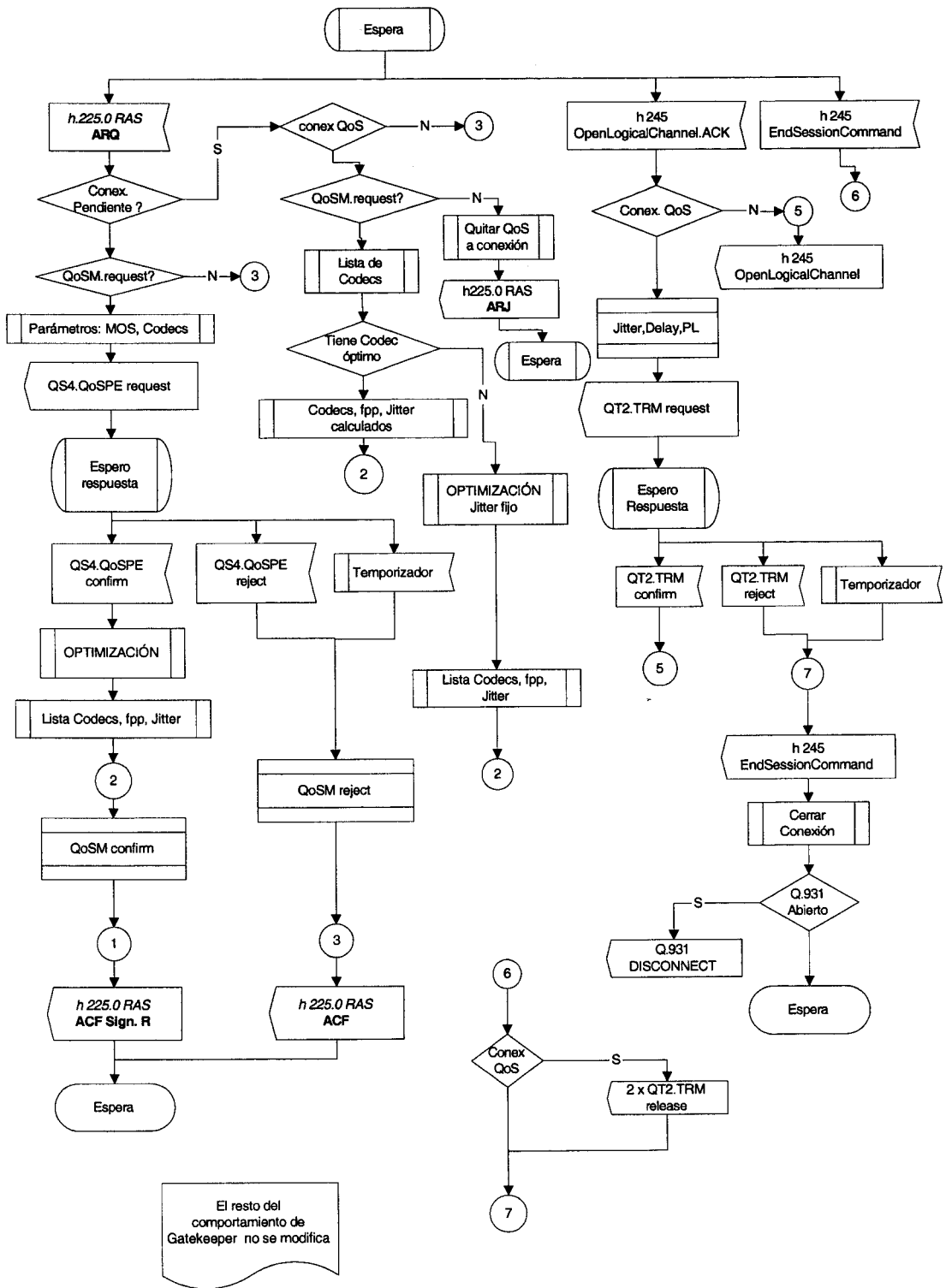


Figura 4.5: Diagrama SDL del Gestor de Calidad de Servicio

para el intercambio de los medios con RTP, se puede obtener del mensaje H.245 *Open logical Channel* enrutado por el GateKeeper).

Una vez enviado dicho mensaje el Gestor de Calidad esperará la llegada de una respuesta por parte del TRM. Si dicha respuesta no llegara, o bien llegara de forma negativa (**QT2. TRM reject**), se debe abandonar la comunicación, cerrándola de forma ordenada (ya que no se puede asegurar la calidad solicitada). Para ello se enviarán a los clientes los mensajes H.245 *EndSessionCommand* y los mensajes Q.931 *DISCONNECT*.

Por el contrario, si el TRM responde de forma afirmativa (se recibe una PDU **QT2. TRM confirm**), debe re-enrutarse el mensaje *H.245 OpenLogical Channel* hasta el otro cliente. Finalizada esta etapa, los dos clientes intercambian los medios que portan la voz digitalizada.

Una vez finalizada la conversación, deberán cerrarse los canales lógicos y anularse la solitud realizada a la red (TRM). Debido a que la Recomendación H.323 permite realizar el proceso de finalización de conversación de diferentes formas, el GateKeeper enviará dos primitivas para la liberación de la calidad solicitada al TRM (**QT2. TRM release**, una para cada sentido de la conversación) cuando llegue algún mensaje H.245 *CloseLogicalChannel* o H.245 *EndSessionCommand* perteneciente a una conexión con QoS.

El diagrama de funcionamiento del Gestor de Calidad de Servicio puede apreciarse en la figura 4.5.

4.5.3. Gestor de Recursos de Transporte (TRM)

El Gestor de Recursos de transporte se encuentra constantemente esperando la llegada de primitivas de solicitud de calidad de servicio de transporte (Régimen binario, Tasa de Pérdida de Paquetes, y Máxima Variación del Retardo) en la red para un flujo determinado entre dos usuarios.

Una vez que este recibe una PDU de solicitud de calidad (**QT2. TRMrequest**), en la que se especifican los parámetros de calidad, así como las direcciones de transporte de los clientes, el TRM debe solicitar autorización para dicha reserva al servidor de autorizaciones del plano de transporte; TPE. Para ello le envía un mensaje **QI4. TPErequest** en el que figuran las direcciones de transporte de ambos clientes y los parámetros de calidad de transporte solicitados.

Hecho esto, el TRM queda a la espera de recibir autorización para gestionar la reserva de calidad en dicho flujo. Si no la recibe pasado un tiempo, o si la recibe de manera negativa (**QI4. TPE reject**), envía un mensaje **QT2. TRM reject** al Gestor de Calidad de Servicio del plano de aplicación rechazando la solicitud de reserva de calidad en el plano de transporte.

En caso de recibir la autorización del TPE (**QI4. TPE confirm**), debe autorizar la entrada y salida del flujo entre los dos clientes en su dominio de transporte. Para ello, envía los mensajes **QI3. ICF request** a las funciones de interconexión (normalmente cortafuegos) del dominio de transporte, en las que se manda las direcciones de transporte que deben ser autorizadas.

Una vez enviadas estas primitivas se queda a la espera de recibir la confirmación por parte de las funciones ICF (mediante **QI3. ICF confirm**). Caso de no recibirlas en un cierto intervalo de tiempo, o bien, recibir una confirmación negativa (**QI3. ICF reject**), se enviará inmediatamente al

Gestor de Calidad de Servicio del plano de aplicación un mensaje **QT2. TRM reject** denegando la solicitud de calidad para el flujo.

Tras la recepción de la confirmación de ICF, se envían a la función de transporte una primitiva solicitando la aplicación de un nivel de Tasa de Pérdidas de Paquetes, Máxima variación del retardo y retardo medio al flujo de medios que circula entre ambos clientes. Esto se hace mediante el envío de la primitiva **QI5. TF request**.

Una vez recibida la confirmación por parte de ésta (**QI5. TF confirm**) se informa al Gestor de Calidad de Servicio del plano de aplicación sobre el éxito de la reserva en el plano de transporte mediante el envío del mensaje **QT2. TRM confirm**.

En caso de no recibir dicha confirmación pasado un cierto tiempo (o bien de recibir una confirmación negativa; **QI5. TF reject**) se informará al QoSM del plano de aplicación de que la reserva solicitada no ha tenido lugar (**QT2. TRM reject**), a la vez que se solicita a la función ICF que no deje pasar el flujo autorizado anteriormente (**QI4. ICF release**).

No se ha contemplado la utilización de mecanismos concretos de reserva de calidad de servicio (RSVP, MPLS, *DiffServ*) en el TRM debido a que el desarrollo del presente trabajo se centra en el plano de aplicación. Se asume la existencia de una 'caja negra' que ajusta los mecanismos necesarios en la función de transporte de tal modo que se asegure para el flujo de medios el retardo medio, máxima variación del retardo y tasa de pérdida de paquetes solicitados.

En la figura 4.6 se representa el diagrama SDL de comportamiento del Gestor de Recursos de Transporte.

4.5.4. Entidad de Política de Transporte (TPE)

Esta entidad tiene como misión autorizar las solicitudes de calidad de servicio de transporte en un dominio concreto. Para ello deberá, basándose en las direcciones de transporte de ambos clientes y en la calidad solicitada a la red de transporte, confirmar o denegar las solicitudes. Debido a que no es objeto del presente trabajo la definición de políticas de admisión, el funcionamiento propuesto es muy simple.

De esta forma, el esquema de funcionamiento será:

Cuando llegue un mensaje de solicitud de autorización desde el TRM (**QI4. TPE request**) se extraerá la información relevante de la misma: direcciones de transporte del flujo y calidad solicitada a la red función de transporte. Basándose en estos datos, se aceptará o no la concesión de QoS para el flujo. Si se autoriza el flujo se devolverá un mensaje **QI4. TPE confirm** al TRM, y en caso contrario se enviará un **QI4. TPE reject**. Tras ello el TPE quedará a la espera de recibir una nueva solicitud.

En la figura 4.7 se refleja el esquema SDL que describe el funcionamiento de TPE.

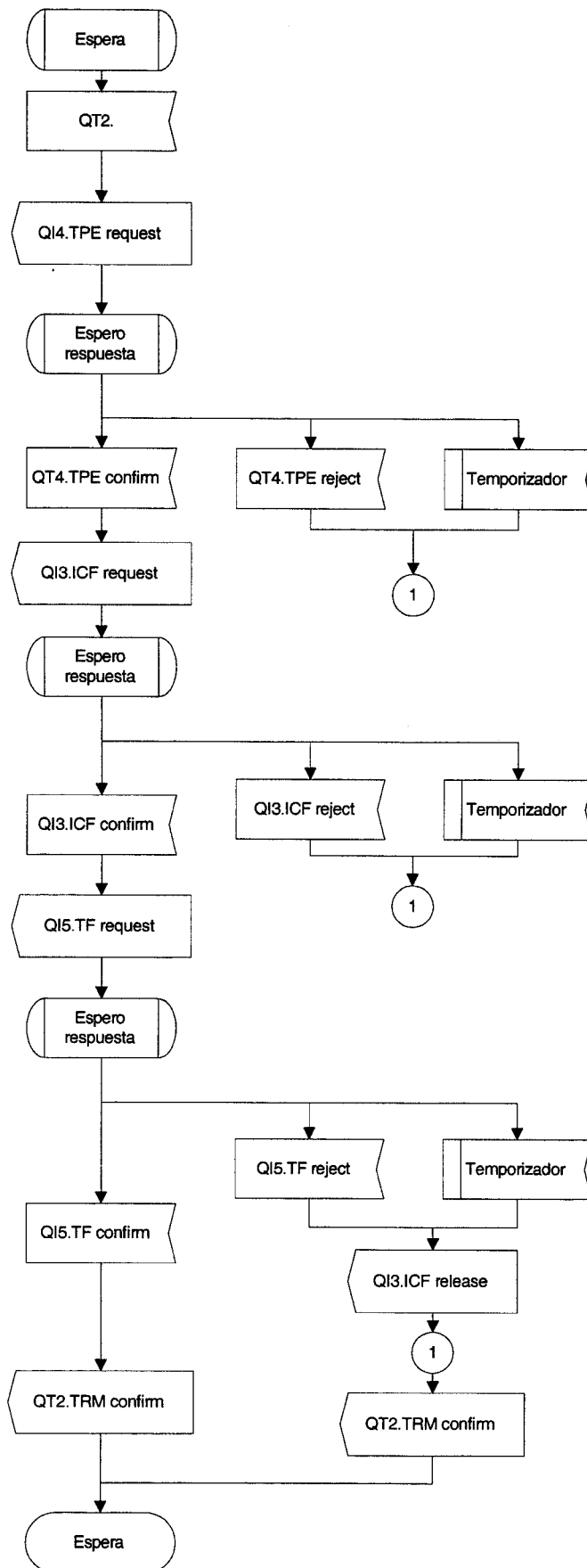


Figura 4.6: Diagrama SDL del TRM

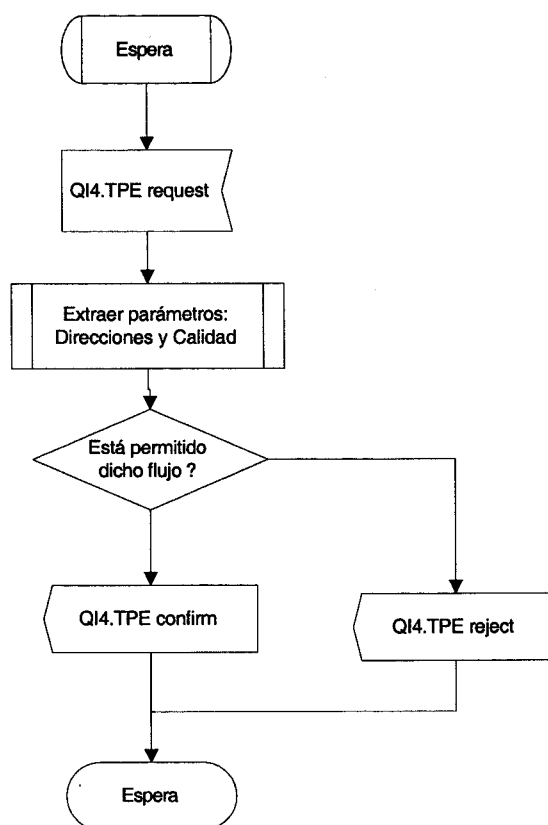


Figura 4.7: Diagrama SDL del TPE

4.5.5. Función de Interconexión (ICF)

La ICF incluye los elementos de interconexión del dominio de transporte con otros elementos fuera de dicho dominio. Normalmente esta función se encuentra implementada en cortafuegos (firewalls).

La entidad funcional ICF espera la recepción de un mensaje **QI3. ICF request** por parte del TRM, en el que se especifica la dirección de transporte de los clientes que van a intercambiar los flujos de medios. El ICF configura adecuadamente el cortafuegos para que deje pasar el flujo con origen y destino en las direcciones de transporte recibidas en la primitiva anterior y responde al TRM con un mensaje **QI3. ICF confirm**. Si no pudo realizar la configuración anterior por cualquier motivo, enviará un **QI3. ICF reject**.

El funcionamiento se puede observar en la figura 4.8

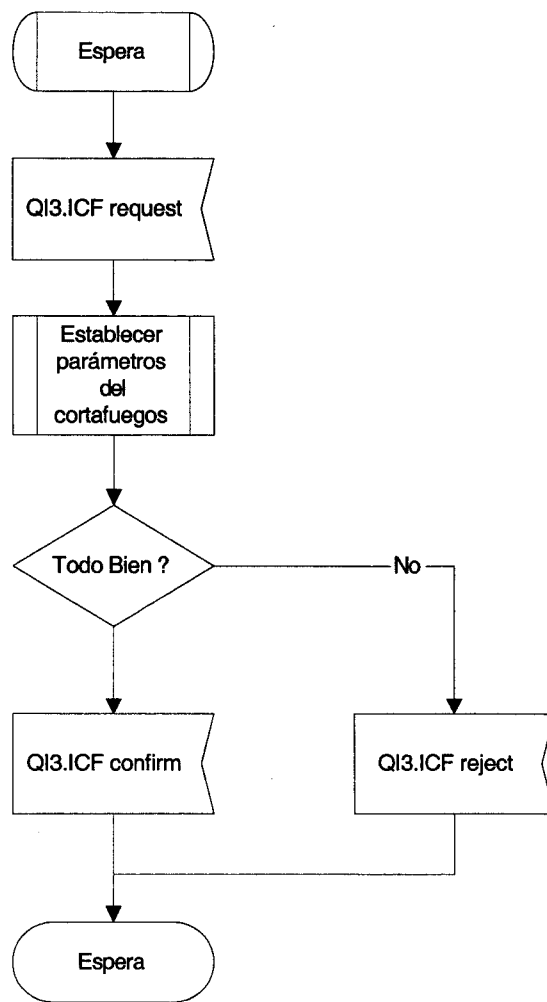


Figura 4.8: Diagrama SDL de la ICF

4.5.6. Función de Transporte (TF)

La TF está constituida por todos aquellos elementos de transporte IP, como encaminadores, conmutadores, con capacidad para el manejo de algún método que pueda garantizar la QoS (implementación de algún protocolo como MPLS o RSVP) en la red IP.

Como el objeto del presente trabajo se centra en el plano de aplicación, la funcionalidad de transporte puede ser ejecutada por un emulador de red en tiempo real. Este emulador permitirá simular la funcionalidad de transporte con bajo ciertas características de QoS, como retardo medio, tasa de pérdida de paquetes y máxima variación del retardo. Por tanto, la entidad funcional TF estará concentrada en un solo elemento (y no dispersada entre todos los elementos de la red de transporte QoS) cuyo comportamiento es el siguiente:

La TF permanece a la espera de un mensaje **QI5. TF request** procedente del TRM. En dicho mensaje deberán figurar las características de QoS de la red de transporte, las características del tráfico que se va a soportar (Régimen Binario y tamaño máximo de PDU) y las direcciones de transporte entre la que se va a establecer el flujo.

La entidad TF informará de estos datos al emulador de red, que afectará a partir de entonces a dicho flujo con este nivel de calidad (y velará por que no se sobrepasen las características que caracterizan a dicho flujo).

Si la reserva se ha podido efectuar en el emulador, el TF enviará un mensaje confirmando la reserva al TRM (**QI5. TF confirm**). Si por el contrario hubo algún tipo de problemas al ejecutar la reserva se enviará un **QI5. TF reject** indicando que no se ha podido efectuar la misma.

El diagrama de comportamiento SDL se muestra en la figura 4.9

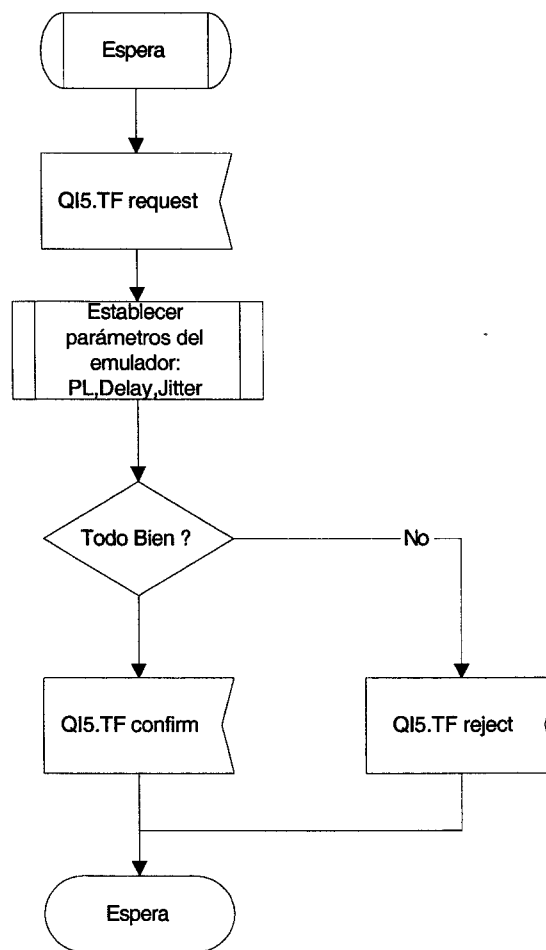


Figura 4.9: Diagrama SDL de la TF

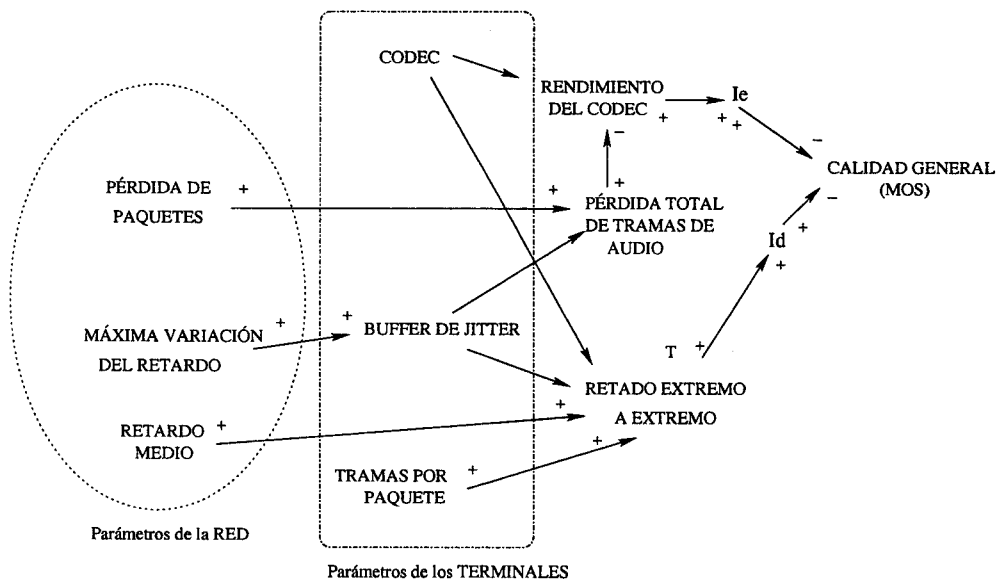


Figura 4.10: Diagrama Causal de la influencia de los parámetros sobre la calidad

4.6. Módulo de Optimización

Según se ha expuesto en las secciones anteriores, los factores que afectan a la calidad en las llamadas de voz sobre IP están localizados en terminales y red de transporte. Teniendo en cuenta que el caso de llamadas desde terminales de otras redes no será contemplado en este trabajo, los factores configurables en una llamada serán:

- **Factores localizados en la Red de Transporte IP:** Retardo Medio, Máxima variación del Retardo y Tasa de Pérdida de Paquetes.
- **Factores localizados en los terminales:** Codec utilizado, número de tramas por paquete, tamaño de la cola para compensar la variación del retardo.

El resto de los parámetros de los terminales son difícilmente configurables y dependen de circunstancias en ocasiones no controlables, como el ruido ambiente, la pérdida de efecto local, o las sensibilidades de micrófono y altavoz.

La influencia de los parámetros controlables en red y terminales sobre el Modelo-E (I_e : factor de equipo e I_d : factor de retardo) viene representada en la figura 4.10. En el Apéndice C de la presente memoria se exponen los procedimientos que permiten hallar los valores de entrada del modelo, así como un método que permite medir el retardo debido a los terminales basados en PC.

El grado de QoS necesario en la red de transporte para conseguir un cierto nivel de calidad en la llamada vendrá condicionado por la selección de los parámetros de los terminales realizada.

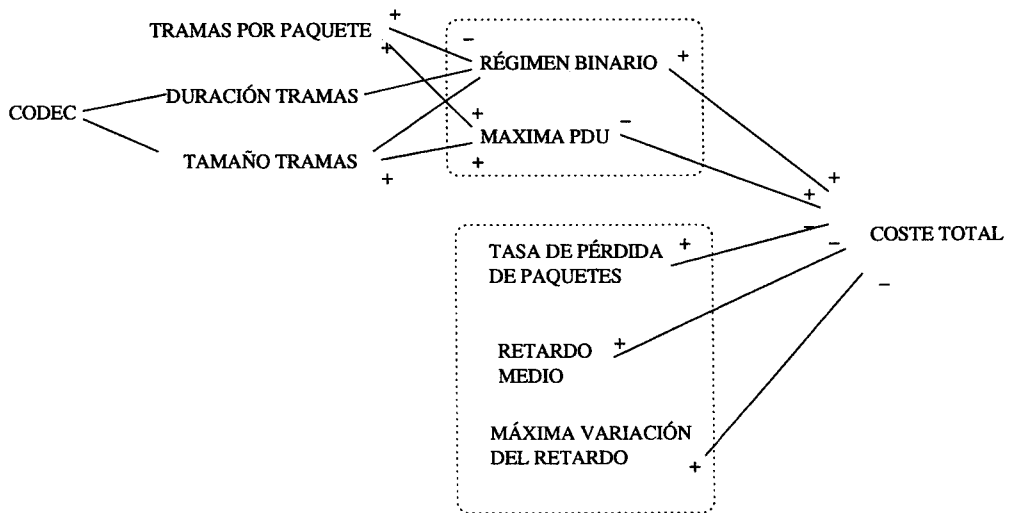


Figura 4.11: Diagrama Causal de la influencia de los parámetros en el coste

Además los parámetros de terminales tramas por paquete y codec tienen una influencia directa en el consumo de recursos de transporte, ya que determinan la caracterización del flujo de datos; es decir Régimen Binario a transmitir y Máximo tamaño de la PDU.

A su vez, los parámetros de terminales y red suponen un consumo de recursos en la red de transporte que, en buena lógica, debe tener un coste asociado. En la figura 4.11 se puede apreciar la influencia de los diversos parámetros de redes y terminales en el coste. Como se exponen en el apartado 4.6.1, para las redes IP con soporte de QoS el coste de transporte de un flujo de datos deberá ser función del nivel de QoS exigido y de la caracterización de dicho flujo.

El objeto del algoritmo es encontrar la configuración de los clientes (codec, tramas por paquete y tamaño de la cola para compensar la variación del retardo) que permita conseguir un cierto nivel de calidad con el mínimo coste(mínima utilización de los recursos de transporte: retardo solicitado, variación del mismo y tasa de pérdida de paquete).

4.6.1. Requisitos de partida

Como punto de partida para determinar los requisitos que debe cumplir el algoritmo tendremos:

1. **Funciones de Coste** : El proveedor de servicio de transporte deberá ofrecer las funciones de coste al proveedor del servicio de telefonía IP. De forma general un cliente i tendrá que pagar [38]:

$$C_i = a(F_i, Q_i)T_i + b(F_i, Q_i)V_i + c$$

Donde:

- F es un conjunto de descriptores de tráfico (Régimen binario y Máximo tamaño de la PDU)
- Q es el conjunto de especificaciones de calidad de servicio para dicho tráfico (Retardo Medio, Variación del retardo y Tasa de pérdida de paquetes)
- T es la duración de la conversación
- V es el volumen del tráfico cursado
- los coeficientes a , b y c pueden determinarse según [39]

Para simplificar sin pérdida de generalidad, se adoptarán las siguientes funciones de coste:

- Las funciones no tienen en cuenta el volumen, sólo la duración (no habrá disminución de precio por volumen, ya que esto se puede aplicar en la fase de tarificación), y todas son proporcionales a la duración de la conversación. El coeficiente b queda pues anulado.
- Las funciones podrán ser arbitrarias y discontinuas.
- Existen cinco funciones de coste independientes entre sí (una para cada parámetro libre). Esto es una simplificación que permite el manejo eficiente de las funciones, pues el uso de matrices no aporta nada al diseño algoritmo. Por tanto el coeficiente a , dependería de las siguientes funciones:
 - f_{Rb} : Función de coste del régimen binario del flujo. Cuanto mayor sea el régimen binario del flujo, mayor deberá ser el valor de la función
 - $f_{Max.PDU}$: Función de coste del Máximo Tamaño de la PDU. Cuando mayor sea el tamaño de la PDU menor será esta función, ya que el proveedor de transporte podrá aprovechar mejor sus recursos.
 - f_D : Función de coste del retardo medio. Cuanto mayor sea el retardo medio que se solicita, menor será el valor de esta función.
 - f_J : Función de coste de la Variación del retardo. Cuanto mayor sea la variación del retardo que se solicita, menor será el valor de esta función.
 - f_{PL} : Función de coste de la tasa de pérdida de paquetes. Cuanto mayor sea la tasa de pérdida de paquetes que se solicita, menor será el valor de esta función.

Así pues, la forma de la función de coste para un flujo i en este caso será:

$$C_i = (F_i + Q_i)T_i + c$$

donde las funciones F y Q valdrán:

$$\begin{aligned} F_i &= f_D + f_J + f_{PL} \\ Q_i &= f_{Rb} + f_{Max.PDU} \end{aligned}$$

La constante c quedará fuera del algoritmo de optimización.

2. Retardo del algoritmo de optimización.

La optimización se realizará por cada solicitud de conexión. Por ello, el cálculo del punto óptimo deberá realizarse en el establecimiento de la conexión entre dos clientes. Este hecho hace necesario minimizar el tiempo consumido por el algoritmo para permitir su uso en tiempo real.

3. Tipos de optimización.

Existirán dos tipos distintos de optimización. Una con todos los parámetros libres y otra con la variación del retardo y las tramas por paquete ya fijados. Esto permite contemplar el caso en el que el cliente destino no tenga disponible el codec que se obtuvo tras el proceso de optimización inicial.

4. Retardo de los clientes.

El algoritmo deberá tener en cuenta los retardos asociados a los distintos clientes en función del codec que implementen, así como los retardos de acceso a subredes. Para el caso en que no se disponga de esta medida en un determinado cliente se arbitrará un valor por defecto.

5. Codecs de los clientes.

Los codecs que los clientes puedan utilizar en la llamada se indicarán de forma dinámica como parte de la señalización de QoS en cada llamada. Esto permite una mayor flexibilidad, ya que por motivos de congestión en la red de acceso un cliente puede verse forzado en un momento a utilizar codecs de baja tasa de bit sin que el Gestor de Calidad de Servicio tenga conocimiento de ello.

4.6.2. Algoritmo propuesto

El algoritmo deberá evaluar la calidad y el coste de un cierto número de puntos. Teniendo en cuenta que el tamaño de la cola para compensar la variación del retardo los clientes siempre será igual a la variación del retardo solicitado a la red, los parámetros que se deben seleccionar con el algoritmo son cinco.

Parámetros en el cliente:

- codec
- tramas por paquete

Parámetros de rendimiento en la red:

- Retardo Medio
- Variación Máxima del Retardo
- Tasa Máxima de pérdida de paquetes

Con todos ellos, los parámetros de caracterización de tráfico (régimen binario y máximo tamaño de PDU) ya quedan fijados. Cada valor de estos parámetros dará lugar a un determinado nivel de calidad (MOS) y a un determinado coste. Partiendo de los siguiente datos

- Codecs posibles = 1:C

- Tramas por paquete = 1:FPP
- Retardo Medios posibles = 1:D (supuesto una resolución de 1 ms)
- Máxima Variación del Retardo = 1:J (supuesta una resolución de 1 ms)
- Tasa Máxima de pérdidas e paquetes = 1:PL

Por lo que el número de puntos posibles será

$$N_{puntos} = C \cdot FPP \cdot D \cdot J \cdot PL \quad (4.1)$$

Cada uno de los parámetros anteriores se encuentra limitado en circunstancias reales de tal forma que normalmente los valores serán:

- $C = 5$ (G.711 , GSM 6.20 , G.723.1 - 5,3k , G.723.1 -6.4k , G.729 A/B)
- $FPP = 32$. Como retardos a partir de 320 ms comienzan a ser no aceptables, y suponiendo que en G.711 las tramas son de 10 ms, el máximo número de tramas por paquete será 32
- $D = 500$. La unidad en que se mide será milisegundo. A partir de 320 ms la calidad se encuentra ya bastante degradada, por lo que un retardo medio de 500 ms será difícilmente superable por ningún punto.
- $J = 250$. La variación del retardo siempre es menor que el retardo, por lo que valores de 250 ms en la variación del retardo pueden dar lugar a retardos de 500 ms, que es el máximo admisible.
- $PL = 10$. La tasa de pérdida de paquete influye exponencialmente en la calidad. Tasa superiores a un 5 % (depende del codec) suponen una degradación severa en la calidad final, por lo que diez puntos (cada uno corresponde a 0,5 % de pérdida de paquetes) son suficientes.

Con los datos anteriores, el número de puntos posibles será de:

$$N_{puntos} = 5 \cdot 32 \cdot 500 \cdot 250 \cdot 10 = 200,000,000$$

Cada uno de ellos tendrá una MOS y un coste asociado, por lo que en función de ellos el proveedor del servicio de telefonía IP deberá seleccionar el punto óptimo.

La manera más intuitiva de implementar el algoritmo sería: partiendo del rango de MOS que se quiere conseguir en la llamada, evaluar el coste de los puntos que cumplen que la MOS correspondiente a ellos se encuentra dentro del rango. De entre esos puntos se elegirá el de menor coste. Esta primera alternativa, que se podría denominar *evaluación dinámica secuencial*, resulta inviable debido al excesivo retardo del algoritmo (con el número de puntos del caso real expuesto sería 2 minutos y 46 segundos en un PC Pentium III 600 MHz).

Analizando el número de puntos que cumplen con una determinada MOS, como se aprecia en la figura 4.12, se puede comprobar que para valores de MOS superiores a 4 sólo existen 1328

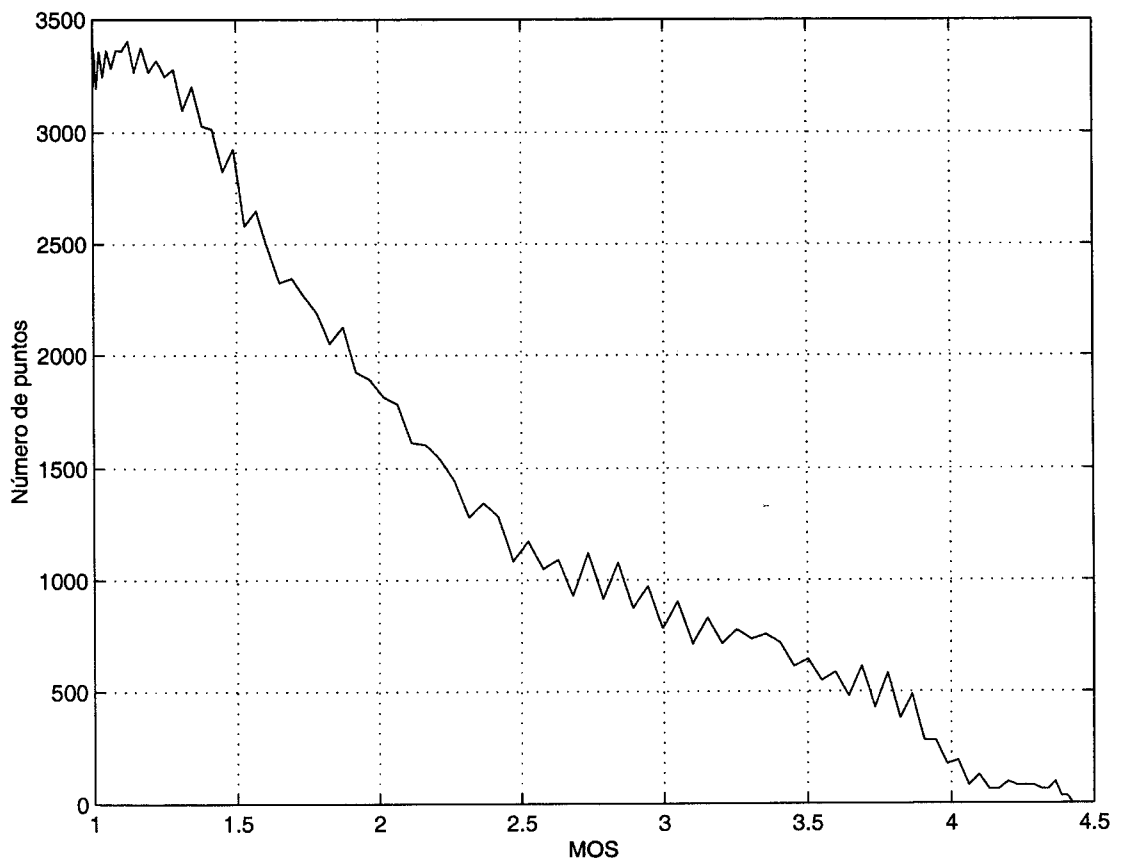


Figura 4.12: Distribución de puntos por MOS

puntos. Esto representa un 0,8 % de los puntos totales. La única forma de acceder a los puntos que cumplan con la MOS buscada sería la realización de una serie de tablas que proporcionasen tanto el valor del punto en cuestión como la MOS de dicho punto. De esta forma, y creado las tablas de índices apropiadas, se permite la selección directa de los puntos cuya MOS es la que se está buscando. Esta segunda alternativa se denominaría *evaluación previa indexada*.

La generación de las tablas de puntos y MOS se realizaría en la fase de inicialización del Gestor de Calidad de Servicio, ya que permanecen invariables. Los costes de dichos puntos serán evaluados dinámicamente en tiempo de ejecución. Esto no supone incremento significativo en el retardo del algoritmo pues el número de puntos a evaluar es mínimo (ver gráfico 4.12), y permite renovar las funciones de coste sin detener el funcionamiento del Gestor de Calidad de Servicio.

Sea **P** un vector que contiene información sobre el Codec, tramas por paquete, retardo, variación del mismo y tasa de pérdida de paquetes.

En primer lugar se debe generar una tabla con todos los puntos posibles, así como la MOS que obtienen dichos puntos. El pseudocódigo empleado para ello es:

```
i=0

PARA codec=1 HASTA C
  PARA fpp=1 HASTA FPP
    PARA delay=1 HASTA D
      PARA jitter=1 HASTA J
        PARA pl=1 HASTA PL
          P(i)=[ codec fpp delay jitter pl]
          MOS(i)=Modelo-E(P(i))
          i=i+1
        FIN_PARA
      FIN_PARA
    FIN_PARA
  FIN_PARA
FIN_PARA
```

A continuación se debe generar una tabla que se utilice como índice de la tabla anterior ordenada por MOS. La primera tarea será ordenar la tabla anterior por MOS y luego obtener la posición de dichos puntos. Esta nueva tabla de índices tendrá el mismo tamaño que la anterior, y se denominará **índice**.

Para eliminar puntos en dicha tabla se eliminan aquellos valores cuya MOS está por debajo de 1 (no utilizados en la práctica). Esta nueva tabla reducirá el número de puntos en un 80 %, ya que como se puede apreciar en la figura 4.12, la mayor parte de los puntos corresponden a valores muy bajos de MOS.

Para facilitar la búsqueda de una determinada MOS se puede crear un nuevo índice en el que se agrupan los puntos por valor de MOS, desde 5,0 hasta 1,0 con saltos de 0,1. La tabla de índice agrupado se denominará, **índice agrupado**).

Las tablas calculadas se realizan al iniciar el Gestor de Calidad de Servicio, ya que permanecen invariables. Los costes serán evaluados directamente por las funciones de coste y no almacenados en tablas. Esto no supone incremento de retardo pues el número de puntos a evaluar es mínimo (ver gráfico 4.12), y permite renovar las funciones de coste sin necesidad de detener el funcionamiento.

Se muestra a continuación un ejemplo de utilización del algoritmo para evaluar los puntos que tienen una MOS entre 3,8 y 4,0

```
MOS_inicial=3.8
MOS_final=4.0

PRIMERO = indice_agrupado[(5.0 - MOS_inicial) * 10]
ULTIMO = indice_agrupado[((5.0 -MOS_final) * 10)-1]

PARA i=PRIMERO HASTA ULTIMO
  PUNTO= P[indice[i]]
  SI (PUNTO ES VALIDO) ENTONCES
    EVALUAR COSTE PUNTO
    SI COSTE < COSTE_MIN ENTONCES
      OPTIMO=PUNTO
      COSTE_MIN=COSTE
    FIN_SI
  FIN_SI
FIN_PARA
```

Donde para que un punto sea válido el retardo total debe ser mayor que el retardo de los clientes incluyendo el empaquetado y la cola para compensar la variación del retardo. Así mismo, el codec de dicho punto debe figurar entre los codec posibles en dicha llamada. En el caso de la optimización total la variación del retardo está fijada, por lo que el retardo deberá ser tal que permita dicho valor de variación en la red para que el punto sea válido.

El número de puntos que se evalúan para esta MOS y con los datos del ejemplo se reducen a 1610 (aplicando las mejoras especificadas en la solución propuesta). Con tablas intermedias para simplificar el cálculo de los costes el tiempo que consume este algoritmo en condiciones similares a los de los algoritmos anteriores es de 13 ms, valor aceptable como retardo de establecimiento de llamada.

A este algoritmo se pueden aplicar un cierto número de mejoras que permiten disminuir todavía mas el número de puntos posibles:

- **Eliminación de los puntos sin calidad**

La aplicación directa del Modelo-E sobre todos los puntos posibles proporciona una distribución de la calidad como la mostrada en la figura 4.12. En ella se puede apreciar que

existe gran número de puntos cuya MOS es de uno. Es improbable que alguien solicite una llamada con la peor calidad por lo que dichos puntos pueden ser eliminados, reduciendo el número de puntos en un 48 %.

■ **Reducción del número de puntos de Variación del Retardo**

Debido a que la variación del retardo siempre será menor que el retardo total, la tabla que se puede formar con ambos parámetros será triangular. Si además se tiene en cuenta que el retardo total será menor de 500 ms, la suma de ambas debe ser menor que dicha cota por lo que la matriz triangular se reduce a la mitad, pues deberán cumplirse ambas condiciones:

- $D + J < 500$
- $J < D$

Esto reduce el número de puntos a considerar en un 25 % de los originales

Precálculo de la Variación del retardo Óptima

La MOS depende del retardo extremo a extremo. En este retardo se incluyen entre otros, el retardo de la red y la variación del retardo de la red. Para alcanzar dicho retardo existen varias posibilidades en función de los valores de retardo de red y variación del mismo . Con los ficheros de coste puede averiguarse cual de estas posibles combinaciones resulta menos costosa. Como siempre se trata de minimizar el coste, el resto se puede despreciar, eliminando una gran cantidad de puntos.

Se realizará una tabla en la que en función del retardo extremo a extremo, se ofrece el valor del retardo de red y de variación del mismo que minimiza el coste.

Con esta medida sólo se deberá tener en cuenta los puntos correspondientes a un cierto retardo extremo a extremo, por lo que el parámetro variación del retardo de red desaparece, quedando

$$N_{puntos} = C \cdot FPP \cdot D \cdot PL \tag{4.2}$$

Y con los datos utilizados anterioremente el número de puntos totales se reduce a :

$$N_{puntos} = 5 \cdot 32 \cdot 500 \cdot 10 = 800,000$$

Es decir un 4 % de los puntos originales. En este caso, la D se corresponde con el retardo extremo a extremo. Dicho retardo se puede descomponer como sigue:

$$D_{total} = D_{Clientes} + FPP \cdot T_{trama} + Variac.Retardo + D_{Red} \tag{4.3}$$

Donde $D_{Clientes}$ se corresponde con el retardo medio de los clientes debido a su implementación. En el Apéndice C de la presente memoria se ofrece una guía para su cálculo.

$$D_{Clientes} = (D_{Llamante} + D_{Llamado})/2 \tag{4.4}$$

El parámetro T_{trama} se corresponde con la duración de las tramas de un determinado codec. En la tabla 2.2 se puede obtener dicho valor.

- **Ajuste del número de tramas por paquete**

Tomando como requisito que el retardo total deberá ser <500 ms. Para codec con tramas de 10 ms el número de tramas por paquete posibles será de 32 (supuesto que el retardo de empaquetado es como máximo 320 ms), pero para codecs con tramas de 30 ms como los G.723 el número de tramas por paquete posibles será de 10. Con este ajuste se logra reducir el número de puntos posibles.

Esta solución ahorra tiempo de ejecución del algoritmo a costa de generar tablas con índices que permiten buscar sólo los puntos que cumplen una determinada MOS. Estas tablas deberán guardarse en memoria por lo que la necesidad de memoria RAM será el precio que se debe pagar a costa de ahorrar el tiempo de ejecución. Tomando puntos de 3 octetos todas las tablas ocupan aproximadamente 7 MB.

En el Apéndice B se resume el formato de las diferentes tablas utilizadas en el algoritmo.

4.7. Validación del Modelo

El modelo descrito ha sido implementado y verificado mediante un protocolo de pruebas, cuyos objetivos son:

- Verificar el funcionamiento del esquema en conjunto ante cualquier petición.
- Comprobar el funcionamiento de la señalización ante situaciones anómalas, lo que permite verificar la robustez del esquema.
- Comprobar la calidad de las conversaciones de forma cualitativa, para confirmar la validez de los resultados de la etapa de optimización.

El escenario donde se han realizado las pruebas aparece esquematizado en la figura 4.13. Se disponía de tres máquinas bajo sistema operativo Linux, ejecutando un Gatekeeper y unos clientes H.323 modificados para soportar señalización de QoS. Adicionalmente, dos máquinas (con Linux) ejecutaban las aplicaciones TRM y TPE. La colocación de cada proceso en una máquina distinta se hizo a fin de poder capturar con un analizador de red las tramas intercambiadas entre las distintas entidades.

El protocolo de pruebas diseñado distingue tres clases de pruebas diferentes:

- Pruebas de señalización e interfuncionamiento.
- Pruebas de robustez en la señalización.
- Pruebas cualitativas de calidad.

A continuación se describe con mayor detalle cada uno de los tres tipos anteriores.

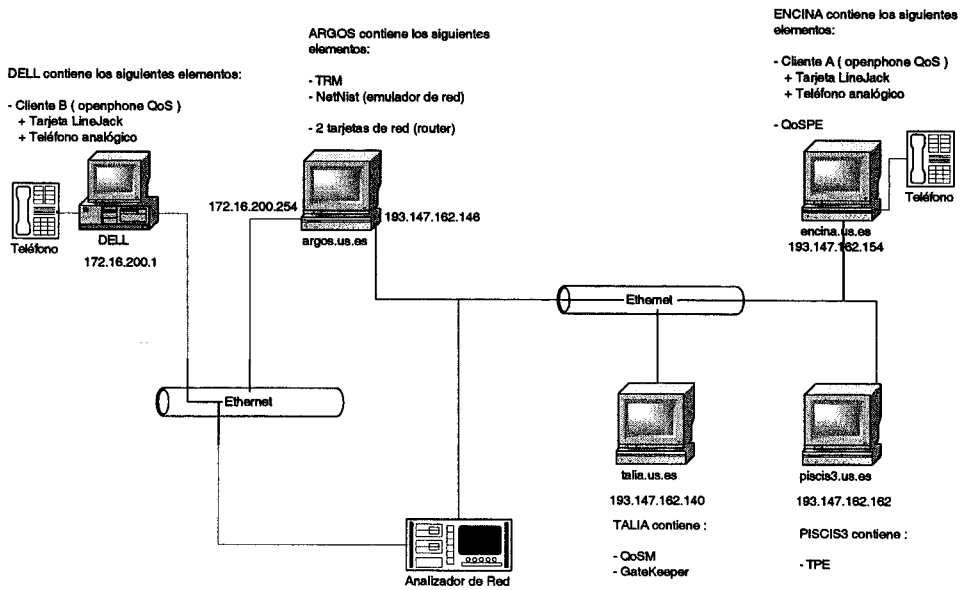


Figura 4.13: Banco de Pruebas

4.7.1. Pruebas Generales de Señalización e Interfuncionamiento

Esta batería de pruebas tiene como objetivo principal verificar el correcto funcionamiento del modelo completo en un escenario de explotación.

Para cada prueba realizada se almacenará un fichero con las tramas capturadas en la red, así como los ficheros de registro (*log*) de cada uno de los elementos implicados. El análisis de las tramas capturadas en la red será utilizado para verificar la estructura de las PDU realizadas, así como el valor de los campos que la forman. Además, dichas tramas permiten establecer una secuencia temporal de tramas que permite chequear el correcto funcionamiento de la señalización.

Para solucionar problemas internos a una entidad, además del análisis de las tramas se utilizaron ficheros de registro, que permiten comprobar el estado interno de dichas entidades, así como depurar problemas en tiempo de ejecución.

La prueba general de señalización e interfuncionamiento consiste en dos clientes QoS que efectúan una llamada con un nivel de calidad seleccionable. Todos los elementos del modelo se encuentran presentes y activos. La prueba se ha realizado dos veces, para permitir que el establecimiento y la finalización de la llamada se corresponda tanto al mismo como a distintos clientes.

Los items a confirmar en cada una de las entidades son:

- Cliente QoS:
 - Se realiza el envío del mensaje *QoSM request* al Gestor de Calidad correctamente.
 - Establece los parámetros de funcionamiento especificados en el mensaje *QoSM com-*

firm.

- Los valores en los campos de los mensajes *QoSM request* son correctos.
 - El comportamiento se corresponde con el definido en la interfaz gráfica.
- Gestor de Calidad (QoSM)
 - Los clientes se registran con sus datos correctamente.
 - La trama *QoS request* recibida en el ARQ es coherente.
 - El módulo de optimización funciona de forma adecuada.
 - Se realiza el envío de la trama *QoS confirm*.
 - En la trama *ACF* se especifica enrutado a través de Gatekeeper.
 - El envío y recepción de tramas al TRM es correcto.
 - La consulta al QoSPE es correcta
 - QoSPE
 - Envío y recepción correctos de los mensajes *QoSPE request, confirm y reject*
 - Plano de Transporte
 - Verificación de que los parámetros de QoS recibidos por el emulador de red NISTNet.
 - Comprobar la correcta adición y eliminación de las reglas necesarias en el cortafuegos cuando se establecen y liberan las conexiones de VoIP.
 - Funcionamiento correcto de las autorizaciones del TPE.

4.7.2. Pruebas de Robustez en la Señalización

Estas pruebas pretenden verificar el correcto funcionamiento del modelo ante situaciones reales no previstas en el diseño inicial.

Se checaron los siguientes items:

- Llamada entre dos clientes H.323 directa.
- Llamada entre dos clientes H.323 utilizando el GateKeeper.
- Llamada entre un cliente H.323 no QoS y otro cliente QoS utilizando un GK no QoS.
- Llamada entre un cliente H.323 no QoS y otro cliente QoS utilizando un GK QoS.
- Llamada entre un dos clientes H.323 no QoS mediante un GK QoS

Cada prueba era repetida tres veces, en las dos primeras el mismo cliente inicia y finaliza la llamada, y en la siguiente, un cliente inicia la llamada y el otro finaliza.

El examen de las tramas de la red permite comprobar los items anteriores.

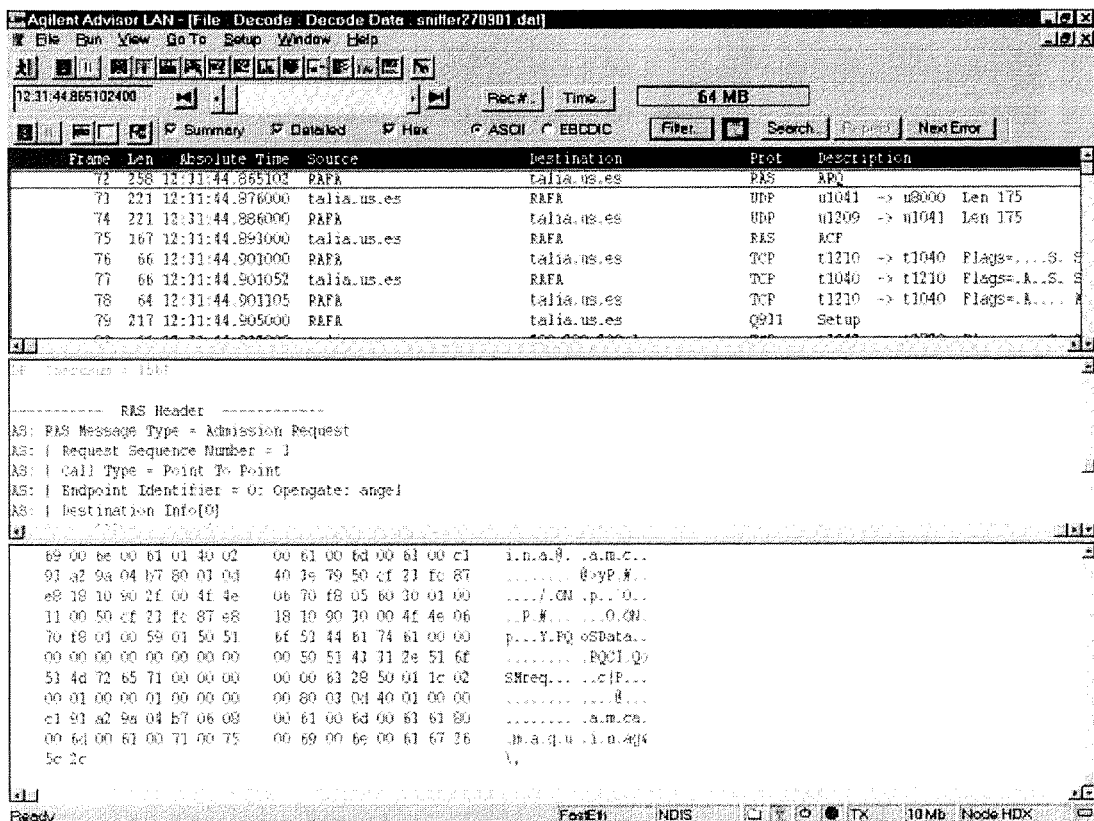


Figura 4.14: Fichero de captura de tramas

Pruebas cualitativas de Calidad

Se mantuvieron conversaciones con distintos niveles de QoS, comprobando de forma subjetiva la calidad.

En la figura 4.14 se puede observar una imagen de las tramas capturadas con el analizador. La primera trama se encuentra decodificada, pudiéndose observar que corresponde a un mensaje ARQ del canal RAS que incorpora una PDU de calidad (en concreto *QoSMrequest*)

Capítulo 5

Conclusiones y Líneas de avance

En este capítulo se resumen las conclusiones más importantes de la Tesis Doctoral, destacando sus principales contribuciones e indicando posibles líneas de continuación del trabajo realizado.

5.1. Conclusiones

En el capítulo 1 se expuso el entorno, las motivaciones y los objetivos de la presente Tesis Doctoral. En concreto:

- Se introdujeron las razones principales que han llevado al desarrollo de las redes de VoIP, entre ellas:
 - Mayor facilidad para introducir nuevos servicios.
 - Reducción de costes debido a integración de servicios en una sola infraestructura.
- Se resaltó como una ventaja competitiva de las redes de VoIP la posibilidad de proveer diversos niveles de calidad en la llamada.
- El motivo de la realización de la presente Tesis Doctoral es proporcionar un modelo para la provisión de llamadas en VoIP con un grado de calidad seleccionable por el usuario, consumiendo para ello los mínimos recursos en la red de transporte.
- El ámbito del estudio se reduce a redes de VoIP sin conexión a las actuales redes de telefonía RTC. Debido a que la calidad en las redes RTC se encuentra suficientemente estudiada, esta decisión no influye en las aportaciones de la Tesis Doctoral.

Una vez establecidos el ámbito y motivos de la Tesis, en el capítulo 2 se presentó el *Estado del Arte* de la calidad del servicio de telefonía. En particular:

- Se introdujeron los conceptos fundamentales de la calidad del servicio de telefonía clásico.
- Se llevó a cabo un estudio de los principales factores físicos que afectan a la calidad en las redes de telefonía tradicionales, particularizando dichos factores para el caso de las redes de VoIP.
- Se realizó una breve exposición de los principales métodos de medida de calidad en telefonía, haciendo especial énfasis en el método basado en parámetros conocido como Modelo-E. Dicho método será utilizado posteriormente en el esquema propuesto en el capítulo 4.

En el capítulo 3 se presentó el *Estado del Arte* de las redes de VoIP, enfocado desde el punto de vista del soporte de calidad de servicio. En concreto:

- Se presentaron los esquemas para el soporte de VoIP propuestos por la ITU-T (H.323) y el IETF (SIP).
- Se repasaron los mecanismos actuales para la provisión de calidad en las redes de transporte TCP/IP.
- Se introdujo el modelo de referencia para el soporte de calidad en las redes de VoIP propuesto por la ETSI en su grupo TIPHON. La señalización propuesta en este modelo servirá como punto de partida para el modelo propuesto por la presente Tesis Doctoral .

La selección de un determinado nivel de calidad en la llamada junto con una tarificación proporcional al mismo, son claves para el despliegue de las redes de VoIP en el mercado de las comunicaciones vocales.

El modelo propuesto para la provisión de un determinado nivel de calidad en una conversación minimizando el consumo de recursos en la red de transporte, se describe en el capítulo 4 de la memoria. En concreto, en dicho capítulo se abordan las siguientes cuestiones:

- Se introducen los requisitos de partida y la arquitectura de capas y planos de dicho modelo.
- Se proporcionan las especificaciones de diseño, entre las cuales destacan:
 - La ubicación de las entidades funcionales Cliente y Gestor de Calidad de Servicio en la entidades funcionales Terminal y GateKeeper del modelo H.323.
 - La encapsulación de las PDU de Calidad de Servicio intercambiadas entre cliente y QoS dentro del canal RAS existente. En concreto dentro del campo *GenericData* de *ARQ* y *ACF*.
 - La determinación del comienzo y el final de las llamadas mediante las PDU *OpenLogicalChannel*, *ACK* y *End SessionCommand* del canal H.245.
- Se describe el funcionamiento básico, así como el modelo de comportamiento de cada una de las entidades funcionales implicadas en el mismo.

- Se especifica el esquema utilizado para la fase de optimización, basado en el Modelo-E. Dicho esquema proporciona los parámetros de funcionamiento en terminales y en la red IP que minimizan el coste de los recursos de transporte (basándose en funciones de coste proporcionada por los proveedores de transporte).
- Por último, se introduce el protocolo de pruebas utilizado para la validación del modelo.

La aplicación de este modelo supone un avance en el despliegue de las redes de VoIP, pues:

- Para proveedores de servicio de VoIP supondría una reducción en el coste de provisión de la llamada. En el caso de que el proveedor de VoIP sea también el proveedor de los recursos de transporte, supondría una optimización en la utilización de los recursos, lo que permitiría soportar un mayor número de clientes en la misma infraestructura. En cualquier caso, una ventaja competitiva.
- Para el usuario final, supondría una reducción en el coste de la llamada, pues los proveedores que implementen este modelo tendrán menores costes, por lo que es de esperar una reducción en sus tarifas.

5.2. Líneas de avance

A continuación se presentan una serie de posibles vías de investigación futuras derivadas directamente del trabajo realizado. En concreto se propone:

- Realizar una verificación del Modelo-E para las redes de VoIP. Debido a las características de los nuevos terminales utilizados, y a la previsible influencia del precio en la opinión de calidad de los usuario, sería conveniente realizar un ajuste del factor de expectativa (A) del Modelo-E.
- Extender el modelo propuesto a múltiples dominios de transporte y de aplicación. Si bien se apunta en [37] cómo debe realizarse dicha extensión, es necesario definir los procedimientos de señalización, criterios de diseño y verificar experimentalmente el correcto funcionamiento.
- Proveer un cierto nivel de calidad a los clientes que no soporten la señalización propuesta en el presente modelo. Esto facilitaría la implantación del modelo propuesto en cualquier entorno a costa de perder el control sobre los parámetros de funcionamiento de los clientes.
- Realizar una selección óptima del proveedor de transporte en función de la calidad deseada y de las funciones de coste proporcionadas por los mismos. Esto permitiría a los proveedores del servicio de telefonía IP minimizar sus costes de forma dinámica en cada llamada.

Apéndice A

Definición de parámetros y primitivas

La señalización utilizada en el modelo se basa en la propuesta realizada por la ETSI en [37]. En dicho documento tan sólo se especifica verbalmente la función de las distintas primitivas, así como los elementos de información que éstas pueden transportar. Por ello es condición necesaria para poder utilizar dicha señalización realizar una especificación previa de las primitivas de calidad. Debido a que las recomendaciones asociadas a H.323 definen su primitivas y elementos de información en ASN.1, se utilizará dicho lenguaje para la especificación formal a realizar, con el fin de reutilizar la mayor parte posible de los parámetros.

En este apéndice se especifican en ASN.1 las primitivas del modelo propuesto, así como los parámetros y elementos de información que éstas utilizan. Posteriormente se describen los procedimientos utilizados para la señalización, así como la encapsulación de las primitivas de calidad sobre el canal RAS de H.225.0.

A.1. Grupos de parámetros

A continuación se especifican en lenguaje ASN.1 los parámetros utilizados por las primitivas de calidad del modelo propuesto.

A.1.1. QoS Service Class

Este parámetro describe el grado de calidad de servicio en la conversación en forma de MOS (valor entre 1 y 5). La sintaxis propuesta es la siguiente:

```
QoSClass ::= SEQUENCE {
    max    INTEGER (0..50),
    min    INTEGER (0..50)
}
```


Donde se emplea un rango en lugar de un valor fijo, con el fin de ofrecer una mayor flexibilidad y una correspondencia más exacta de los recursos de transporte reservados con la clase de servicio solicitada. Así, para especificar una MOS de 4 más menos 0,5, los valores serían 35 y 45.

A.1.2. Codec Type and Packetization

Este parámetro describe el tipo de códec (opcionalmente, una lista de códecs posibles) usado en la portadora, y la forma en que el flujo de medios es *empaquetado* en los datagramas IP. La sintaxis propuesta es la siguiente:

```

IMPORTS AudioCapability,
        VideoCapability,
        DataApplicationCapability,
        UserInputCapability
FROM MULTIMEDIA-SYSTEM-CONTROL; -- H.245 v.7

CodecTypePacketization ::= SEQUENCE OF CodecType

CodecType ::= SEQUENCE {
    codec          Codec,
    framesPerPacket INTEGER(1..256)
}

Codec ::= CHOICE {
    audioCodec      AudioCapability,
    videoCodec      VideoCapability,
    dataCodec       DataApplicationCapability,
    userCodec       UserInputCapability
}

```

Junto a cada códec, se especifica el máximo número de tramas por paquete que es capaz de generar. En [37] aun permanece bajo estudio la posibilidad de incluir nuevos campos en el tipo de datos *CodecType*.

A.1.3. Transport QoS Parameters

Este grupo de parámetros especifica las características del flujo de transporte que soporta el flujo de medios. La sintaxis propuesta es la siguiente:

```

TransportQoSParameters ::= SEQUENCE {
    controlQoS          BOOLEAN DEFAULT TRUE,
    fixedDelay          BOOLEAN DEFAULT FALSE,
    maxDelay            Delay,
    fixedMaxDelayVariation  BOOLEAN DEFAULT FALSE,
    maxPacketDelayVariation Delay,
    fixedPacketLoss     BOOLEAN DEFAULT FALSE,
    maxPacketLoss       PacketLoss
}

Delay ::= INTEGER(0..65535) -- milliseconds
PacketLoss ::= INTEGER(0..1000) -- parts per thousand

```

Los campos lógicos (booleanos) permiten que no se controle alguna de las características de QoS. De cualquier modo, los valores por defecto asignados a estos campos han sido elegidos para que indiquen control completo de QoS.

A.1.4. Traffic Descriptor

Este parámetro caracteriza las necesidades de recursos de un flujo de datos de aplicación (excluidas las necesidades del flujo de transporte). La sintaxis propuesta es la siguiente:

```

IMPORTS BandWidth FROM H323-MESSAGES; -- H.225.0 v.4

TrafficDescriptor ::= SEQUENCE {
    peakBit          BandWidth, -- in 100ths of bit
    maxPacketSize   PacketSize -- in 100ths of bit
}

PacketSize ::= INTEGER(0..4294967295)

```

Esta descripción del tráfico presupone que su tasa de bit es constante (CBR). Cuando se utiliza detección de actividad de voz (VAD), el tráfico generado no responde al modelo CBR, sino a una tasa de bit variable (VBR). Aún permanece bajo estudio en [37] la posibilidad de añadir campos adicionales para describir de forma más precisa flujos VBR.

A.1.5. Caller and Called IDs

Este parámetro especifica la identidad de las entidades llamante y llamada. La sintaxis propuesta es la siguiente:

```

IMPORTS AliasAddress FROM H323-MESSAGES; -- H.225.0 v.4

```

```
PartiesID ::= SEQUENCE {
    caller  AliasAddress,
    called  AliasAddress
}
```

Para facilitar las posibilidades de comunicación interdominio, se siguen las directrices del Anexo G de [24], de forma que ambos objetos *AliasAddress* contendrán una dirección de correo electrónico o bien un número público E.146.

A.1.6. Transport Addresses

Este grupo de parámetros especifica información de direccionamiento con el fin de definir el flujo de transporte que acarrea la portadora. La sintaxis propuesta es la siguiente:

```
IMPORTS TransportAddress FROM H323-MESSAGES; -- H.225.0 v.4

TransportAddresses ::= SEQUENCE {
    sendAddress      TransportAddress,
    rcvAddress       TransportAddress
}
```

Teniendo en cuenta que un *TransportAddress* especifica una dirección IP más un único número de puerto, no es factible para el QoS llevar a cabo agregación de recursos al señalar con el plano de transporte. La extensión de este tipo para el soporte de agregación en el plano de telefonía IP permanece en estudio en [37].

A.1.7. Application Data Transport Protocol

Este parámetro especifica el protocolo de transporte de datos de aplicación de la portadora. La sintaxis propuesta es la siguiente:

```
IMPORTS GenericIdentifier FROM H323-MESSAGES; -- H.225.0 v.4

ADTransportProtocol ::= GenericIdentifier
```

En forma de número entero, el identificador debe tomar el valor de un número de protocolo asignado por el IANA. En forma de *GloballyUniqueID*, debería tomar el valor de una cadena ASCII descriptiva (p. e. *RTP*).

A.1.8. Packet Transport Protocol

Este parámetro especifica el protocolo de transporte de paquetes. La sintaxis propuesta es la siguiente:

```
IMPORTS GenericIdentifier FROM H323-MESSAGES; -- H.225.0 v.4
```

```
PTProtocolIdentifier ::= GenericIdentifier
```

Este tipo de datos será interpretado de igual forma al tipo anterior.

A.1.9. QoS Policy

Este parámetro describe la política que determina el derecho de los usuarios a una clase de servicio. Aún está por definir en [37].

```
QoSPolicy ::= NULL -- TBD
```

A.1.10. QoS Mechanism

Este grupo de parámetros describe el mecanismo empleado en el plano de transporte. La sintaxis propuesta es la siguiente:

```
IMPORTS NonStandardParameter FROM H323-MESSAGES; -- H.225.0 v.4
```

```
QoSMechanism ::= SEQUENCE {  
    type           QoSMechType,  
    parameters     SEQUENCE OF NonStandardParameter,  
    authToken      Token  
}
```

```
QoSMechType ::= ENUMERATED {  
    none(0),  
    rsvp(1),  
    diffserv(2),  
    mpls(3)  
}
```

```
Token ::= NULL -- TBD
```

Los parámetros específicos a cada mecanismo y el formato del testigo de autorización aún han de ser definido en [37].

A.2. Primitivas de QoS

A continuación se especifican en ASN.1 las primitivas utilizadas en las distintas interfaces del modelo de calidad de TIPHON descrito en el capítulo 3. En [37] se describen verbalmente los parámetros que deben llevar la mayoría de las primitivas (otras se dejan como trabajos futuros).

A.2.1. Interfaz QC1

Este punto de referencia se ubica entre las entidades QoS_M de usuario y proveedor de servicio. Se definen tres primitivas:

QC1. QoS_Mreq La sintaxis propuesta es la siguiente:

```
QC1_QoSreq ::= SEQUENCE {
    qosClass                QoSClass,
    codecTypePacketization CodecTypePacketization,
    transportQoSParameters TransportQoSParameters
                           OPTIONAL,
    trafficDescriptor       TrafficDescriptor
                           OPTIONAL,
    transportAddresses       TransportAddresses,
    partiesID                PartiesID,
    applicationProtocol      ADProtocolIdentifier
                           OPTIONAL,
    transportProtocol        PTProtocolIdentifier
                           OPTIONAL
}
```

QC1. QoS_Mconf La sintaxis propuesta es la siguiente:

```
QC1_QoSconf ::= SEQUENCE
{
    qosClass                QoSClass,
    codecTypePacketization CodecTypePacketization,
    transportQoSParameters TransportQoSParameters
                           OPTIONAL,
    transportAddresses       TransportAddresses,
    applicationProtocol      ADProtocolIdentifier
                           OPTIONAL,
    transportProtocol        PTProtocolIdentifier
                           OPTIONAL,
    qosMechanism             QoSMechanism OPTIONAL
}
```

QC1. QoS_Mrej La sintaxis propuesta es la siguiente:

```
QC1_QoSMrej ::= SEQUENCE {
    rejectReason    QC1_RejectReason
}

QC1_RejectReason ::= ENUMERATION {
    undefinedReason (0),
    notQoS          (1),
    ...
}
```

Aún no han sido definidos códigos de rechazo específicos. En su lugar, se definen dos códigos genéricos:

- *undefinedReason*: para señalar errores inesperados.
- *notQoS*: para señalar motivos específicos de aplicación.

Estas primitivas se agrupan en un tipo de datos que representa un mensaje QC1. Su definición es como sigue:

```
QC1_QoSMdata ::= CHOICE {
    qc1_qosMreq      QC1_QoSMreq,
    qc1_qosMconf     QC1_QoSMconf,
    qc1_qosMrej      QC1_QoSMrej
}
```

La sintaxis de las primitivas **QC1. QoS_Mrelreq** y **QC1. QoS_Mrelconf** aún está por definir, ya que no se describen en [37] y no son necesarias para verificar el modelo propuesto en la presente Tesis Doctoral.

A.2.2. Interfaz QS4

Este punto de referencia se ubica entre el QoS_M y el QoS_{PE} de un proveedor de servicio. Se definen tres primitivas:

QS4. QoS_{PE}req La sintaxis propuesta es la siguiente:

```
QS4_QoSPEreq ::= SEQUENCE {
    qosClass          QoSClass,
    partiesID        PartiesID
}
```

QS4. QoSPEconf La sintaxis propuesta es la siguiente:

```

QS4_QoSPEconf ::= SEQUENCE {
    qosClass          QoSClass,
    partiesID        PartiesID
}

```

QS4. QoSPErej La sintaxis propuesta es la siguiente:

```

QS4_QoSPErej ::= SEQUENCE {
    rejectReason      QS4_RejectReason
}

QS4_RejectReason ::= ENUMERATED {
    undefinedReason (0),
    notQoS          (1),
    ...
}

```

Estas primitivas se agrupan en un tipo de datos que representa un mensaje QS4. Su definición es como sigue:

```

QS4_QoSMDdata ::= CHOICE {
    qs4_qosPEreq      QS4_QoSPEreq,
    qs4_qosPEconf     QS4_QoSPEconf,
    qs4_qosPErej      QS4_QoSPErej
}

```

A.2.3. Interfaz QT2

Este punto de referencia se ubica entre el QoSM de un proveedor de servicio, y el TRM de un operador de red. Se definen cinco primitivas:

QT2. TRMQreq La sintaxis propuesta es la siguiente:

```

QT2_TRMQreq ::= SEQUENCE {
    transportQoSParameters TransportQoSParameters,
    trafficDescriptor        TrafficDescriptor,
    transportAddresses        TransportAddresses,
    transportProtocol         PTProtocolIdentifier
                                OPTIONAL
}

```

QT2. TRMQconf La sintaxis propuesta es la siguiente:

```

QT2_TRMQconf ::= SEQUENCE {
    transportQoSParameters TransportQoSParameters,
    transportAddresses      TransportAddresses,
    transportProtocol       PTProtocolIdentifier
                           OPTIONAL,
    qosMechanism            QoSMechanism OPTIONAL
}

```

QT2. TRMQrej La sintaxis propuesta es la siguiente:

```

QT2_TRMQrej ::= SEQUENCE {
    rejectReason    QT2_RejectReason
}

QT2_RejectReason ::= ENUMERATED {
    undefinedReason (0),
    notQoS          (1),
    ...
}

```

QT2. TRMQrelreq El conjunto de parámetros que esta primitiva debería contener no ha sido definido por TIPHON aún. Teniendo en cuenta que esta primitiva es imprescindible para el procedimiento de señalización de QoS descrito en el apartado A.3, se ha proporcionado una definición que contiene la información necesaria para identificar de forma inequívoca un flujo de transporte: Las direcciones de ambos extremos, y el protocolo de transporte. La sintaxis propuesta es la siguiente:

```

QT2_TRMQrelreq ::= SEQUENCE {
    transportAddresses      TransportAddresses,
    transportProtocol       PTProtocolIdentifier
                           OPTIONAL
}

```

QT2. TRMQrelconf Al igual que en el caso de la primitiva anterior, se ha proporcionado una definición básica. La sintaxis propuesta es la siguiente:

```

QT2_TRMQrelconf ::= SEQUENCE {
    transportAddresses      TransportAddresses,
    transportProtocol       PTProtocolIdentifier
                           OPTIONAL
}

```


Estas primitivas se agrupan en un tipo de datos que representa un mensaje QT2. Su definición es como sigue:

```
QT2_TRMQdata ::= CHOICE {
    qt2_trmQreq      QT2_TRMQreq,
    qt2_trmQconf    QT2_TRMQconf,
    qt2_trmQrej     QT2_TRMQrej,
    qt2_trmQrelreq  QT2_TRMQrelreq,
    qt2_trmQrelconf QT2_TRMQrelconf
}
```

La sintaxis de la primitiva QT2. TRMQperfnofit está aún por definir en [37].

A.2.4. Interfaz QI4

Este punto de referencia se ubica entre las entidades TRM y TPE de un operador de red. Se definen tres primitivas:

QI4. PQreq La sintaxis propuesta es la siguiente:

```
QI4_PQreq ::= SEQUENCE {
    transportQoSParameters  TransportQoSParameters
                           OPTIONAL,
    trafficDescriptor       TrafficDescriptor
                           OPTIONAL,
    transportAddresses      TransportAddresses,
    transportProtocol       PTProtocolIdentifier
                           OPTIONAL,
    qosMechanism            QoSMechanism
                           OPTIONAL
}
```

QI4. PQconf La sintaxis propuesta es la siguiente:

```
QI4_PQconf ::= SEQUENCE {
    transportQoSParameters  TransportQoSParameters
                           OPTIONAL,
    trafficDescriptor       TrafficDescriptor
                           OPTIONAL,
    transportAddresses      TransportAddresses
                           OPTIONAL,
    qosMechanism            QoSMechanism
                           OPTIONAL
}
```

QI4. PQrej la sintaxis propuesta es la siguiente:

```

QI4_PQrej ::= SEQUENCE {
    rejectReason    QI4_RejectReason
}

QI4_RejectReason ::= ENUMERATED {
    undefinedReason (0),
    notQoS          (1),
    ...
}

```

Estas primitivas se agrupan en un tipo de datos que representa un mensaje QI4. Su definición es como sigue:

```

QI4_PQdata ::= CHOICE {
    qi4_pQreq      QI4_PQreq,
    qi4_pQconf     QI4_PQconf,
    qi4_pQrej      QI4_PQrej
}

```

A.3. Procedimientos de señalización

En esta sección se presenta de forma esquemática el procedimiento de señalización propuesto en el modelo. Las entidades funcionales involucradas, así como su relación con elementos H.323, se muestran en la figura A.1.

Ambos terminales proponen un conjunto de códecs al QoSM; éste elige un subconjunto común óptimo de códecs, e indica a los terminales que los empleen. Cuando se abre un canal lógico H.245, se establece una portadora con QoS controlada mediante señalización entre QoSM y TRM del dominio de transporte asociado. La señalización de QoS dentro del plano de transporte se lleva a cabo de la manera usual. Este procedimiento asume que tanto la señalización Q.931 como H.245 son enrutadas a través el Gatekeeper.

Los flujos de información se muestran en la figura A.2.

Los flujos contienen los parámetros recogidos en el cuadro A.1.

El grupo de parámetros *Transport QoS Parameters* transportado por las primitivas **QC1. QoS Mconf** contendrá, en su campo *maxDelay*, una estimación de la variación del retardo en la red, de forma que las aplicaciones puedan ajustar su cola. El resto de campos de este grupo de parámetros no contendrá información útil.

El grupo *CodecTypePacketization* transportado por las primitivas **QC1. QoS Mconf** indica a los terminales la lista óptima de codecs a emplear. En el caso del terminal llamante, el QoSM

Primitive	Parameters
QC1. QoSReq	QoS Service Class Codec Type and Packetization Application Data Transport Protocol Packet Transport Protocol
QS4. QoSPEreq	QoS Service Class Caller & Called IDs
QS4. QoSPEconf	QoS Service Class Caller & Called IDs
QC1. QoSMconf	QoS Service Class Codec Type and Packetization (parcialmente optimizado) Application Data Transport Protocol Packet Transport Protocol Transport QoS Parameters
QC1. QoSReq	QoS Service Class Codec Type and Packetization Application Data Transport Protocol Packet Transport Protocol
QC1. QoSMconf	QoS Service Class Codec Type and Packetization (completamente optimizado) Application Data Transport Protocol Packet Transport Protocol Transport QoS Parameters
QT2. TRMReq	Transport QoS Parameters Traffic Descriptor Transport Addresses Packet Transport Protocol
NOTA: La señalización de plano de transporte tiene lugar aquí	
QT2. TRMConf	Transport QoS Parameters Traffic Descriptor Transport Addresses Packet Transport Protocol

Cuadro A.1: Parámetros transportados por las primitivas de QoS

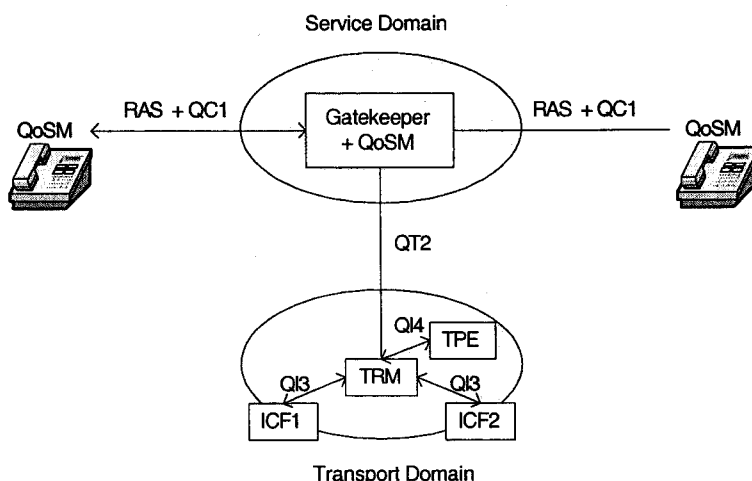


Figura A.1: Elementos funcionales involucrados en el establecimiento de llamada

no dispone aún de toda la información, y por tanto esta optimización es parcial. En el caso del llamado, la optimización es completa.

No es necesaria la definición de primitivas de rechazo de interfaz QC1. En caso de que el establecimiento de llamada sea denegado por motivos de QoS, se detendrá el proceso rechazando la admisión de llamada a nivel H.225.0 RAS, es decir, mediante el envío de una primitiva RAS *ARJ*.

La liberación de los recursos de transporte es solicitada de forma implícita por el QoS cuando se cierra un canal lógico o la llamada finaliza, de forma que no son necesarias tampoco primitivas de liberación.

A.3.1. Encapsulado de mensajes de QoS

Las primitivas de interfaz QC1 se empaquetan dentro de objetos *genericData*, pertenecientes a la señalización H.225.0 RAS. El elemento identificador de un *GenericData* que transporte mensajes de QoS debe permitir conocer esta circunstancia. El identificador propuesto será un objeto de tipo *GloballyUniqueID*, conteniendo la siguiente cadena ASCII: "TIPHON_QOS_QC1_".

Un objeto *GenericData* puede contener uno o más elementos *EnumeratedParameter*. Cada uno de ellos contiene un identificador y, opcionalmente, un elemento de datos del tipo *Content*, descrito en la versión 4 de la Recomendación H.323 como sigue:

```
Content ::= CHOICE
{
    raw          OCTET STRING,
    text        IA5String,
```

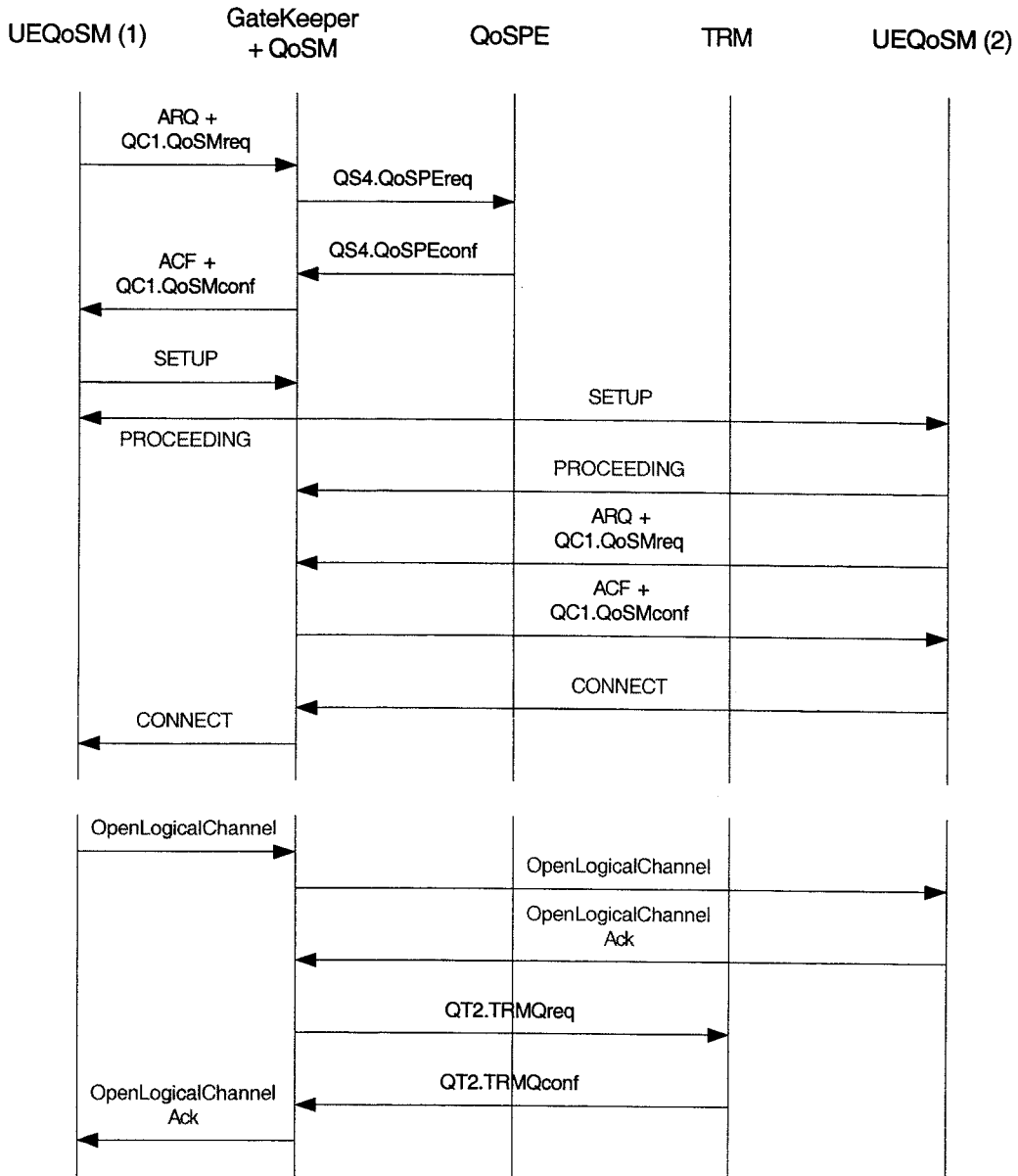


Figura A.2: Flujos de información

```

    unicode      BMPString,
    bool         BOOLEAN,
    number8     INTEGER (0..255),
    number16    INTEGER (0..65535),
    number32    INTEGER (0..4294967295),
    id          GenericIdentifier,
    alias       AliasAddress,
    transport   TransportAddress,

    compound    SEQUENCE (SIZE (1..512))
               OF EnumeratedParameter,
    nested     SEQUENCE (SIZE (1..16))
               OF GenericData,
    ...
}

```

A fin de no modificar la actual definición del tipo de datos *Content* definido en la Recomendación H.225.0, se proponen que las primitivas de QoS sean transportadas como mensajes de QoS serializados a cadenas de octetos (content tipo *raw*) utilizando la codificación PER¹ de ASN.1 como sintaxis de transferencia. El mensaje de QoS se define de la siguiente forma:

```

QoSMessage ::= CHOICE {
    qc1Data      QC1_QoSMDdata,
    qs4Data      QS4_QoSPEdata,
    qt2Data      QT2_TRMQdata,
    qi4Data      QI4_PQdata,
    ...
}

```

El campo identificador de cada objeto *EnumeratedParameter* debería ser un *GloballyUniqueID* conteniendo una cadena ASCII que identifique el tipo de primitiva transportada, p. e. "QC1._QoSMreq_...". El objeto de esta medida es facilitar la depuración, permitiendo identificar a las tramas capturas en la red de forma simple.

A.3.2. Especificación de las PDU de QoS

Para que dos entidades puedan intercambiar señalización de QoS, debe definirse un protocolo. Las unidades de datos (PDU) enviadas contendrán primitivas de QoS, así como posiblemente información adicional de contexto.

¹Packet Encoding Rules

Dada la naturaleza de la señalización de QoS, orientada a un intercambio simple petición – respuesta, es natural que ese protocolo tome una forma cliente – servidor, basado en dos PDUs, una de solicitud y otra de respuesta. La información de contexto que transporten puede incluir:

- Un número de secuencia, para relacionar una primitiva de confirmación o rechazo con la solicitud a la que responde.
- Un campo de identificación referido al remitente del mensaje.
- Un campo para indicar al cliente que redirija sus peticiones a un servidor más apropiado.
- Un código de rechazo, en caso de que la solicitud no pueda ser atendida por razones ajenas a la gestión de QoS.

Se han propuesto dos PDUs básicas para el transporte de primitivas de QoS a través de cualquier interfaz, cuya sintaxis es la siguiente:

```

IMPORTS RequestSeqNum,
        GenericIdentifier,
        GenericData,
        TransportAddress
FROM H323-MESSAGES; -- H.225.0 v.4

QoS_PDU ::= CHOICE {
    request          QoS_Request,
    response         QoS_Response
}

QoS_Request ::= SEQUENCE {
    sequenceNumber  RequestSeqNum,
    senderID        GenericIdentifier,
    qosData         SEQUENCE OF GenericData,
    ...
}

QoS_Response ::= SEQUENCE {
    sequenceNumber  RequestSeqNum,
    senderID        GenericIdentifier,
    qosData         SEQUENCE OF GenericData,
    ... ,
    redirectAddress TransportAddress
                    OPTIONAL,
    rejectReason    QoS_RejectReason
                    OPTIONAL
}

```

```

}

QoS_RejectReason ::= ENUMERATED {
    undefinedReason      (0),
    notQoS                (1),
    ...
}

```

El empaquetado de mensajes de QoS dentro de los objetos *GenericData* transportados por las PDUs se hará como se indica en el apartado A.3.1, pero utilizando los identificadores adecuados. Por ejemplo, si un objeto *GenericData* transporta primitivas QS4, su identificador deberá ser un *GloballyUniqueID* cuyo valor sea la cadena ASCII de longitud 16 “TIPHON_QOS_QS4_”.

Estas PDUs son susceptibles de ser enviadas tal como están definidas sobre un protocolo de transporte como UDP o TCP, para permitir a las aplicaciones de QoS interoperar siguiendo las definiciones y procedimientos especificados. En el modelo propuesto se ha empleado UDP como mecanismo de transporte para las interfaces QS4, QT2 y QI4.

A.3.3. Procedimiento auxiliar de Registro

El QoSM puede necesitar de información adicional, asociada al terminal, para determinar en su algoritmo de optimización los parámetros de funcionamiento de los clientes y la red IP. Por ello, el terminal al registrarse debe indicar la siguiente información:

- El retardo de la red de acceso a la cual está conectado.
- El retardo de proceso del terminal

Para ello, se propone que la gestión de QoS TIPHON sea negociada al registrarse el terminal, haciendo uso de las capacidades de negociación de características no estándares de la señalización RAS. Al registrarse, el terminal añadirá a la lista de características deseadas (objeto *desiredFeatures* dentro del juego de características – *featureSet* – de la primitiva RRQ), un objeto *GenericData* con el identificador “TIPHON_QOS_”.

Este objeto debe contener dos parámetros, ambos del tipo entero de 16 bits (*int16*). El primero de ellos indicará el retardo de la red de acceso del terminal (en milisegundos), y el segundo indicará el retardo de proceso del propio terminal (en milisegundos de nuevo). El valor de los identificadores de ambos parámetros es irrelevante, pero con el fin de facilitar la depuración, se propone que ambos sean objetos *GloballyUniqueID* con los valores “NET_DELAY” y “TERM_DELAY”, respectivamente.

Aquel Gatekeeper que soporte el procedimiento descrito en esta memoria, en caso de aceptar la solicitud de registro, deberá responder con un mensaje RCF del canal RAS que incluya en la lista de capacidades soportadas (objeto *supportedFeatures* dentro de su juego de características) un objeto *GenericData* cuyo identificador sea “TIPHON_QOS_”. No se requiere que éste transporte ninguna información adicional.

Apéndice B

Diseño de las aplicaciones

En el presente capítulo se realizará una descripción de las aplicaciones realizadas para implementar el modelo propuesto en el capítulo 4.

Para cada una de ellas se realizará un doble análisis:

- Estático, describiendo la jerarquía de tipos que forma la estructura de la aplicación.
- Dinámico, analizando los procesos relevantes para la negociación de QoS: registro de terminal en Gatekeeper, y establecimiento de llamada.

Así mismo en el análisis del QoSM se especificarán los formatos de las tablas utilizadas en el algoritmo de optimización realizado.

B.1. Cliente de QoS

El cliente de voz es una modificación realizada al cliente de libre distribución OpenPhone, código que acompaña a la librería OpenH323. La descripción del funcionamiento de la aplicación se ha centrado en aquellas partes relacionadas con la negociación de la QoS.

B.1.1. Análisis estático

La jerarquía de clases del sistema se muestra en la figura B.1. Los tipos definidos por la aplicación (*OpenPhoneConnection* y *OpenPhoneEndPoint*) sobrecargan algunas funciones virtuales para sacar por pantalla mensajes de estado en los pasos más importantes del proceso de una llamada. La verdadera funcionalidad del cliente está soportada por el resto de tipos, definidos en la librería OpenH323.

La implementación del terminal gira en torno al tipo *EndPoint*. Un terminal estará representado en general por una única instancia de este tipo. Este objeto almacena información sobre:

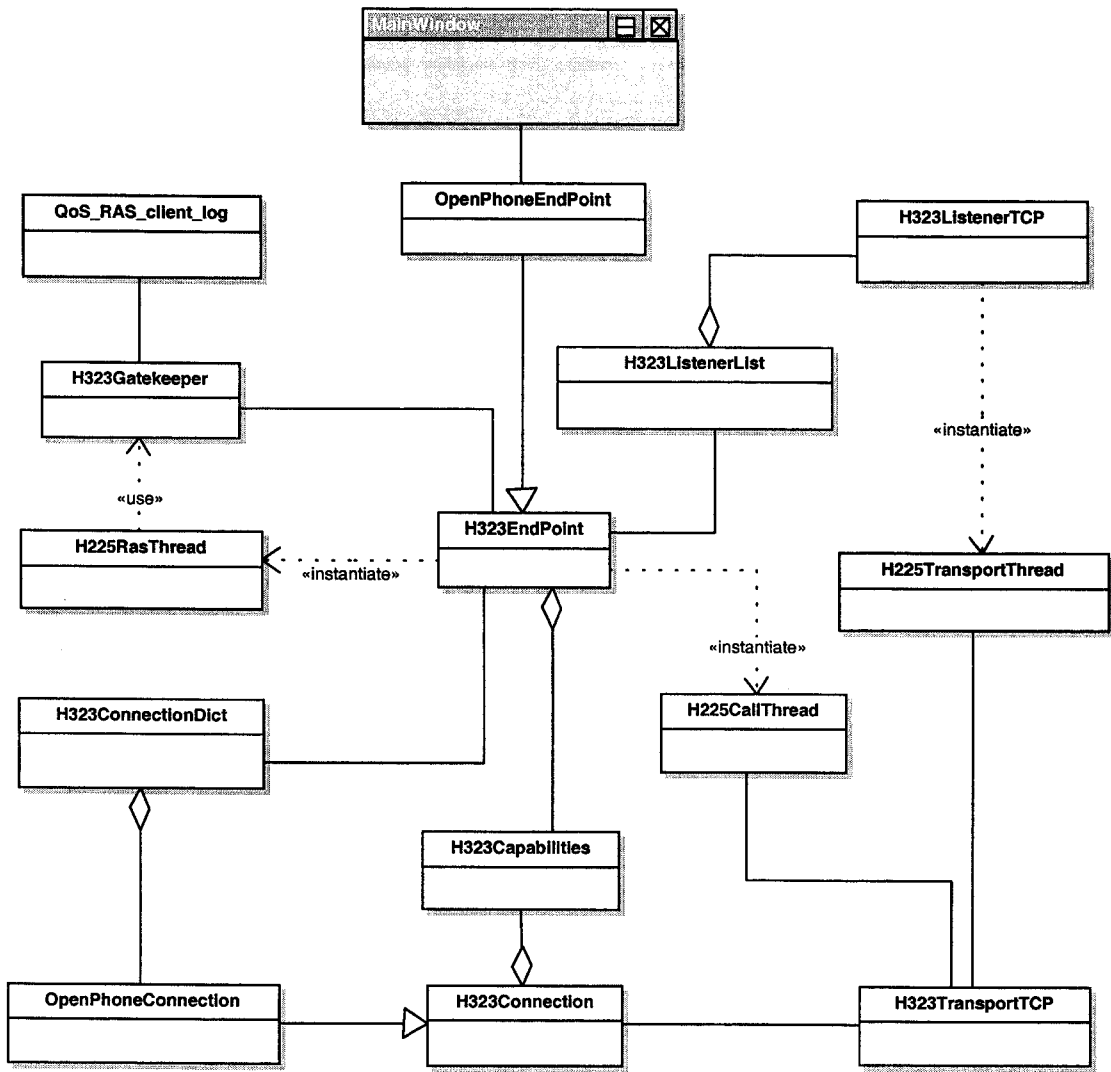


Figura B.1: Jerarquía del cliente de voz

- El Gatekeeper en el que el terminal está registrado. La conexión RAS con el Gatekeeper está gestionada por un objeto del tipo *H323Gatekeeper*. Se dedica un hilo de servicio (objeto de tipo *H225RasThread*) para atender y procesar los mensajes recibidos por este canal.
El tipo *H323Gatekeeper* ha sido modificado para soporte de QoS. Se ha añadido una lista de peticiones (objeto *QoS_RAS_client_log_list*). Cada elemento de la lista contiene la información de QoS transportada por la primitiva **QC1.QoSConf** incluida en cada mensaje *ACF* recibido en el canal RAS. La información contenida en esta lista se utiliza para establecer los códecs disponibles para la negociación H.245 (véase el apartado B.1.2).
- Las conexiones activas. Una llamada activa es gestionada mediante tres elementos:
 1. Un objeto de transporte (objeto *H323TransportTCP*), que controla el canal físico de señalización.
 2. Un manejador de conexión (objeto *H323Connection*), que gestiona el protocolo H.225.0 CS (señalización Q.931).
 3. Un hilo de servicio que presta su tiempo de procesador para la gestión de la conexión. En el caso de las llamadas entrantes, este hilo es del tipo *H225TransportThread*; para las llamadas salientes, es del tipo *H225CallThread*.
- Los sockets activos en espera de conexiones H.225 CS. Cada socket es monitorizado por un hilo. Por cada conexión TCP entrante se abre una nueva conexión H.225.0 CS.
- Las capacidades del terminal. Éstas identifican los códecs que un terminal es capaz de manejar. Cada objeto *Connection* hace una copia de estas capacidades, de forma que alterando esa copia sea posible modificar el conjunto de las mismas que será usado en cada conexión.

Para añadir soporte de QoS, ha sido necesario definir el tipo *MyH323Gatekeeper*, derivado de *H323Gatekeeper*, que sobrecarga las funciones necesarias para interceptar el envío y recepción de las primitivas RAS *RRQ*, *ARQ* y *ACF*, y manipular la información de QoS que éstas transportan.

El soporte de QoS en el cliente se completa con la adición de un nuevo método al objeto *MyH323Connection*. Esta función, *SetOptimumCapabilities*, tiene la misión de reordenar los códecs que serán empleados en la negociación H.245 según el criterio expresado por el QoSM en la primitiva **QC1.QoSConf**, como se verá más adelante.

Por último, se ha definido un tipo adicional, *CodecTag*, que simplifica la manipulación de los objetos *CodecType* devueltos por el QoSM extrayendo de ellos la mínima información necesaria: identificador de tipo (audio, vídeo, etc) y subtipo, y tramas por paquete.

B.1.2. Análisis dinámico

Establecimiento de llamada

El proceso de establecimiento de llamada con origen en el terminal se recoge en la figura B.3. En este proceso, se han distinguido cuatro partes:

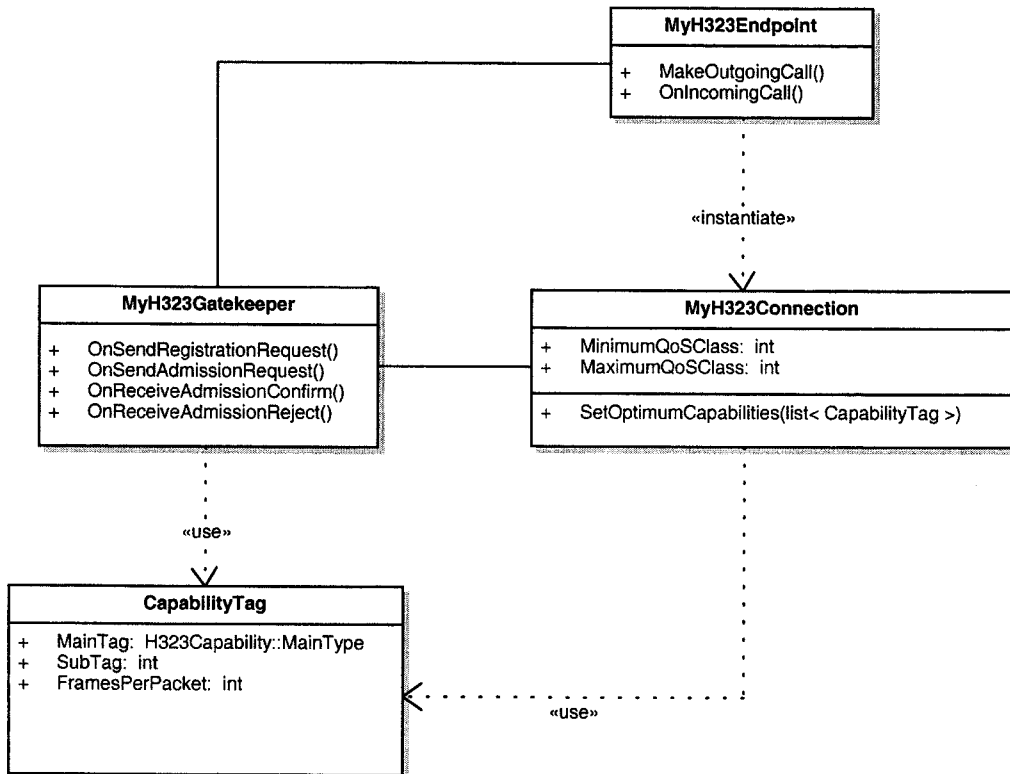


Figura B.2: Jerarquía simplificada del cliente de voz

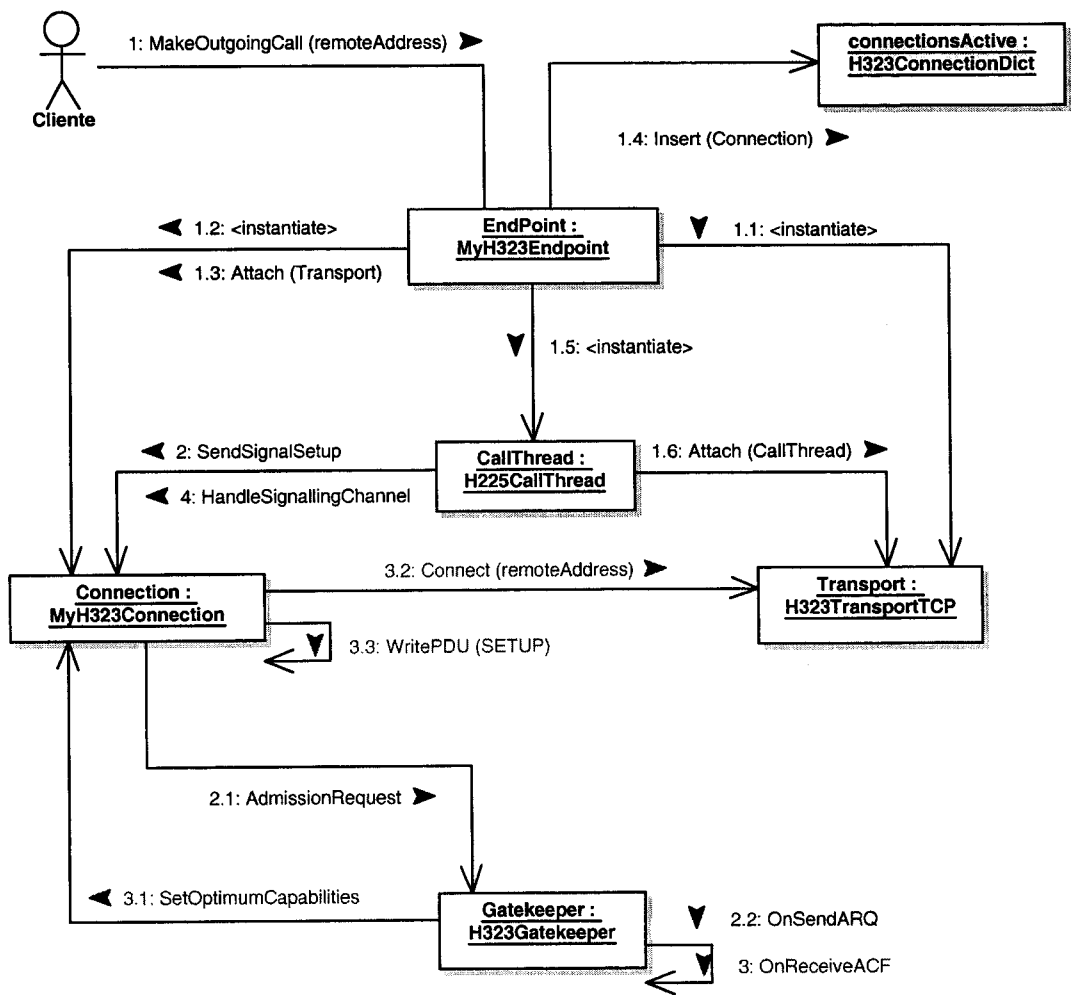


Figura B.3: Establecimiento de llamada

1. El usuario especifica que se desea iniciar una llamada. Se instancian los objetos necesarios para gestionar la conexión: transporte, conexión e hilo de servicio.
2. Se ejecuta el cuerpo del hilo. Éste invoca la función *SendSignalSetup*, comenzando la etapa de conexión. Antes de abrir el canal TCP, se solicita al Gatekeeper permiso para realizar la llamada, a través de la función *AdmissionRequest*. Ésta envía el mensaje ARQ del canal RAS, conteniendo una primitiva QC1. QoSMreq, y espera la respuesta.
3. Al recibirse el mensaje de confirmación (*ACF*), con la primitiva de calidad de servicio QC1. QoSMconf encapsulada, se extrae de ella la información de calidad de servicio. La función *OnReceiveAdmissionConfirm* establece el orden de preferencia de uso de los códecs, invocando al método *SetOptimumCapabilities* del objeto *MyH323Connection* responsable de la conexión.

Los códecs que se pueden emplear en la conexión se reordenan de acuerdo a lo especificado por el QoSM. La tasa de tramas por paquete para cada códec se establece a lo indicado. Si existe algún códec que no aparezca en la lista de códecs óptimos, es excluido de la negociación H.245.

4. Una vez recibida la autorización del Gatekeeper se abre la conexión con el extremo remoto, primero a nivel TCP y seguidamente a nivel H.225 mediante el envío de un mensaje *SETUP*. Se devuelve el control al hilo de servicio, que entra en el bucle indefinido de gestión del canal.

Recepción de llamada

El proceso de recepción de llamada se recoge en la figura B.4. En este proceso, se han distinguido de nuevo cuatro partes:

1. Al ser construido, el terminal instancia un socket de espera (*H323ListenerTCP*), y se registra en el Gatekeeper, indicando el puerto en el que escucha. En el proceso de registro con el Gatekeeper, le comunica información de QoS estática asociada al terminal (ver apartado A.3.3).
2. Al recibir una solicitud de conexión, se crea un nuevo hilo de servicio y se atiende la conexión entrante. El hilo utiliza la función *HandleFirstSignallingChannelPDU* del transporte para esperar la llegada de la primera PDU. Éste primer mensaje es distinto de los demás, pues si bien durante el resto de la vida de la conexión no hay un tiempo máximo de espera, el primer mensaje ha de llegar antes de quince segundos.
3. Al recibir la primera PDU, el transporte crea un objeto *H323Connection* y lo utiliza para procesar la PDU recibida. Como primer paso se emite un mensaje *PROCEEDING* de vuelta al otro extremo de la comunicación, y a continuación se solicita permiso al Gatekeeper para establecer la llamada. La función *OnSendAdmissionRequest* añade la información de QoS pertinente.

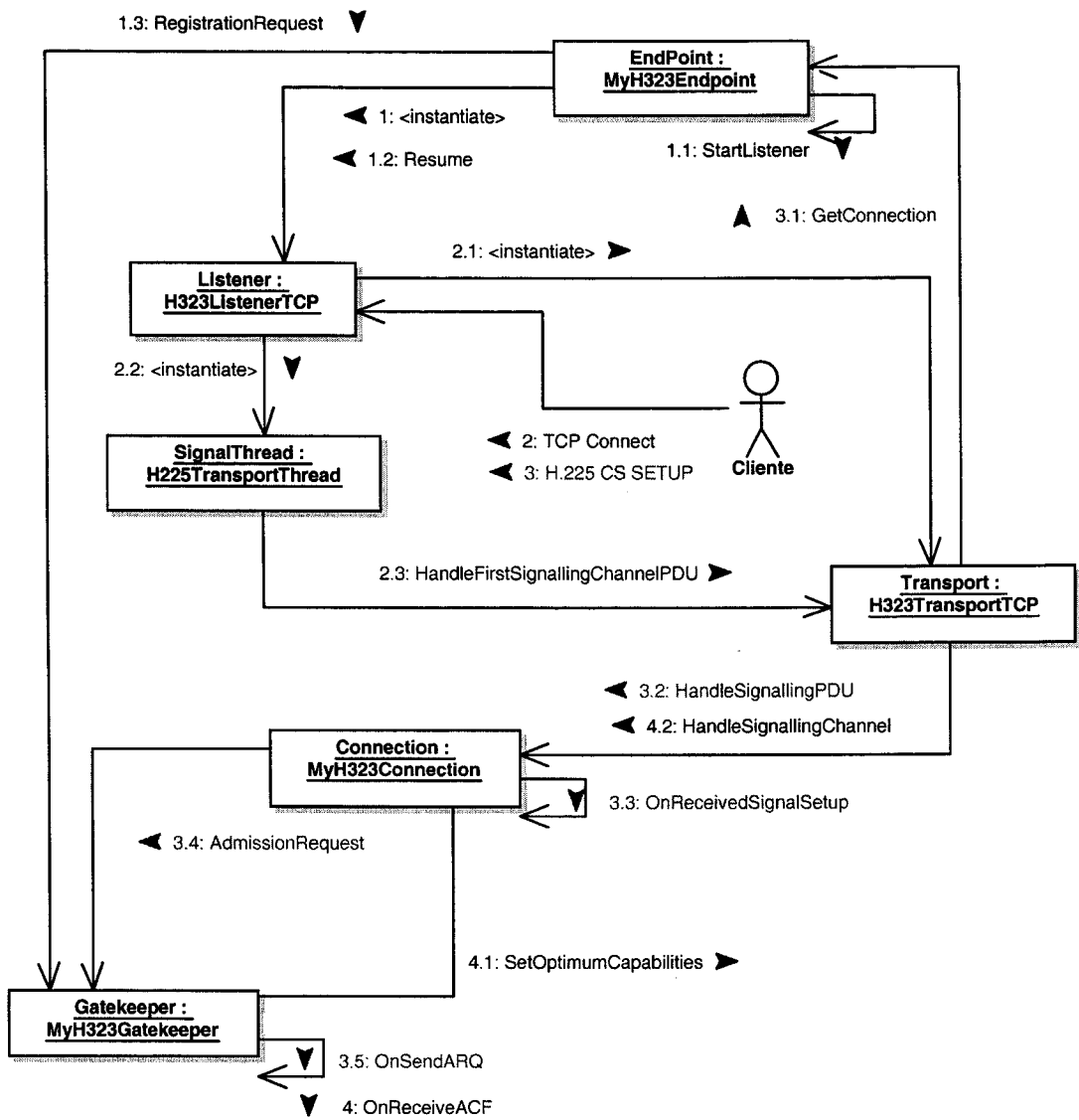


Figura B.4: Recepción de llamada

- Al recibirse la PDU de confirmación *ACF* por el canal RAS, se extrae de ella la información de calidad de servicio, y se realiza el mismo proceso que en el tercer paso de la sección anterior.

Señalización de interfaz QC1

El establecimiento de llamada, tanto de forma activa como pasiva, siempre pasa por un mismo punto: la solicitud de admisión al Gatekeeper. Es en este punto donde se introduce en el esquema la señalización de interfaz QC1. En la figura B.5 se ilustra este procedimiento.

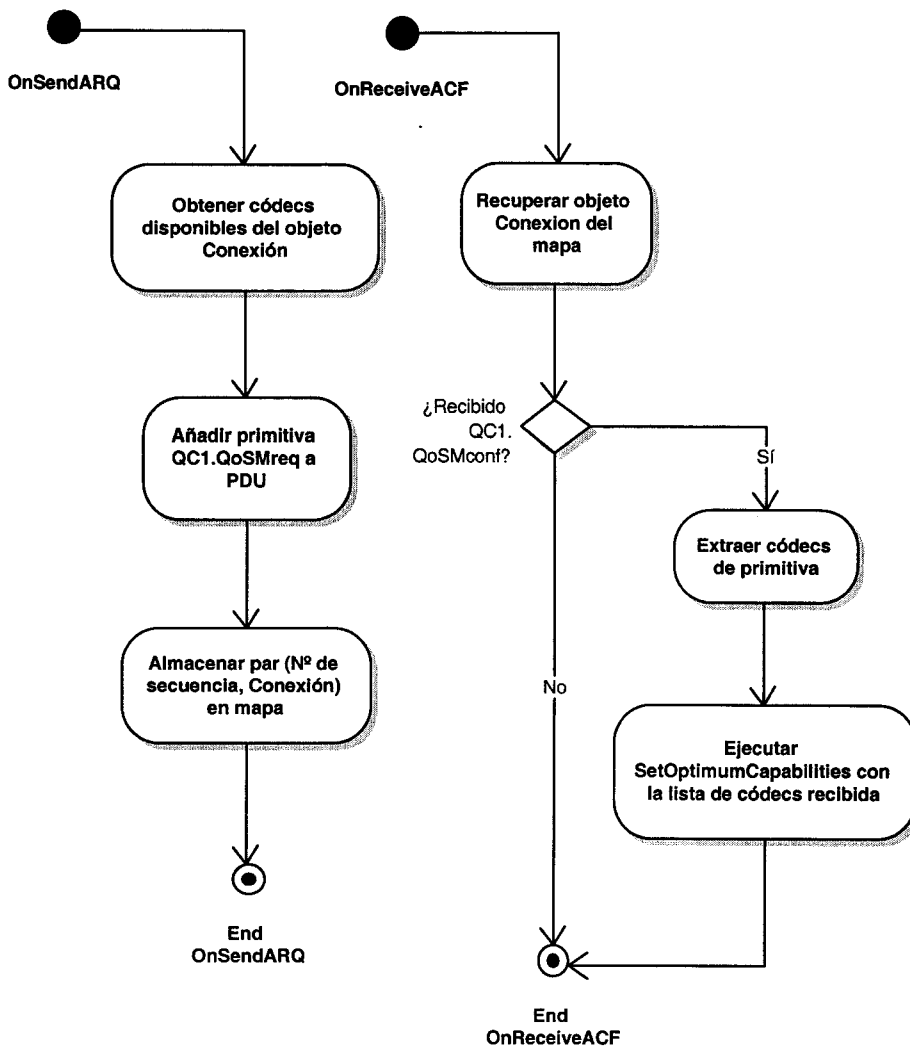


Figura B.5: Intercambio de mensajes en el interfaz QC1

El objeto MyH33Gatekeeper mantiene una tabla, donde existe una entrada por cada mensaje

ARQ enviada y que aún no ha tenido respuesta. En esa tabla, se asocian el número de secuencia de la PDU enviada con una referencia al objeto *MyH323Connection* que solicitó el envío.

Al recibir un *ACF* del canal RAS, se verifica que su número de secuencia corresponde al de alguna de las peticiones que permanecían en espera de respuesta. De ser así, se recupera la referencia al objeto *MyH323Connection*, se extrae la información de QoS contenida en la PDU, y se invoca a *SetOptimumCapabilities*, que reordena los códecs a usar y selecciona el número de tramas por paquete.

Aparte del conjunto de códecs óptimo a utilizar, el QoSM envía también una estimación de la variación del retardo media que se podrá producir en la red, información que se utiliza para dimensionar la cola para compensarlo en la conexión.

B.2. Servidor QoSM

El servidor QoSM y el Gatekeeper H.323 han sido integrados en una única entidad, que realiza las tareas propias de ambas. Como base para la aplicación, se utilizó un Gatekeeper ya existente de libre distribución, *opengate*, y se le añadió funcionalidad de QoSM. Así mismo el QoSM incluye el módulo de optimización, que será descrito brevemente en este apartado.

B.2.1. Análisis estático: GateKeeper

La figura B.6 muestra la estructura de clases del servidor. Para facilitar el análisis se ha dividido el conjunto de clases en tres bloques, que se abordarán por separado.

Núcleo

El bloque al que se ha denominado “núcleo” es el formado por la clase principal del programa, *OpenGate*, y las clases que representan y gestionan la lista de terminales registrados y de llamadas enrutadas, gestionadas ambas por objetos del tipo *Environ*.

Cada elemento de la tabla de terminales registrados es un objeto de tipo *EndPoint*. Así mismo, cada elemento de la tabla de llamadas es un instancia de la clase *CallDetails*. Con el fin de añadir soporte para el procedimiento de señalización de QoS descrito, se ha definido un paradigma de eventos, donde la recepción de ciertas PDUs genera un evento que es redirigido a un conjunto de objetos de manejo (“Callbacks”). Cada instancia de *EndPoint* o *CallDetails* guarda un conjunto de referencias a objetos de manejo, herederos de las clases *EndpointCallback* y *RASCallback*, respectivamente.

Estos objetos son usados para procesar la información genérica que puedan contener las primitivas recibidas por el Gatekeeper. La clase *EndpointCallback* define un conjunto de funciones virtuales puras relacionadas con el procedimiento de registro de un terminal: un juego de funciones a invocar al recibir cierto tipo de primitivas (por ejemplo, al recibirse un *RRQ*, un *DRQ*, etc), y otro a invocar antes de enviar cierto tipo de primitivas (por ejemplo, *RCF*, *RRJ*, etc). Así mismo,

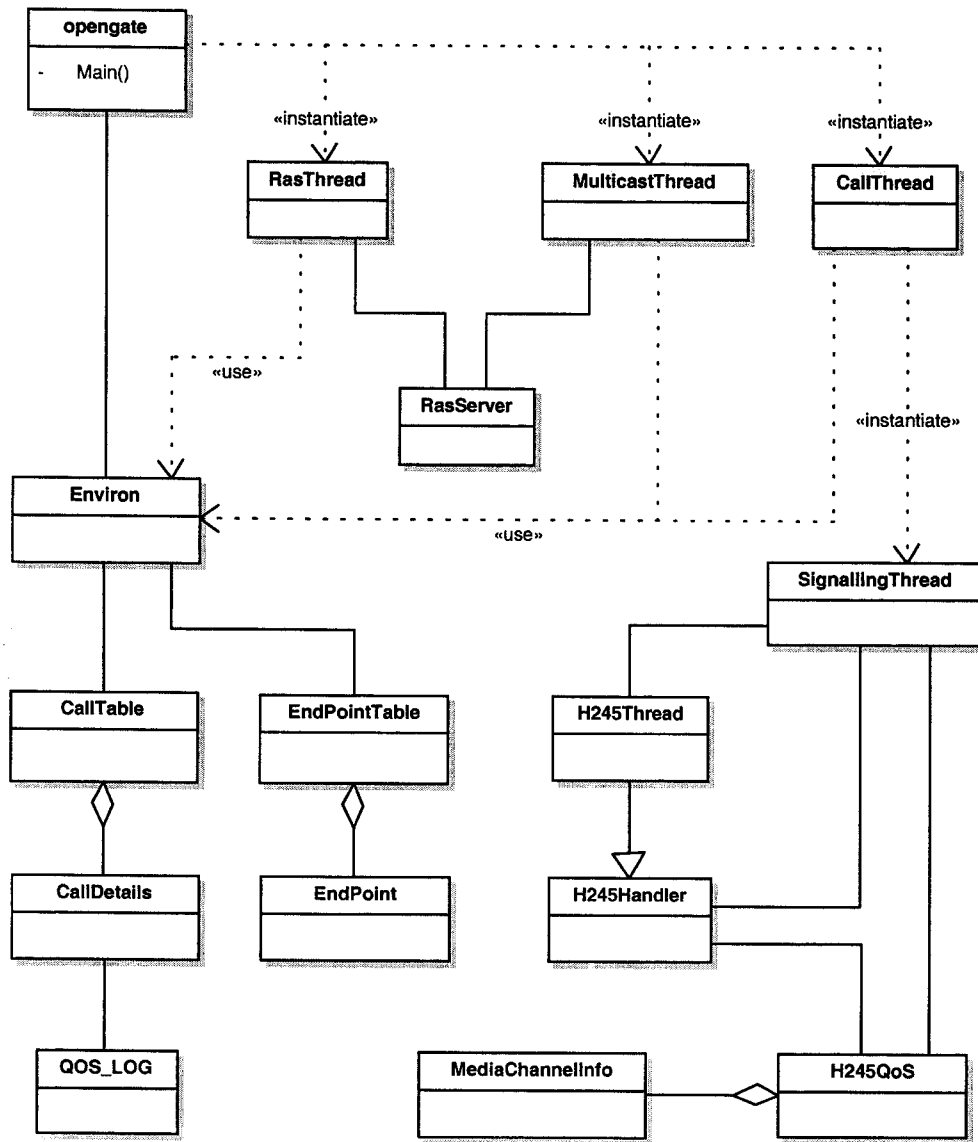


Figura B.6: Jerarquía de clases del Gatekeeper – QoSM

la clase *RASCallback* define dos juegos similares de funciones, a ser invocadas al recibir o enviar cierto tipo de PDUs RAS (*ARQ*, *BRQ*, *ARJ*, etc).

Gestión RAS

El segundo bloque lo componen las clases que se encargan de la señalización RAS. Son los tipos *RasThread*, *MulticastThread* y *RasServer*. Éste último es el tipo encargado de la gestión del canal; los dos primeros son hilos de servicio que prestan su tiempo de procesador para escuchar el canal en espera de mensajes. El primero escucha en un socket UDP enlazado a la dirección IP local, en el puerto RAS estándar. El segundo escucha en una dirección IP multicast, para permitir a los terminales emplear el procedimiento de descubrimiento de Gatekeeper definido por H.323.

Enrutado de llamada

Este tercer bloque lo componen los tipos orientadas a gestionar conexiones H.225 CS enrutadas a través del Gatekeeper. La clase principal del bloque es *CallThread*, un hilo encargado del puerto de señalización H.225 CS del Gatekeeper, donde éste espera intentos de conexión TCP.

Por cada conexión H.225 CS enrutada por el Gatekeeper, se crea un hilo (objeto *SignallingThread*) para la gestión del canal. Controla dos sockets TCP: el primero conectado al terminal llamante, y el segundo al terminal llamado.

El objeto *SignallingThread* permite monitorizar los mensajes H.245 intercambiados:

- Los mensajes H.245 transportados por PDUs H.225.0 (*tunneling*) son gestionados por un objeto *H245Handler*, variable miembro del hilo.
- La apertura de un canal H.245 CTRLS dedicado es interceptada, y el canal se crea enrutado y controlado por un nuevo hilo, objeto *H245Thread* heredero de la clase *H245Handler*.

Cada objeto *SignallingThread* mantiene también referencias a un nuevo conjunto de objetos de manejo, herederos de la clase *H245Callback*, que define funciones a ser llamadas al recibir diversos mensajes H.245 (**OpenLogicalChannel**, **EndSessionCommand**, etc). Esto permite al QoSM detectar la apertura y cierre de los canales lógicos, y efectuar la reserva de recursos en consecuencia. Los objetos *H245Handler* y *H245Thread* asociados a una misma llamada comparten un mismo *Callback*.

B.2.2. Análisis estático: QoSM

La jerarquía de clases que se encarga de prestar el servicio de QoSM aparece en la figura B.7. Se utiliza el modelo de eventos y el concepto de objetos de manejo para separar claramente el código del QoSM del correspondiente al Gatekeeper, creándose tres tipos de objetos de manejo:

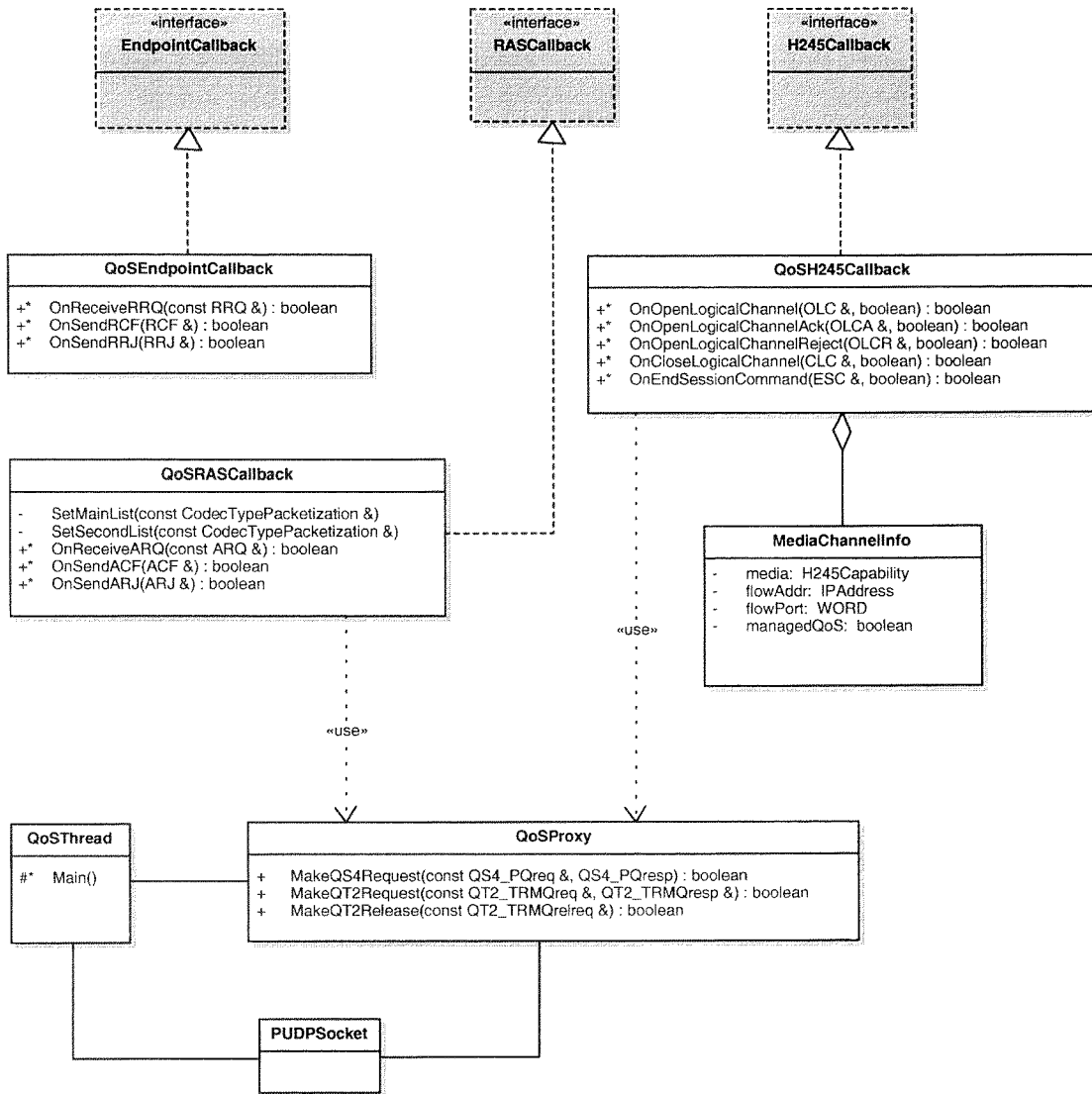


Figura B.7: Jerarquía de clases del QoSM

QoSEndPointCallback: Implementa las funciones *OnReceiveRRQ* y *OnSendACF*. Busca en el juego de características de la PDU *RRQ* el descriptor de soporte de QoS (como se define en A.3.3) y, si lo encuentra, incluye la respuesta en la PDU *RCF*.

QoSSRASCallback: Implementa las funciones *OnReceiveARQ* y *OnSendACF*. Se encarga de llevar a cabo la ordenación óptima de códecs.

QoSSH245Callback: Implementa las funciones relacionadas con la apertura y cierre de canales lógicos, y el comando de cierre de sesión. Se encarga de la señalización de QoS con el plano de transporte, cada vez que un canal lógico de voz o audio es abierto o cerrado.

Para el control de canales lógicos, mantiene un mapa donde asocia número de canal (más dirección del flujo: llamante → llamado o viceversa) a instancias de la clase *MediaChannel*. Esta clase representa el conjunto de datos mínimo necesarios para el control de QoS sobre el flujo: tipo de datos transportado, direcciones origen y destino, y un indicador que registra el estado de la reserva de recursos para el canal (activa / inactiva).

Toda la comunicación con servidores externos se lleva a cabo a través de una instancia de la clase *QoSProxy*. Éste recupera las direcciones de transporte de los servidores QoSPE y TRM del fichero de configuración global, y establece un canal de comunicación con cada uno. Actualmente, dichos canales se sustentan en un socket UDP compartido para ambos servidores. Las instancias de las clases *QoSSRASCallback* y *QoSSH245Callback* utilizan un mismo Proxy estático para ponerse en contacto con los servidores QS4 y QT2, respectivamente.

La escritura en el canal corre a cargo del propio objeto Proxy; éste envía el mensaje y entra en espera pasiva durante un tiempo máximo determinado. La recepción de los mensajes de respuesta enviados al QoSM por el resto de servidores es tarea de un hilo de ejecución independiente, implementado por la clase *QoSThread*, lo que hace posible tener varios hilos dormidos a la espera simultáneamente, utilizando el número de secuencia de cada PDU para decidir a cuál despertar cuando se recibe una respuesta.

B.2.3. Análisis dinámico: GateKeeper

En su comportamiento habitual como Gatekeeper, la función *Main* del único objeto estático de tipo *OpenGate* que se instancia crea tres hilos:

- Uno de tipo *RasThread*, para la recepción de primitivas RAS unicast.
- Otro de tipo *MulticastThread*, para la recepción de primitivas RAS multicast.
- Y el último, de tipo *CallThread*, para el establecimiento de conexiones H.225 CS enrutadas.

La mayor parte del código del Gatekeeper no ha sufrido modificaciones para añadir la nueva funcionalidad de QoSM; los cambios se han concentrado en únicamente tres procedimientos:

1. Registro de terminal (recepción de *RRQ* del canal RAS).

2. Admisión de llamada (recepción de primitivas *ARQ* del canal RAS)
3. Apertura de canales de medios (detección de primitiva *OpenLogicalChannel* en el canal de control H.245).

Registro de terminal

Al recibir el Gatekeeper una primitiva RAS (por puerto unicast o multicast), Se invoca a la función *HandleRasPDU* del objeto *RasServer* para procesarla. Este método determina el tipo de PDU recibida y utiliza una función especializada para proporcionarle el servicio adecuado. En el caso de una solicitud de registro, la función invocada es *OnRRQ*.

Esta función ha sido modificada para que, al añadir el nuevo elemento a la tabla de terminales registrados, añada a la nueva entrada la información de QoS aportada por el terminal.

Admisión de llamada

Al recibir un mensaje *ARQ*, la función invocada por *HandleRasPDU* es *OnARQ*. Esta función crea una instancia de la clase *CallDetails*, que será insertada en la tabla de llamadas si finalmente el establecimiento es aceptado. La función ha sido modificada para que anexe a esta instancia un nuevo objeto *QoSASRACallback* para la gestión de la QoS, y genere los eventos correspondientes tanto a la recepción del *ARQ* como al envío del *ACF* asociado. Esto da al QoSM la oportunidad de llevar a cabo la optimización, y almacenar los resultados obtenidos el *ACF* de respuesta.

Apertura y cierre de canales de medios

Los paquetes H.245 interceptados por el Gatekeeper en el modo de llamada rutado son primeramente clasificados en los cuatro tipos posibles: solicitud (*Request*), respuesta (*Response*), comando (*Command*) e indicación (*Indication*). Cada una de estas categorías es procesada por un conjunto de funciones distinto de la clase *H245Handler*.

Para permitir al QoSM que detecte la apertura y cierre de canales, se ha modificado el comportamiento de las funciones de proceso de las solicitudes **OpenLogicalChannel**, **CloseLogicalChannel** y **RequestCloseLogicalChannel**, de las funciones de proceso de respuesta **OpenLogicalChannelAck** y **OpenLogicalChannelRej**, y de la función de proceso del comando **EndSessionCommand**, de forma que dirijan los eventos correspondientes al objeto de manejo H.245 de la llamada.

Además, todas las funciones de proceso de mensajes H.245 han sido modificadas para que propaguen la información del origen del mensaje (terminal llamante o llamado), pues esto es necesario para clasificar los canales lógicos, ya que un número de canal sólo tiene sentido en una dirección.

B.2.4. Análisis dinámico. QoS

Al iniciarse la ejecución del Gatekeeper, se ejecuta una rutina de inicialización del QoS que carga las tablas de optimización (véase el capítulo B.2.5), crea una instancia estática de la clase *QoSProxy*, e inicia el hilo de escucha del socket. El resto del código de esta parte entra en acción cuando el Gatekeeper genera alguno de los eventos que el QoS es capaz de procesar.

Eventos RAS de registro de terminal

El tipo *QoSEndpointCallback* sobrecarga la función correspondiente a la recepción de una PDU *RRQ* del canal RAS (*OnReceiveRRQ*). En ella, se comprueba si el terminal ha incluido el descriptor de característica (*FeatureDescriptor*) correspondiente a la señalización expuesta en este documento (véase A.3.3), y de ser así, se almacena el valor de retardo de red de acceso y terminal contenidos en él. Estos valores son utilizados posteriormente en el proceso de optimización como mínimo retardo admisible.

Eventos RAS de control de llamada

El tipo *QoSRRASCallback* sobrecarga las funciones correspondientes a la recepción de los mensajes del canal RAS *ARQ* (*OnReceiveARQ*) y *ACF* (*OnSendACF*).

Los mensajes *ARQ* son capturadas con el objetivo de comprobar la presencia en ellas de primitivas de QoS, según el procedimiento descrito en el apartado A.3. La forma en que se procesan estas PDUs depende sustancialmente de su “carácter”: llamante o llamado.

- Las PDUs *ARQ* generadas con el objetivo de obtener admisión para una llamada saliente son tratadas como se muestra en el diagrama B.8. En primer lugar, se lleva a cabo el procedimiento de admisión a través del QoSPE, se haya negociado soporte de QoS con el terminal o no. La clase de Servicio solicitada al QoSPE será la indicada por el terminal o un valor por defecto si éste no ha especificado ninguna. Si se recibe una primitiva de confirmación, la Clase de Servicio asignada al cliente será la contenida en dicha primitiva. Si no, se le asignará una Clase de Servicio mínima. Estos valores definitivos se indican al cliente en la primitiva de confirmación, de manera que si éste no acepta la Clase de Servicio asignada, puede desistir de la llamada.

Una vez determinado el servicio a ofrecer, y si el cliente proporciona información de QoS, se ejecuta el procedimiento de optimización, delegando en la función *SetMainList*, y se activa una bandera que indica que el llamante solicitó QoS.

- Las PDUs generadas con el objetivo de obtener admisión para una llamada entrante son tratadas como se muestra en el diagrama B.9. Sólo en el caso de que tanto llamante como llamado soliciten control de QoS, se lleva a cabo el último paso de la optimización, invocando a la función *SetSecondList*.

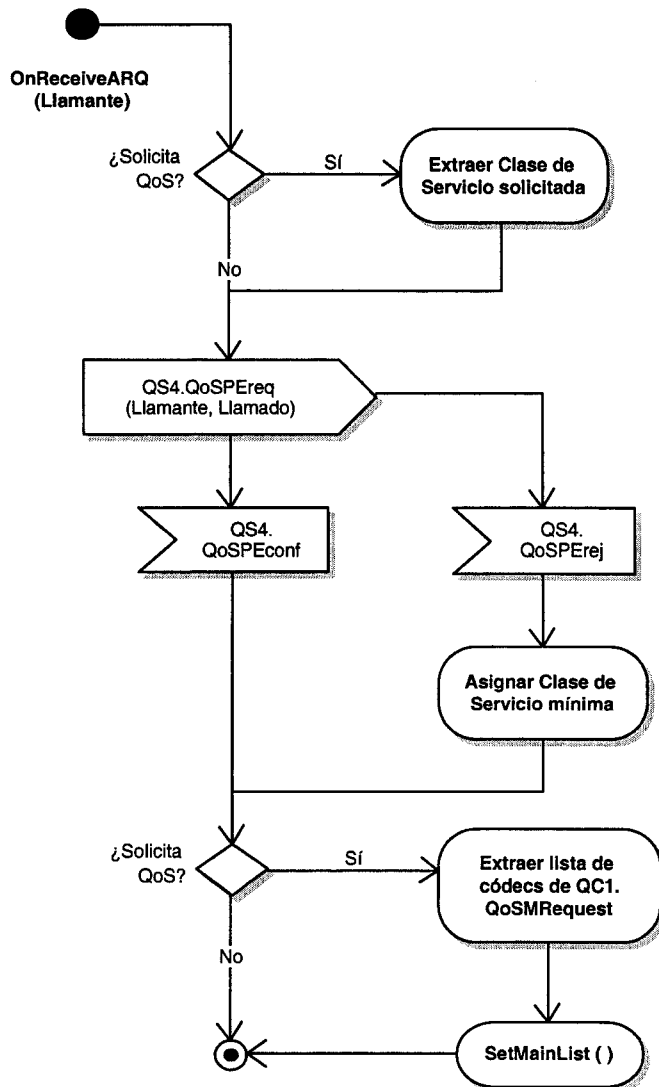


Figura B.8: Procesamiento de ARQ (solicitud de admisión para efectuar llamada)

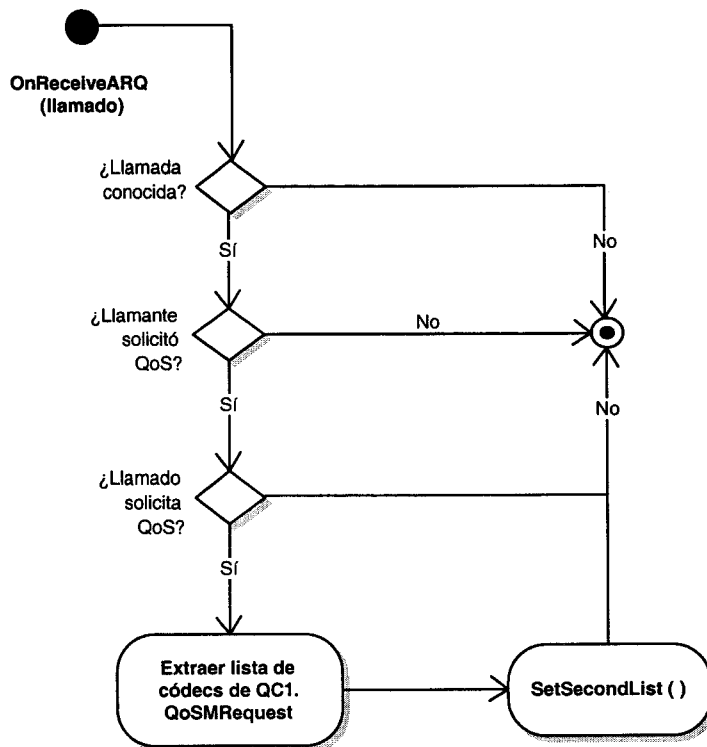


Figura B.9: Procesamiento de ARQ (solicitud de admisión para recibir llamada)

El orden de preferencia óptimo respecto al coste de los códecs a emplear, lo determinan las funciones *SetMainList* y *SetSecondList* del objeto *QoSRASCallback* asociado a la llamada.

La función *SetMainList* aparece en la figura B.10. En la primitiva de QoS transportada por la PDU *ARQ*, el cliente especifica tres rangos: el intervalo de valores admisibles para la Clase de Servicio, el conjunto de códecs soportados, y el número máximo de tramas por paquete admitidas por cada códec.

Variando estos tres grados de libertad, se tienen múltiples combinaciones, cada una con un coste. Esta función elige, para cada códec, la combinación de mínimo coste, y de acuerdo a éste inserta el códec en la posición adecuada en una lista, junto con el número de tramas por paquete de la combinación óptima. Esta información será enviada de vuelta al cliente en la primitiva **QC1.QoSMconf**, para que elija el códec y empaquetado más económico al abrir los canales de medios.

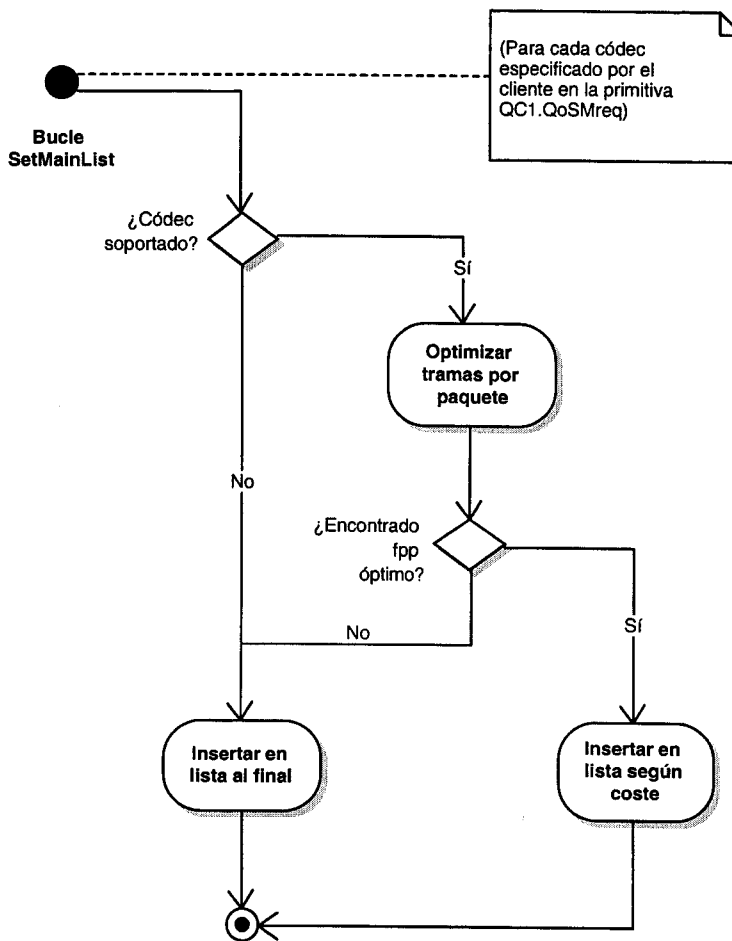


Figura B.10: Ordenación de códecs según coste

Para determinar el coste de cada una de las combinaciones de clase de servicio, códec y

tasa de empaquetado, se toman todas aquellas condiciones de retardo, variación del mismo y probabilidad de pérdida de paquetes admisibles que, para la tasa de empaquetado y códec dados, resultan en la clase de servicio requerida. Entre éstas, aquella con mínimo coste representa la condición óptima para la combinación, y especifica los parámetros de transporte a emplear en caso de que se abriera un canal lógico de esas características. El gasto asociado a la terna de parámetros se calcula empleando información suministrada por el proveedor de red en forma de ficheros de coste, como se indica en el capítulo B.2.5.

La función de optimización implementada no reconoce actualmente todos los códec soportados por H.323, limitándose a un subconjunto de códec audio. Por tanto, es posible que no sea capaz de determinar el coste mínimo asociado a un determinado códec. En ese caso, por convención se le supone un coste infinito, y se inserta en la última posición de la lista.

La función *SetSecondList* se ocupa de combinar la información de QoS enviada por el llamante, con la enviada por el llamado, para llevar a cabo una optimización completa cuando ya se disponen de todos los datos necesarios de ambos terminales. El proceso de optimización es igual al del caso anterior, pero en esta ocasión, en lugar de optimizar todos los códec enviados por el terminal, se procesan únicamente aquellos presentes tanto en el llamante como en el llamado.

Eventos de señalización H.245

La reserva de recursos de transporte se realiza de forma individual por cada flujo de medios, en el momento en que se detecta su apertura.

La función *OnOpenLogicalChannel* del objeto *QoSH245Callback* es invocada cuando se recibe una primitiva H.245 *OpenLogicalChannel*. La función registra el identificador de canal lógico y el tipo de información que se va a transmitir por él (códec y empaquetado), a la espera de que la apertura sea confirmada.

La función *OnOpenLogicalChannelAck* es invocada cuando se recibe una primitiva H.245 *OpenLogicalChannelAck*. En este punto, el canal lógico tiene fijado un códec y una tasa de tramas por paquete, registradas cuando se solicitó su apertura. Esta función determina el conjunto de condiciones de red (retardo, variación del retardo y probabilidad de pérdida) que, para el códec, empaquetado dados, y clase de servicio requeridas, resultan en un coste mínimo, y solicita la reserva de esos recursos al gestor del dominio de transporte (TRM) a través de la interfaz QT2. Este proceso de optimización es idéntico al descrito en el apartado anterior, pero contando únicamente con el valor de la Clase de Servicio como grado de libertad.

Cuando se abre un canal en alguno de los dos sentidos, el QoSM puede averiguar el número de puerto destino gracias a la información contenida en la primitiva *OpenLogicalChannelAck*, pero no le es posible determinar el puerto origen. En este caso, se le indica al TRM que enmascare cualquier tráfico entre ambos terminales con destino al puerto empleado por el canal lógico, cualquiera que sea su puerto origen, dándole a este último un valor 0 en la solicitud.

La función *OnCloseLogicalChannel* libera los recursos asociados a un canal lógico, cuando el cliente solicita su cierre mediante la primitiva H.245 *CloseLogicalChannel*.

La función *OnEndSessionCommand* se ejecuta cuando la conexión es liberada por ambos extremos. Los canales lógicos activos al recibir la primitiva H.245 *EndSessionCommand* son implícitamente cerrados por ambos terminales, de forma que el QoSM libera todos los recursos de transporte adquiridos por la comunicación.

B.2.5. Optimización

En el cálculo de la clase de servicio o factor R intervienen varios parámetros. Mediante la gestión de recursos de transporte y la señalización de QoS, se dispone de libertad para actuar sobre cinco de ellos:

- El códec empleado para la transmisión.
- El tipo de empaquetado (tramas por paquete).
- El retardo.
- La variación del retardo.
- La probabilidad de pérdida de paquetes.

Cada vector de este espacio de cinco dimensiones tiene asociado un cierto factor R, así como un coste. El objetivo del QoSM es encontrar, entre todos los vectores compatibles con lo solicitado por el cliente, aquél que conlleva un coste mínimo.

Dado un vector, calcular el R asociado es inmediato, pero no así a la inversa. Por ello, el QoSM necesita hacer uso de tablas que relacionen cada valor del factor R con todos aquellos vectores que lo proporcionan. Para acelerar el proceso de optimización, estas tablas son precalculadas, y cargadas por el QoSM en el proceso de arranque. En este apartado se discute el formato de dichas tablas, así como de las tablas de coste.

Tablas de factor R

Además de los parámetros mencionados arriba, en el cálculo del factor R influyen muchos otros (ver Apéndice C). Se han hecho las siguientes suposiciones:

- Ambos clientes utilizan el mismo códec
- El retardo total extremo a extremo será igual al retardo indicado por el vector.
- El factor I_e será función únicamente del códec y de la probabilidad de pérdida.
- Cancelación de eco perfecta ($TELR = 65$).

- T_a = retardo total extremo a extremo; $T = T_a$; $T_R = 2T_a$.
- Para el resto de parámetros, se suponen valores por omisión.

A fin de limitar el tamaño de las tablas, sólo se han considerado válidos aquellos vectores cuyo factor R fuera estrictamente mayor que cero, y cuyo retardo total fuera inferior a 500 ms.

Por otra parte, la forma en que la probabilidad de pérdida de paquetes afecta al factor R está tabulada para cada posible códec, de manera que únicamente hay hasta nueve condiciones distintas de probabilidad de pérdida posibles. En definitiva, el rango de variación de cada parámetro está limitado según aparece en el cuadro B.1

Códec	Máximo tramas por paquete	Valores de pérdida de paquetes
GSM	16	10
G.711	32	10
G.7235	10	1
G.7236	10	10
G.729	32	10

Cuadro B.1: Valores límite en vectores válidos

El QoS hace uso de dos tablas: la tabla de R ordenada, y la tabla de índices.

Tabla de R ordenada

Cada entrada de esta tabla consta de dos elementos: un vector, y el factor R asociado. El vector está representado mediante un número entero; el valor de sus bits es:

bits	significado
0 ... 8	retardo total
0 ... 12	prob. de pérdida ^a
13 ... 18	tramas por paquete
19 ... 23	códec ^b

^aNo como un porcentaje, sino como índice en una tabla de probabilidades de pérdida

^b0: GSM, ... 4:G.729

El valor de R está redondeado al entero más próximo. Los puntos en la tabla están ordenados siguiendo un triple criterio. En primer lugar, se ordenan por valor de R descendente. Para aquellos puntos con un mismo R, se ordenan por número de códec ascendente. Por último, aquellos puntos con igual R y códec se ordenan por número de tramas por paquete ascendente.

Esta tabla se implementa mediante un fichero de texto (*SortedMOS.txt*), donde cada línea es una entrada con el entero que representa al vector, en hexadecimal, y su factor R redondeado a un entero decimal, separados por espacio en blanco. En la implementación actual, la tabla consta de 151.644 entradas.

Tabla índice

Para facilitar la consulta de la tabla de factor R ordenado, se ha creado una tabla índice. Para cada combinación de factor R, códec y tramas por paquete, la tabla índice contiene un valor que señala al *primer elemento de la tabla de R ordenado* que corresponde con un vector de esas características.

De esta forma, para encontrar el vector óptimo para un factor R, códec y máximo de tramas por paquete dados (información que proporciona el cliente al QoSM), basta buscar en aquellos vectores en entre $\text{índice}(R, \text{códec}, 1)$ e $\text{índice}(R, \text{códec}, \text{tramas por paquete} + 1)$.

Esta tabla se implementa mediante un fichero de texto (*IndexMap.txt*), donde existe una entrada por cada tupla (factor R – 100...1 –, códec – GSM...G.7239, tramas por paquete – 1...máximo fpp correspondiente al códec). En la implementación actual, el número de tuplas es de 10.000.

Tablas de coste

Para calcular el coste asociado a un determinado vector, se utiliza un total de cinco tablas, que contienen el coste parcial imputable a cada uno de estos parámetros:

- Ancho de banda necesario.
- Tamaño de la PDU.
- Retardo máximo exigido a la red.
- Varaiación máxima del retardo.
- Máxima probabilidad de pérdida de paquetes.

Estas tablas son generadas a partir de información proporcionada por el operador de red, en forma de ficheros de coste. Cada uno de estos ficheros define el coste asociado a uno de los conceptos anteriores en forma de un polinomio, que puede estar descrito a tramos. El formato de estos ficheros es una secuencia de pares de líneas donde cada par describe un tramo:

- La primera de las dos líneas de cada par define el intervalo, con el siguiente formato: # Mínimo:Máximo. El valor máximo puede ser omitido, en cuyo caso se toma como infinito.
- La segunda contiene la descripción del polinomio, en forma de una secuencia de coeficientes separados por espacio en blanco, empezando por el coeficiente de X^0 .

Estos polinomios se emplean al inicio de la ejecución del QoSM para precalcular los costes asociados a cada vector de los definidos en el fichero de R ordenado. El rango de los valores

de salida de dichos polinomios debe ser del mismo orden, para que los costes parciales de cada parámetro sean comparables entre sí. El rango de valores de entrada se define en la siguiente tabla:

Parámetro	Fichero	Rango de entrada	Unidades
Ancho de banda	Rb. coste	0... ^a	bps
Tamaño de PDU	Tam. coste	0... ^b	bits
Retardo	DI. coste	0...500	milisegundos
Variación del retardo	Ji. coste	0...500	milisegundos
Prob. de pérdida	Pl. coste	0...1000	tanto por mil

^a oscilará entre aprox. 6400 del códec G.7235 con 10 tramas por paquete, y 96000 del G.711 – incluyendo la cabecera RTP –

^b oscilará entre aprox. 400 del códec G.729 con 1 trama por paquete, y 20800 del G.711 con 32 – incluyendo la cabecera RTP –

B.3. Servidor QoSPE

Este servidor es el encargado de mantener la información de admisión asociada a cada usuario, y de aplicar las políticas de admisión definidas cada vez que se solicita permiso para el establecimiento de una llamada; es invocado desde el QoSM.

En la implementación actual, el QoSPE se limita a autorizar cualquier establecimiento solicitado, si bien la estructura interna del programa se diseñó de forma que la ampliación del mismo para añadir políticas de admisión fuera en un futuro sencilla.

B.3.1. Análisis estático

La estructura del servidor se muestra en la figura B.11. El tipo principal es *QoSPE_Process*, heredero de *PProcess*, y que define la función *Main()* del ejecutable. El tipo *QoSPE_Srv* se encarga de la gestión del puerto de servicio, 8000.

En la actualidad, las solicitudes son confirmadas, asignándole a los parámetros el mismo valor contenidos en la solicitud. En previsión de una futura ampliación que añada la administración de políticas de admisión, se definieron un conjunto de clases de utilidad:

- La clase *QoSPE_EndPoint* contiene un conjunto de información útil para administrar políticas de admisión a un terminal: Su identificador, su dirección IP, la Clase de Servicio garantizada al terminal por defecto, la Clase de servicio máxima que se puede autorizar al terminal, y dos listas: Una primera de destinos prohibidos, y una segunda de franjas horarias prohibidas.
- La clase *QoSPE_Reader* ofrece las funciones necesarias para cargar de fichero la lista de objetos *QoSPE_EndPoint* conocidos por el servidor. El fichero ha de ser una secuencia de líneas de texto, conteniendo cada una:
 1. Una lista de alias del terminal, separados por comas.

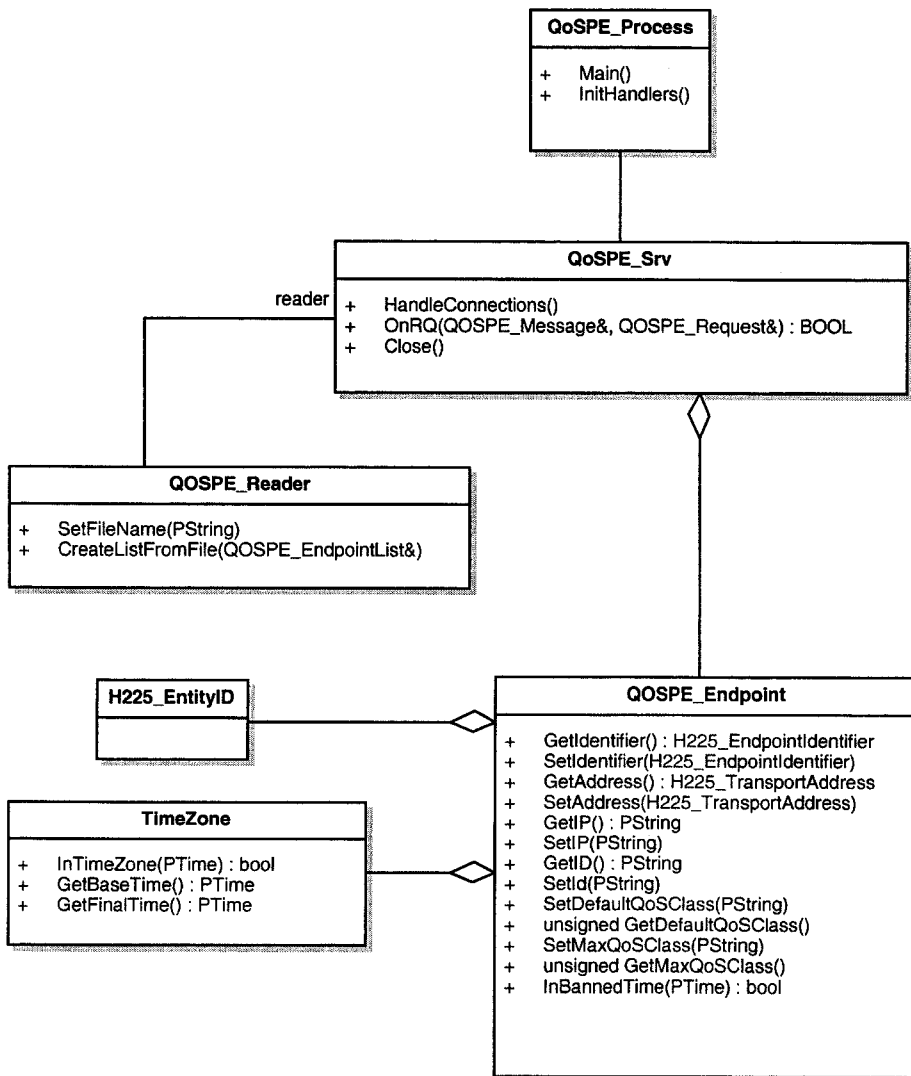


Figura B.11: Jerarquía de clases del servidor QoSPE

2. Una lista de alias de los terminales a los cuales tiene prohibido llamar.
3. Una lista de franjas horarias en las cuales no se le permite la conexión.
4. Un valor numérico que indica la clase de servicio garantizada por defecto al terminal.
5. Un valor numérico que indica la clase de servicio máxima para la que el terminal puede obtener autorización.

Los elementos se separarán mediante el carácter “%”. En la implementación actual, los elementos segundo y tercero de la lista no se usan, de forma que en realidad una línea consta de tres campos: alias del terminal y clases de servicio garantizada y máxima.

B.3.2. Análisis dinámico

La secuencia de ejecución se puede dividir en dos tramos: un primer tramo de inicialización, y un segundo de escucha.

- **Inicialización:** El proceso comienza por la ejecución del método *Main* de un objeto *QoSPE_Process* estático.
 1. El programa establece un manejador de señales de Windows, para poder interceptar la señal de terminación y actuar en consecuencia.
 2. Se crea un objeto de tipo *QoSPE_Server*, que escuchará en el puerto UDP 8000.
 3. Se carga desde un fichero la lista de terminales reconocidos. Esta lista no se usa en la implementación actual.
 4. Se lanza el bucle de escucha infinito del objeto *QoSPE_Srv*, la función *Handle_Connections*.
- **Escucha:** Esta fase es el cuerpo de la función *Handle_Connections*. La función permanece en un bucle infinito, en el que se atienden los mensajes recibidos y se genera una respuesta. Esta respuesta consiste en una primitiva **QS4. QoSPEconf** en caso de que la solicitud recibida sea un **QS4. QoSPEreq**, y una primitiva **QS4. QoSPErej** en caso contrario.

B.4. Entidades de transporte

Las entidades del dominio de transporte (TRM y TPE) han sido implementadas empleando una base común. Se ha creado un servidor de QoS genérico, capaz de atender la llegada de PDUs de QoS a través de un socket UDP, y sobre este servidor se han implementado las funcionalidades de TRM y TPE en forma de módulos de servicio (“*plug-ins*”) cargables en tiempo de ejecución.

En primer lugar, se expondrá la estructura de la parte común, servidor de QoS. A continuación, se analizarán los módulos que proporcionan el servicio de TRM y TPE.

B.4.1. Servidor de QoS. Análisis estático

El servidor de QoS, parte común de TRM y TPE, puede dividirse en varios subsistemas:

- Subsistema de configuración.
- Subsistema de registro (“*trace*”).
- Subsistema de carga de módulos.
- Subsistema de comunicaciones.
- Subsistema de QoS.

Subsistema de Configuración

Esta subsistema está encargado de la lectura del fichero de configuración. La estructura de clases del mismo consiste en un único tipo, que ofrece funciones para leer un fichero y descomponerlo en una lista de pares clave – valor. El fichero consiste en una secuencia de líneas. La primera palabra de cada línea es considerada una clave; el valor asociado a la misma consiste en el contenido del resto de la línea. Los parámetros de configuración que deben ser incluidos en el fichero dependerán del tipo de servidor lanzado (TRM o TPE).

Subsistema de Registro

El subsistema de registro lo forman un conjunto de clases, mostrado en la figura B.12, que simplifican la gestión del archivo de registro.

El objetivo de la jerarquía es formatear de manera uniforme los datos enviados al fichero de registro, de forma que puedan ser fácilmente procesados por los programas que forman la interfaz Web.

Los tipos *growingbuf* y *GrowStream* facilitan la gestión de memoria, representando colass que crecen automáticamente bajo demanda, y que ofrecen la interfaz de la clase *std::ostream* para añadirles contenido. La clase *Filter* aplica un posproceso a la información que va a ser escrita al registro, para garantizar la uniformidad de formato, permitiendo la división de la información de un evento en distintos niveles de urgencia. Por último, la clase *TracePoint* representa un evento de trazado.

Subsistema de Proceso

Este es el subsistema fundamental. Define la estructura básica del servidor, y la forma en que los módulos de servicio encajan en esta estructura y se integran al servidor para completarlo. Su esquema aparece en la figura B.13.

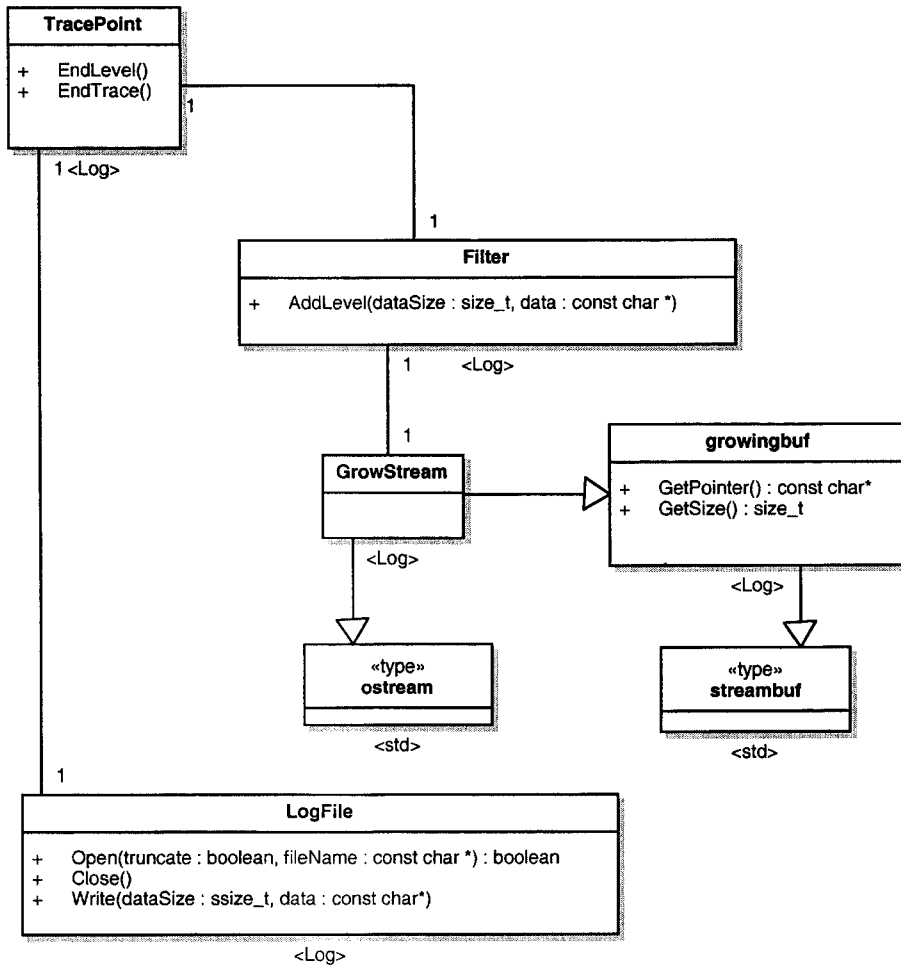


Figura B.12: Subestructura de registro

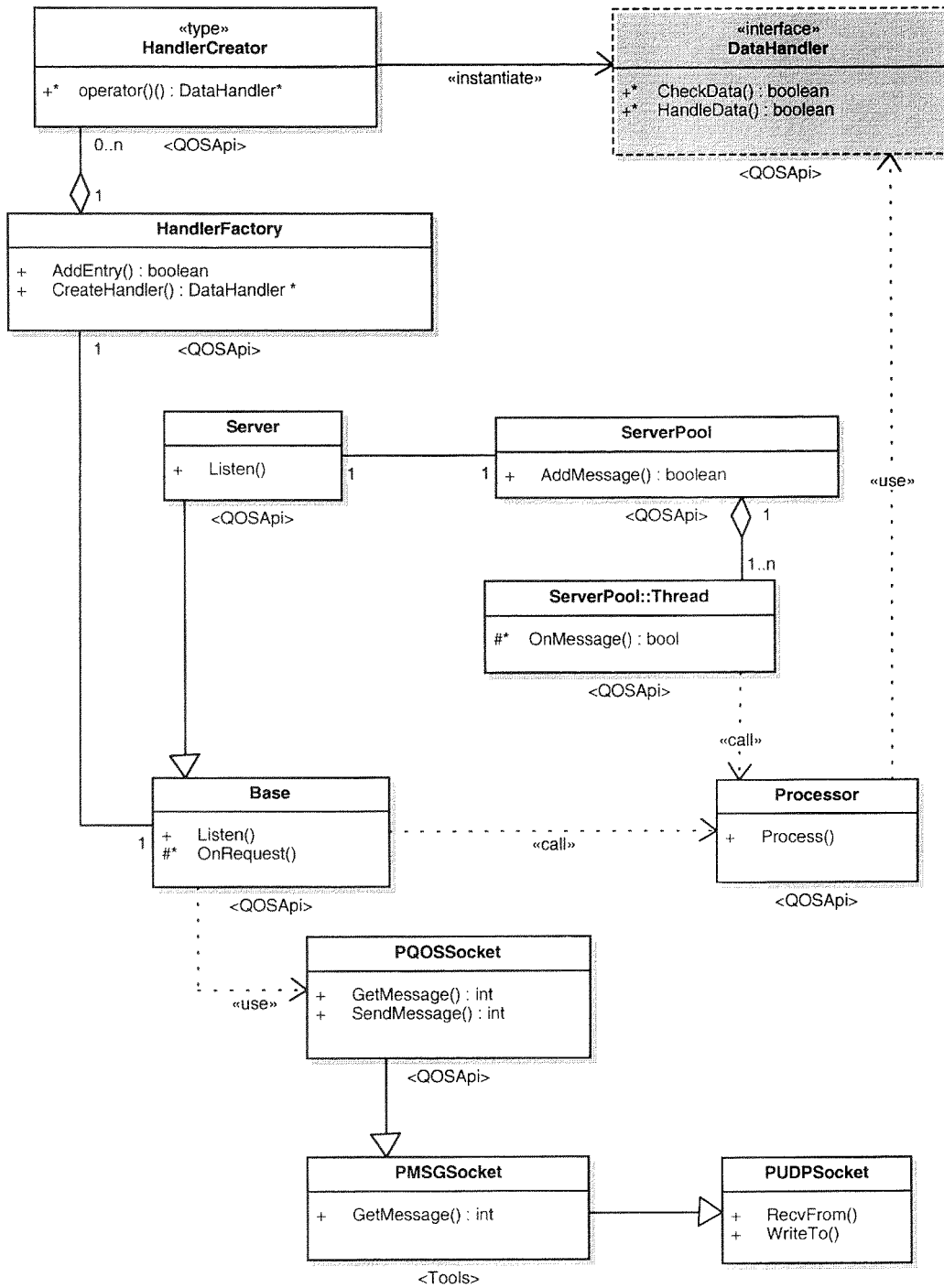


Figura B.13: Subsistema de proceso

La clase *Main* se encarga de las tareas de puesta en marcha del programa: lectura del fichero de configuración, carga de módulos, establecimiento del gestor de interrupciones, y creación del socket UDP de servicio.

La clase *PQOSSocket* eleva la interfaz del tipo *PUDPSocket* (socket UDP) de la librería *PWlib*, pasando las unidades básicas de información de ser cadenas de octetos a ser PDUs de QoS.

La clase *ServerPool* se encarga de la gestión de hilos de ejecución. Los objetos de este tipo contienen un conjunto de hilos y una cola compartida por ellos. Las PDUs recibidas son encoladas en el objeto, y procesadas por algún hilo de servicio tan pronto como sea posible.

La clase *Server* atiende el puerto UDP en espera de peticiones de QoS, y las encola en un objeto *ServerPool* tan pronto como llegan.

También se admite la posibilidad de que las peticiones sean atendidas nada más llegar sin emplear objeto *ServerPool*, en lugar de ser encoladas; esto permite que el servidor pueda ejecutarse en modo monohilo. Este comportamiento lo implementa la clase *Base*. Es posible, mediante un parámetro de configuración, elegir cuál de los dos modelos de servicio se aplicará.

La clase *Processor* lleva a cabo el procesamiento de las PDUs de QoS. El objeto *Processor* debe valerse del identificador que transporta cada elemento *GenericData* de la PDU recibida para determinar el servicio requerido por él, y proporcionárselo.

La clase abstracta *DataHandler* es el instrumento que el tipo *Processor* utiliza para dar el servicio necesario a las solicitudes recibidas. Este tipo define dos funciones virtuales puras: una función de verificación (“*CheckData*”) y una de servicio (“*HandleData*”). Estas funciones serán implementadas por “Manejadores de Servicio”, tipos derivados de *DataHandler* definidos por los módulos de servicio. Mediante esos manejadores, el tipo *Processor* es capaz de servir cualquier solicitud de QoS soportada por alguno de los módulos cargados.

La clase *HandlerFactory* es un mapa de servicios: Asocia un valor de identificador de *GenericData*, con el Manejador de Servicio adecuado. Cada módulo cargado registrará su manejador en esta tabla, para hacerlo accesible a los objetos de tipo *Processor*.

Subsistema de carga de módulos

Este subsistema se ocupa de la gestión de los módulos que se cargan en el servidor. Los módulos se cargan de forma local, es decir, la tabla de símbolos de un módulo cargado dinámicamente no está disponible para el resto. Por tanto, todo aquel módulo que necesita exportar datos, debe anotarlos en un mapa de símbolos. Cada módulo implementa su propio mapa de símbolos; El módulo principal mantiene una lista que permite acceder a todos ellos (objeto de tipo *List*), consultable desde cualquier módulo.

Las rutinas que cada módulo hace accesible al servidor de QoS son cuatro:

- Una de activación, que es invocada al ser cargado el módulo. Devuelve una cadena de caracteres (alias del módulo), que puede ser utilizada por otros módulos para recuperar su mapa de símbolos.
- Una de reactivación, invocada al recibir el servidor la señal *SIGHUP*.
- Una de desactivación, invocada justo antes de ser descargado.
- Una última que permite tener acceso al mapa de símbolos ofrecido por el módulo.

Estas cuatro rutinas se han incorporado como funciones virtuales puras al objeto *EntryPoint*, que aparece en la figura B.14. Cada módulo exporta una función *extern "C" Mod::EntryPoint *GetEntryPoint (void)* que permite recuperar su objeto *EntryPoint*.

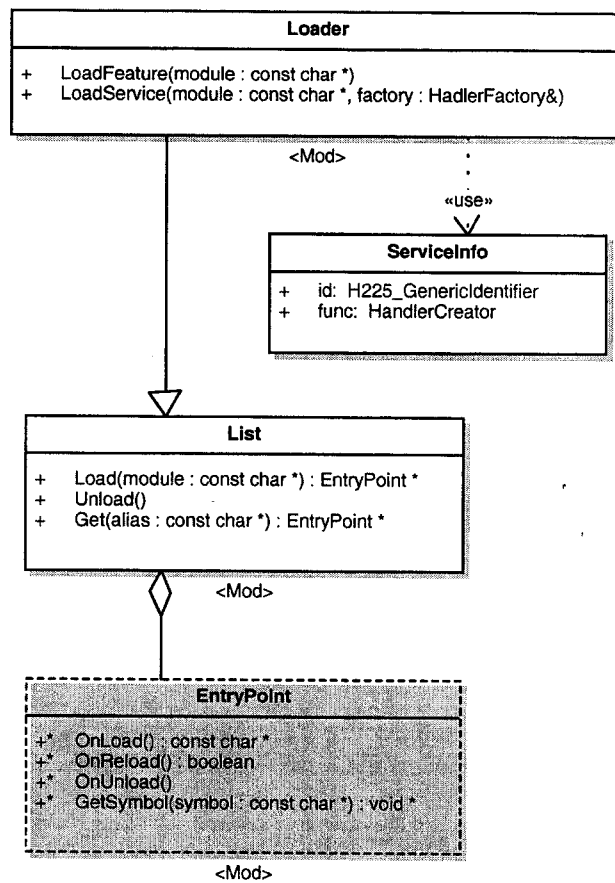


Figura B.14: Subsistema de carga de módulos

Los módulos se han dividido en dos clases: módulos de servicio (aquellos que implementan un Manejador de Servicio), y módulos de características. En el caso de los primeros, se define una entrada especial en su mapa de símbolos: utilizando la clave "RegisterService", el mapa devuelve

un objeto de tipo *ServiceInfo*, que contiene el valor de identificador de *GenericData*, y un Creador de Servicio (función que instancia Manejadores de Servicio), asociados al módulo.

B.4.2. Servidor de QoS. Análisis dinámico

En ejecución, el comportamiento del servidor puede dividirse en dos áreas:

- **Bucle de escucha:** Actividades realizadas para inicializar el servicio y abrir y mantener un socket UDP donde recibir peticiones de QoS.
- **Provisión de servicio:** Tareas realizadas para satisfacer el servicio que demandan las peticiones recibidas.

A su vez, esta segunda parte se puede a su vez subdividir en dos tareas: proceso independiente de servicio, ejecutado por el servidor de QoS, y proceso específico de servicio, llevado a cabo por los módulos cargados. Sólo la primera es tratada en esta sección.

Bucle de escucha

Al ser ejecutado el servidor, se lee el fichero de configuración y se cargan los módulos especificados. A continuación, se entra en el bucle de escucha infinito, del que sólo se sale por error de socket o interrupción (en caso de recibir la señal *SIGHUP*, el fichero de configuración es recargado, y se reinicia el servicio).

Las PDUs de QoS de tipo *Requets* recibidas a través del socket son encoladas, a la espera de ser atendidas por algún hilo de servicio. En caso de que el servidor esté configurado en modo monohilo, las PDUs no se encolan, sino que son inmediatamente procesadas. En ambas situaciones, el proceso que se sigue una vez que existe un hilo disponible es el mismo, descrito en el siguiente apartado.

Proceso independiente de servicio

Se pueden distinguir tres etapas:

1. La primera etapa consiste en la validación de la PDU. Se detecta la presencia de errores básicos (p. e. , si la PDU recibida era excesivamente grande y ha sido truncada en recepción).
2. Los elementos *GenericData* de la PDU se agrupan por interfaz, y se instancia un manejador de servicio por cada grupo. Éste se utiliza para verificar que no existen errores de protocolo, y procesar la información.
3. Las respuestas generadas por los manejadores de servicio son integradas en una PDU de respuesta y enviadas de vuelta al cliente.

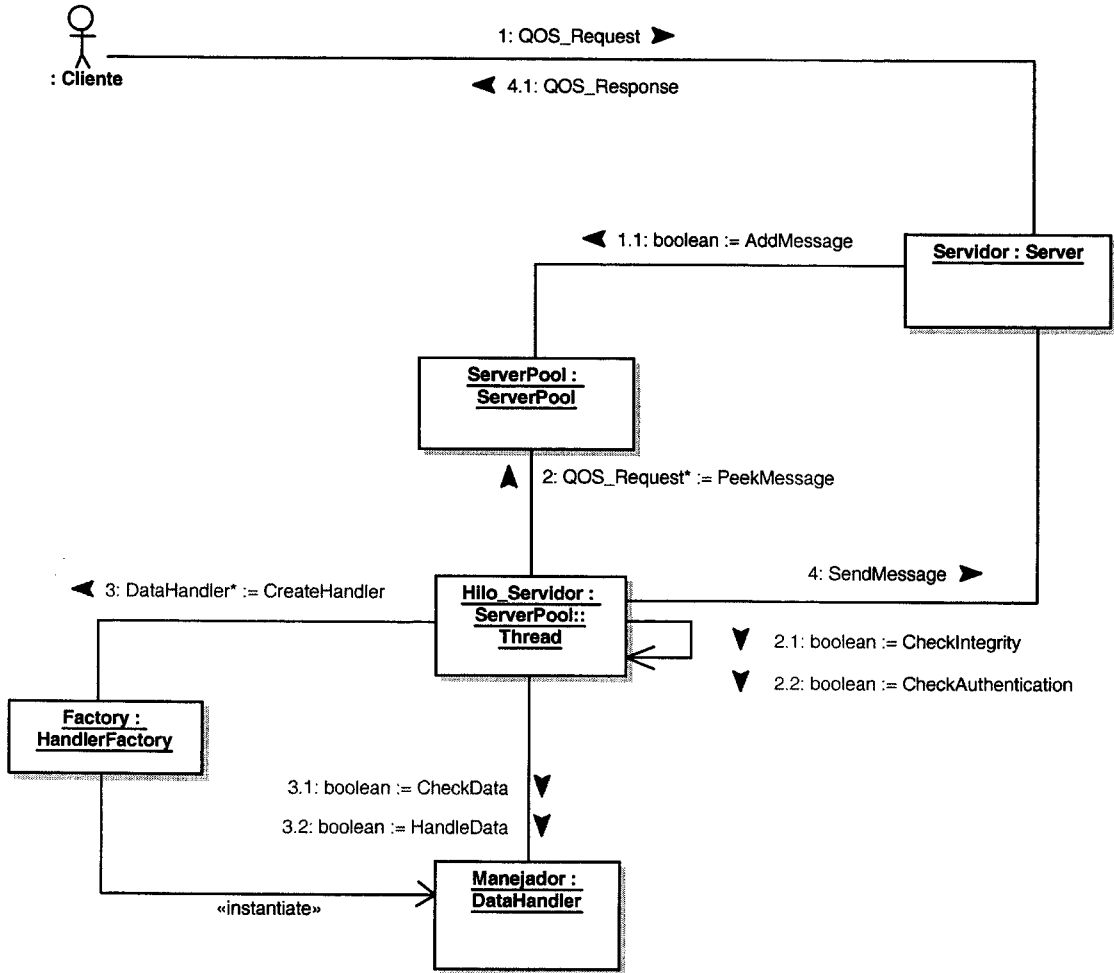


Figura B.15: Proceso independiente de servicio

B.4.3. Módulos instalables. Análisis estático

En la actualidad, se han implementado tres módulos: Uno para cada una de las interfaces QT2 y QI4, y otro que contiene los datos característicos del servidor TRM que necesita la interfaz QT2: Tabla de flujos e información sobre el servidor TPE a emplear para obtener las autorizaciones.

Módulo de servicio QI4

Este módulo implementa la funcionalidad del TPE. En esta implementación, no se han definido mecanismos ni políticas de control de admisión, de manera que la única tarea que realiza el servidor es responder de manera afirmativa las solicitudes remitidas por el TRM. Su estructura de clases consiste simplemente en un tipo, heredero de *DataHandler*, que implementa las funciones de chequeo y servicio de manera trivial.

Módulo de servicio QT2

La estructura de clases del módulo compone una cadena de herencia, basada en el tipo *DataHandler*, como se muestra en la figura B.16. Cada eslabón de la cadena se encarga de una subtarea:

- Validación de elementos de datos, detectando errores de protocolo. Asignada al tipo *ServerHandler*.
- Solicitud de autorización al servidor TPE para el establecimiento de portadoras. Asignada al tipo *AuthHandler*.
- Actualización de los ficheros de registro y estado, para mantener al corriente la información mostrada por la interfaz Web. Asignada al tipo *UnpackerHandler*.
- Señalización con la funcionalidad de transporte (TF). Asignada al tipo *Nistnet::Handler*.
- Señalización con la funcionalidad de interconexión (IWF). Asignada al tipo *Firewall::Handler*.

El proceso de las primitivas se realiza mediante los métodos de los tipos *NISTNet::Handler* y *Firewall::Handler*, que se encargan de la comunicación con la funcionalidad de transporte (NISTNet) y la de interconexión (cortafuegos), respectivamente. Las interfaces QI3 y QI5 han sido implementadas en forma de llamadas a funciones de librería, puesto que tanto el emulador de red como el cortafuegos son entidades locales al TRM.

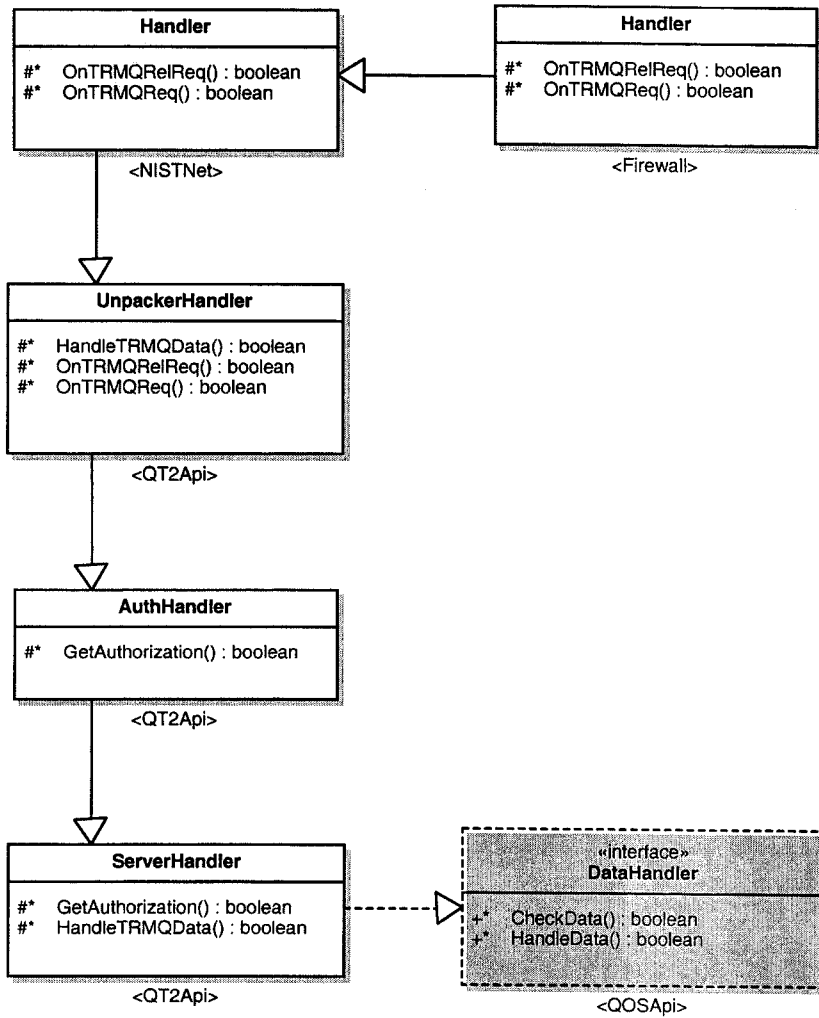


Figura B.16: Módulo de interfaz QT2

Módulo de características TRM

En este módulo se agrupan los tipos de datos de representan información característica del TRM:

- Servidor TPE empleado: dirección IP y puerto.
- Estado del dominio de transporte: información sobre los flujos establecidos.

Sobre cada flujo establecido, se almacena información de direccionamiento y de recursos adquiridos. Esta información se exporta a un fichero de estado, de forma que esté accesible y pueda ser consultada a través de la herramienta de gestión Web diseñada. Este fichero de estado es de tamaño fijo, por razones de eficiencia, lo que implica que existe un número máximo de conexiones admisibles, más allá del cual el fichero desbordaría. Esta cota puede establecerse mediante parámetros de configuración.

B.4.4. Módulos instalables. Análisis dinámico

Módulo de servicio QI4

La implementación de las funciones de verificación y servicio es trivial: La función de verificación se limita a comprobar que las primitivas de QoS recibidas pertenecen a la interfaz QI4, y la función de servicio genera tantas primitivas QI4. PQconf como primitivas QI4. PQreq se hayan recibido.

Módulo de servicio QT2

Validación de elementos *GenericData*: la realiza la función *CheckData* del tipo *ServerHandler*, comprobando que todos los campos del elemento de datos contienen un objeto identificable como primitiva de solicitud de interfaz QT2 (QT2. TRMQreq ó QT2. TRMQrelreq).

Procesado de elementos *GenericData*: Se realiza en diferentes pasos, cada uno asignado a una clase de la cadena de herencia. En las figuras B.17 y B.18 se muestra el proceso de un mensaje *TRMQreq* y de un *TRMQrelreq*. Es necesario indicar que todos los bloques del diagrama representan diferentes vistas de un único objeto, de clase *Firewall::Handler*, que invoca métodos miembro definidos en alguna de sus clases base para ejecutar el proceso.

Todas las primitivas QT2. TRMQreq contenidas en los objetos *GenericData* que se estén procesando son autorizadas a la vez, para minimizar el tráfico entre TRM y TPE. Las solicitudes denegadas por el TPE interrumpen su proceso aquí; el resto son procesadas una a una. Se reserva un sitio en la tabla de estado (el fichero de estado es de tamaño fijo), y se da servicio al mensaje, añadiendo a las tablas del emulador de red y del cortafuegos las entradas correspondientes, mediante las funciones de librería que hacen las veces de interfaz QI5 y QI3. En caso de éxito, el nuevo flujo se almacena en el fichero de estado, y se devuelve al

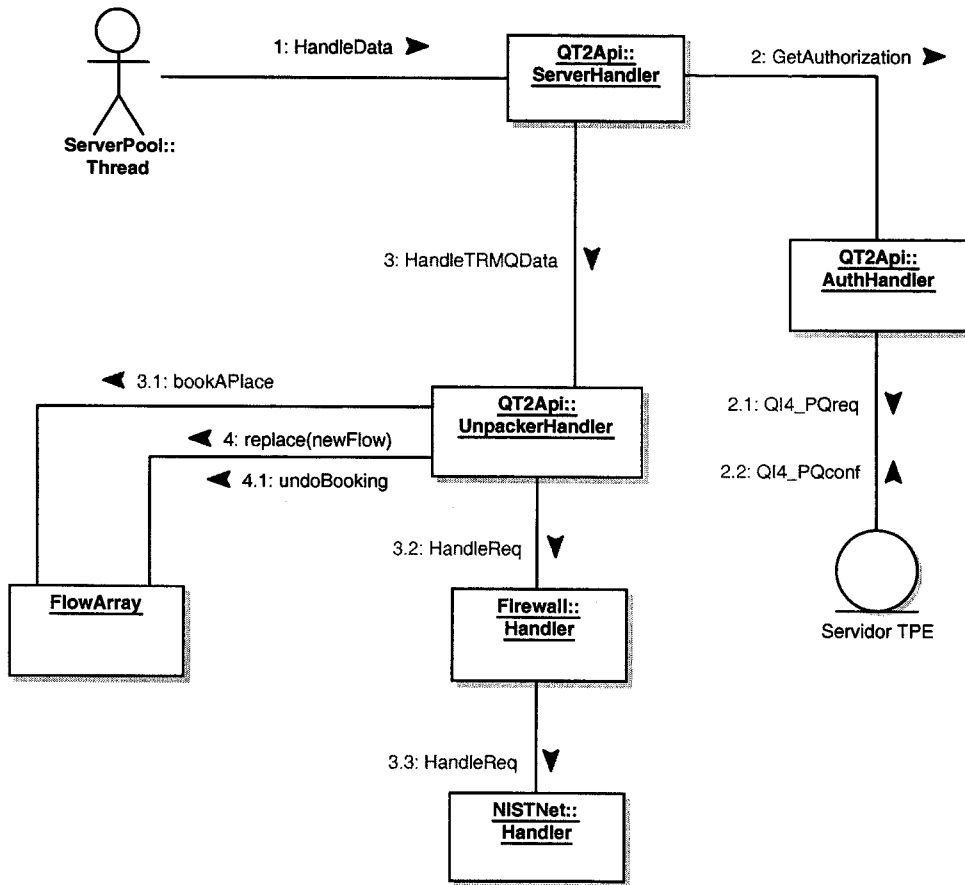


Figura B.17: QT2. Proceso de petición de adquisición de recursos

hilo servidor una primitiva **QT2. TRMQconf**. En otro caso, se devuelve una primitiva **QT2. TRMQrej**.

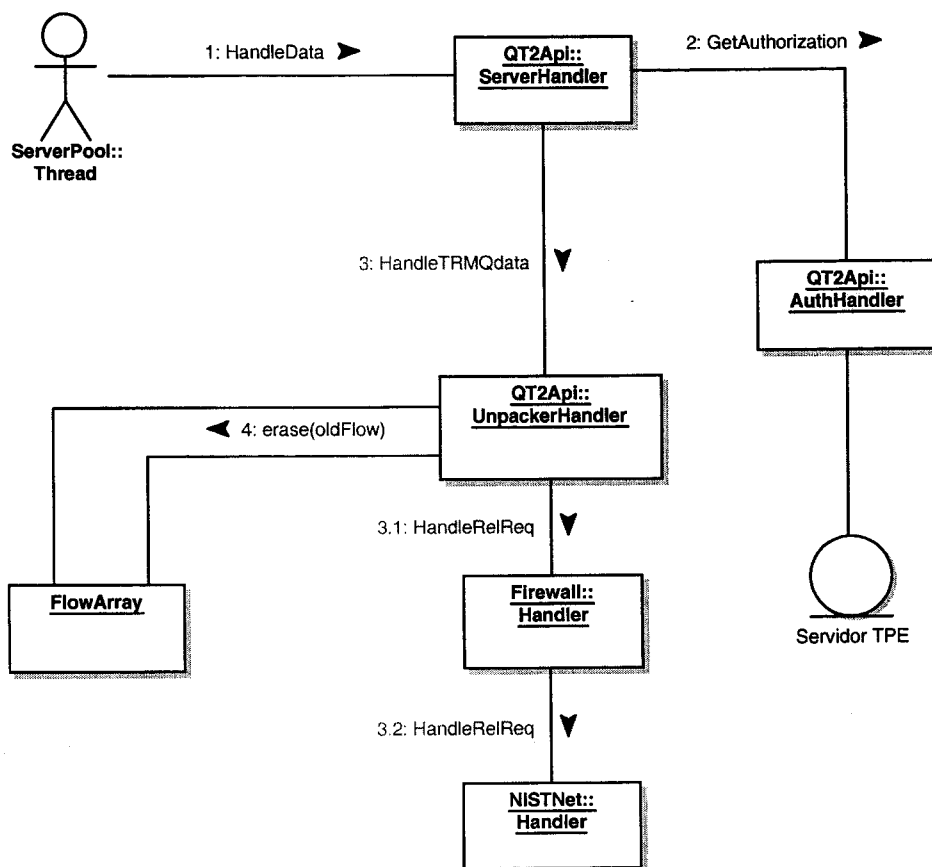


Figura B.18: QT2. Proceso de petición de liberación de recursos

El proceso de una primitiva **QT2. TRMQrelreq**, representado en el esquema B.18, es muy similar, con la diferencia de que no es necesario autorizar las solicitudes a través del servidor TPE. El flujo especificado por la primitiva es eliminado de las tablas del emulador de red, el cortafuegos, y del fichero de estado.

Apéndice C

Cálculo de los parámetros de entrada al Modelo-E

C.1. Introducción

El Modelo-E, descrito en la sección 2.3.2 es utilizado para el cálculo del factor R , que indica el nivel de calidad en una conversación telefónica. Dicho modelo constituye la base de la optimización descrita en la sección 4.6.1. Para aplicarlo a las redes de VoIP es posible seguir la guía proporcionada en [40]. En este Anexo se presentan los valores por omisión de los diversos parámetros de entrada al modelo, así como las referencias necesarias para su cálculo.

Por último también se introduce un protocolo de medidas que permite obtener el retardo debido a los terminales. Estas medidas son necesarias para el correcto funcionamiento del modelo propuesto en la presente Tesis Doctoral, por lo que se puede aplicar dicho protocolo de medidas cuando el fabricante del cliente no proporcione datos relativos al retardo.

C.2. Determinación de los valores de entrada al Modelo-E

La figura 2.3 muestra los diversos parámetros de entrada. Los valores por omisión de cada uno de dichos parámetros se reflejan en el cuadro C.1.

Las redes de VoIP sin conexión con RTC son completamente digitales, por lo que las fuentes de eco se sitúan en los terminales, luego el valor de T será igual al de T_a . Así mismo T_r será el retardo de ida y vuelta. En [41] se describen dos métodos que permiten hallar T_r y T en las redes de VoIP. El más sencillo para redes de VoIP consistiría en la aplicación de los datos de monitorización proporcionados por RTCP. Ambos métodos asumen un retardo simétrico en la red para el cálculo del retardo en un sentido.

En los terminales utilizados en redes de VoIP el ajuste de ganancias permite controlar el valor de las pérdidas en los terminales, por lo que los valores de SLR y RLR permanecen apro-

156 APÉNDICE C. CÁLCULO DE LOS PARÁMETROS DE ENTRADA AL MODELO-E

Parámetro	Abreviatura	Unidad	Valor
Índice de Sonoridad en Emisión	SLR	dB	+7
Índice de Sonoridad en Recepción	RLR	dB	+3
Efecto local en el camino del hablante	STMR	dB	15
Efecto local en el camino del oyente	LSTR	dB	18
Factor D del Teléfono	D	-	3
Pérdidas de Eco del hablante	TELR	dB	65
Pérdidas de Eco del oyente	WEPL	dB	110
Retardo medio en un sentido del trayecto del eco	T	ms	0
Retardo en un bucle a 4 hilos	Tr	ms	0
Retardo absoluto en conexiones sin eco	Ta	ms	0
Número de Unidades de distorsión de cuantización	qdu	-	1
Factor de Equipo	Ie	-	0
Ruido de Circuito (en 0dBr)	Nc	dBm0p	-70
Ruido de planta en el lado del receptor	Nfor	dBmp	-64
Ruido de la sala en el lado del hablante	Ps	dB(A)	35
Ruido de la sala en el lado del oyente	Pr	dB(A)	35
Factor de expectativa	A	-	0

Cuadro C.1: Valores por omisión en el cálculo del Modelo-E

ximadamente igual a los que muestra la tabla C.1. En [42] y [43] se describen los principios generales y la metodología para la determinación de los índices de sonoridad.

El factor D puede ser medido mediante los procedimientos descritos en la Recomendación [3].

El valor del efecto local puede ser ajustado a niveles confortables para el usuario en aquellos terminales basados en PC, por lo que $STMR$ y $LSTR$ podrán tomar sus valores por omisión.

Para determinar las pérdidas de eco ($TELR$ y $WEPL$), así como el ruido del circuito (Nc) es posible utilizar los procedimientos descritos en [44].

Los valores para el ruido de sala en ambos lados (Pr y Ps) corresponden a una oficina normal sin un ruido excesivo, por lo que en la mayoría de los casos podrán aplicarse los de la tabla C.1. En [21] se proporcionan tablas con valores de la mayoría de los parámetros.

El factor de expectativa (factor A) toma diversos valores según la ventaja de acceso de la red. Para redes de VoIP provisionalmente se toma el mismo valor que para las redes de telefonía fija, esto es, 0. En [40] se proporciona una guía para la aplicación del factor A en las redes de VoIP.

La distorsión de equipos especiales (factor Ie) tiene dos componentes:

- Intrínseca (I_e^{codec}): depende únicamente del codec utilizado. En el cuadro C.2 se muestran los valores para los distintos codecs.

APÉNDICE C. CÁLCULO DE LOS PARÁMETROS DE ENTRADA AL MODELO-E 157

- Extrínseca (I_e^{PL}): depende de la tasa de pérdida de paquetes (PL). La influencia de esta varía en función del codec utilizado. En el cuadro C.3 se muestra, en función del codec y la tasa de pérdida de paquetes (PL), la disminución del factor de calidad R. Los codecs de baja tasa de bit utilizan mecanismos que permiten mitigar la influencia de la pérdida de paquetes. En el codec G711 se puede implementar algún algoritmo que permita disminuir la influencia de este fenómeno (PLC, Packet Loss Concert) .

Por tanto, la influencia total de la distorsión vendrá dada por el factor

$$I_e = I_e^{codec} + I_e^{PL}.$$

Codec	Régimen Binario (kbit/s)	I_e
G.726, G.727	40	2
G.721, G.726, G.727	32	7
G.726	24	25
G.726, G.727	6	50
G.728	26	7
	12.8	20
G.729	8	10
G.729-A + VAD	8	11
GSM 06.10 full rate	13	20
GSM 06.20 half rate	5.6	23
GSM 06.60 enhancement rate	12.2	5
G.723.1	5,3	19
G.723.1	6,3	15

Cuadro C.2: Factor de degradación debido al codec I_e^{codec}

Cuando existe transcodificación se debe sumar el factor I_e de cada uno de los codecs implicados.

Así mismo, en [45] se proporcionan métodos subjetivos que permiten obtener el valor de I_e debido al uso de otros equipos especiales como canceladores de eco. Para medir la influencia en I_e de teléfonos manos libres se deberá utilizar los métodos de medida propuestos en la Recomendación [46] de la ITU-T.

C.3. Medida del Retardo en Terminales basados en PC

Se pueden diferenciar dos tipos distintos de terminales en VoIP:

- Terminales digitales: se pueden implementar como programas software que se ejecutan en un PC (terminales basados en PC) o como dispositivos hardware (tarjetas o teléfonos IP) .

158 APÉNDICE C. CÁLCULO DE LOS PARÁMETROS DE ENTRADA AL MODELO-E

Pérdida de Paquetes (%)	G.729-A	G.723.1	GSM EFR	G.711	G.711 PLC Aleat.	G.711 PLC Ráfaga
0	11	15	5	0	0	0
0.5	13	17	-	-	-	-
1	15	19	16	25	5	5
1.5	17	22	-	-	-	-
2	19	24	21	35	7	7
3	23	27	26	45	10	10
4	26	32	-	-	-	-
5	-	-	33	55	15	30
8	36	41	-	-	-	-
10	-	-	-	-	25	40
16	49	55	-	-	35	45

La duración de la Trama G.711 es 10 ms. En G.729-A se utilizan 2 tramas por paquete y en G.723.1 y GSM EFR 1 trama por cada paquete

Cuadro C.3: Factor de degradación debido a la pérdida e paquetes I_e^{PL}

De acuerdo con el sistema de audio utilizado, los terminales digitales a su vez, se pueden clasificar en :

- terminales de microteléfono,
 - terminales de auriculares con micrófono incorporado
 - terminales con altavoces y micrófono (también denominados: manos libres).
- Adaptadores para terminales analógicos tradicionales.

Como parte de las especificaciones del producto el fabricante debe proporcionar el retardo de su equipo con los diversos codecs. Cuando los clientes están constituidos por equipos PC tal especificación resulta difícil de ofrecer, pues la eficiencia depende de las características y carga del equipo. A continuación se ofrece un método que permite obtener el retardo de los equipos terminales basados en PC, así como una serie de medidas que pueden servir como referencia.

El escenario está constituido por dos equipos terminales basados en PC (equipos A y B). A fin de minimizar la influencia de la red, ambos equipos se encuentran conectados mediante una red IP simplificada (un cable cruzado UTP uniendo las dos tarjetas de red). Un tercer equipo se encargará de emitir la señal de voz de entrada a un cliente y de grabar dicha señal junto con la recibida por el otro cliente, tal y como se describe en la figura C.1. Las ganancias han sido ajustadas a un nivel tal que las potencias de la señal emitida y recibida sea la misma. Los codecs y

APÉNDICE C. CÁLCULO DE LOS PARÁMETROS DE ENTRADA AL MODELO-E 159

Componentes Hardware	
Terminales H.323	2 PC (A y B) PIII, 650 Mhz, 256 MB RAM (carga media 20 %).
Sistema de Sonido	Sound Blaster 16 card (Creative Labs). Line Jack card (QuickNet Technologies)
Grabación PC (C)	PI 166Mhz, 64MB RAM.

Componentes Software	
Sistema Operativo	Windows98
Clientes Software	Microsoft NetMeeting y OpenPhone
Codecs soportados	G-711, GSM 06-10, G723-1, G729

Cuadro C.4: Componentes Utilizados

los clientes utilizados, son idénticos en ambos terminales, lo que permite determinar el retardo de un sólo terminal.

Como resultado del experimento se obtiene un fichero estéreo donde cada señal se encuentra retrasada una cantidad equivalente al retardo boca-oido (figura C.2). Mediante una correlación entre ambas señales se puede averiguar con exactitud dicho retardo.

Cada medida del retardo se debe repetir 10 veces, de acuerdo con [41].

Medidas sobre diferentes tipos de terminales ¹:

Los equipos y componentes utilizados se muestran en el cuadro C.4. El tamaño de la cola para compensar la variación del retardo se ajustó a 50 ms (suficiente para compensar las variaciones de retardo de la red IP utilizada).

En el cuadro C.5 se reflejan los resultados obtenidos. Las dos últimas columnas indican la media y la varianza de los retardos. También se midió el tiempo de establecimiento de conexión (columna *Tiempo Estab*), por ser este un parámetro que influye en la calidad, tal y como indica en [47]).

En la columna *BW* se muestra el ancho de banda (régimen binario) consumido por los clientes en función del codec (incluyendo los 40 bytes/paquete de las cabeceras).

Por último la columna *Retardo Mín* muestra el mínimo retardo teórico posible teniendo en cuenta tan sólo el retardo de predicción del codec, el tamaño de la cola para compensar la variación del retardo y el número de tramas por paquete. Así para el codec G.723.1 con 3 tramas por paquete y cuyo tiempo de predicción es de 7,5 ms, tendremos:

$$Retardo_{min} = 3 * 30ms + 7,5ms + 50ms = 147,5ms$$

Para calcular el retardo de un cliente se debe restar al valor medio de la tabla el retardo mínimo teórico (columna *Retardo min*). El resto será debido a los dos clientes. Asumiendo que el

¹Los parámetros utilizados para cada codec son: para G.711 una trama de 30 ms, para G.723.1 un total de 3 tramas por paquete (fpp), para G.729 y GSM 06.10 FR se utilizaron 6 f. p y 4 f. p respectivamente.

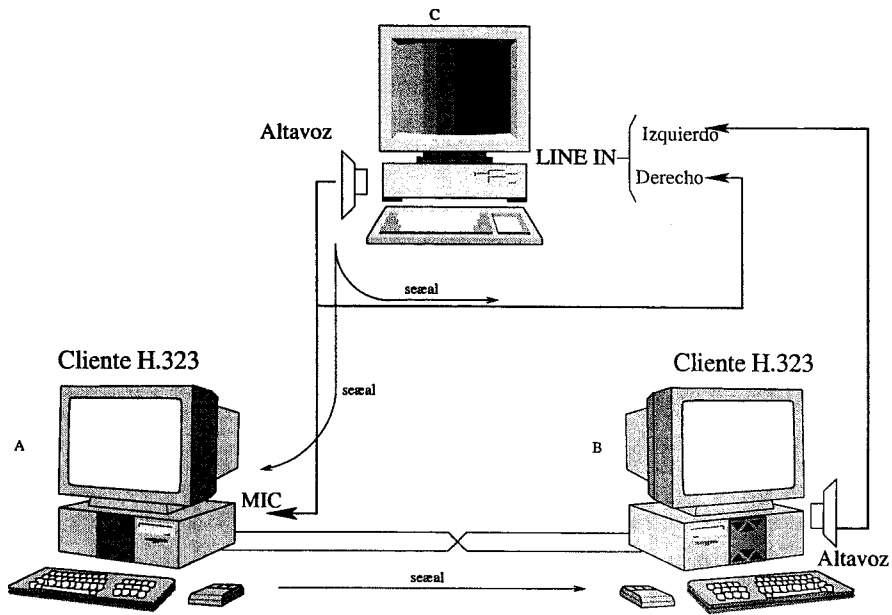


Figura C.1: Medidas de retardo en los terminales

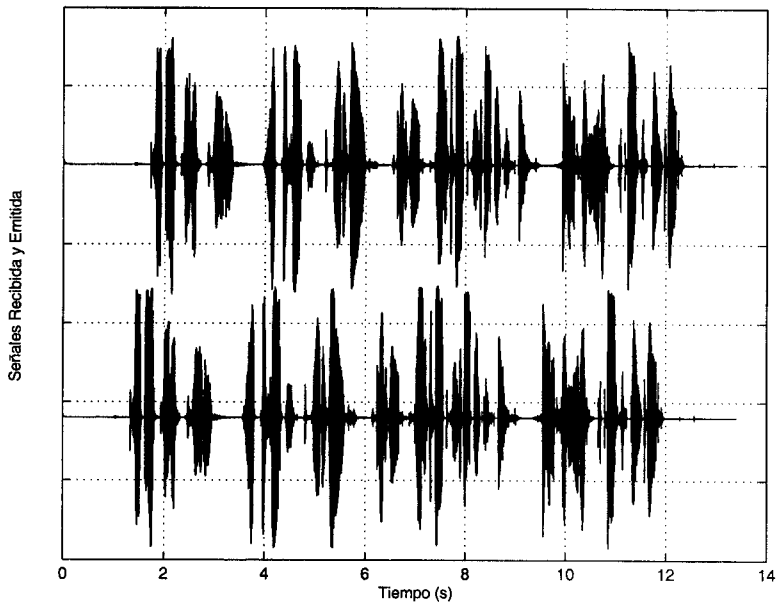


Figura C.2: Señal recibida y emitida tras atravesar la red.

APÉNDICE C. CÁLCULO DE LOS PARÁMETROS DE ENTRADA AL MODELO-E 161

	CODEC	BW(kbps)	Retardo mín. (ms)	Tiempo estab. (s)	μ (ms)	σ (ms)
Codecs	G.711(30ms)	74.67	80	1.215	670	20
Software	GSM 06.10	17.2	130	1.209	550	3.3
	G.723.1	9.96	147.5	1.231	771	19.3
Codecs	G.711(30ms)	96	80	2.416	99	15.9
Hardware	G.723.1	9.96	147.5	1.282	341	2.4
	G.729	13.33	115	1.905	180	2.7
	GSM 06.10	17.2	130	1.265	208	5.6
OP vs NM	G.711	74.67	80	0.213	668	65.3
	G.723.1	9.96	147.5	0.232	587	97
NM vs OP	G.711	74.67	80	0.258	763	2.4
	G.723.1	9.96	147.5	0.238	695	0.5

Cuadro C.5: Resultados del Experimento

retardo en emisión es el doble que en recepción (o bien calculando esta proporción con los datos de [48]) es posible hallar el retardo de un determinado cliente. De esta forma, el retardo del cliente Hardware con el codec G.723.1 sería:

$$Retardo_{terminal} = 341ms - Retardo_{min} = 193,5ms$$

Por lo que el retardo de un cliente en emisión será:

$$Retardo_{Emisión} = Retardo_{terminal} / 3 * 2 = 129ms$$

mientras que en recepción sería de:

$$Retardo_{Recepción} = Retardo_{Ambos} - Retardo_{Emisión} = 64,5ms$$

Apéndice D

Interfaz de Gestión

D.1. Introducción

Todos los servidores implementados generan un fichero de registro (traza de los eventos sucedidos desde el inicio del servicio), para facilitar al administrador las labores de monitorización. Algunos servidores mantienen además información adicional interesante para el control del sistema y el depurado de las aplicaciones, como es el caso del TRM y los datos asociados a los flujos de medios establecidos en el dominio de transporte.

Con el fin de poder acceder a esta información de forma centralizada y remota, se acometió el diseño de una interfaz de gestión o consola donde el administrador del sistema pudiera consultar estos datos. Para la gestión remota se suele emplear en la actualidad alguna de estas dos alternativas:

- La implementación en cada servidor de un agente SNMP, usando como consola un gestor estándar.
- La creación de una interfaz de gestión Web.

En este trabajo, se eligió crear la interfaz de gestión en forma de página Web por dos motivos principales: portabilidad y costes. En concreto:

- Cualquier plataforma dispone de variedad de navegadores web de calidad comercial y gratuitos.
- El modelo Web proporciona además la posibilidad de crear una interfaz gráfica amigable y adaptada al problema que se pretende resolver.
- Por último, la utilización de lenguajes muy especializados, como HTML y JavaScript, minimiza el coste de desarrollo y maximiza la flexibilidad, facilitando la modificación o ampliación de la herramienta.

La implantación de la gestión se ha realizado en dos fases. La primera, descrita en este apartado, aborda la gestión de los elementos del plano de transporte. La segunda se encargará de extender este modelo de gestión al plano de aplicación.

D.2. Diseño

En esta sección se realizará una descripción del diseño realizado. El interfaz de gestión responde al modelo cliente – servidor, por lo que especificaran ambas partes por separado.

D.2.1. Parte Servidor

La parte servidor de la interfaz de gestión consiste en el conjunto de herramientas que permiten recuperar la información de registro almacenada por los servidores, para proporcionársela al cliente. En el caso del TRM, además, es necesario recuperar la información de estado que éste mantiene.

Esta información será servida al cliente en forma de página Web; una por cada servidor de QoS monitorizado. Para esta tarea, en principio se barajaron dos opciones:

1. Que fueran los propios servidores de QoS los que atendieran las peticiones, añadiéndoles soporte de protocolo HTTP.
2. Que fueran aplicaciones CGI o de script, ejecutadas por un servidor Web estándar, las que atendieran estas peticiones.

La primera solución es menos restrictiva, en el sentido de que no es necesaria la instalación de un servidor Web. Sin embargo, la segunda es más flexible y reduce la complejidad del desarrollo, y teniendo en cuenta la libre disponibilidad del servidor *Apache* para las plataformas donde se ejecuta el modelo (Linux y Windows), fue el método elegido. Es por este motivo que el servidor TRM exporta su información de estado a un fichero, de forma que pueda ser recuperada por estas aplicaciones externas.

Como lenguaje para crear las páginas que serán servidas, se eligió PHP, por su potencia, su natural integración con HTML y el completo soporte con que cuenta en Apache. En la actualidad, los servidores TPE y QoSPE no generan información de registro interesante, de manera que sólo se ha creado un script PHP, que sirve la información de registro y estado correspondiente al servidor TRM (El script que servirá la información correspondiente al QoSM está en desarrollo).

Visor de registro

La tarea principal de los scripts PHP de lado servidor es recuperar la información disponible en el fichero de registro del servidor al que atiendan, y darle el formato adecuado para presentarla al cliente. El formato elegido es simplemente un área de texto, donde se vuelca la información.

El fichero de registro puede ser excesivamente grande para ser transmitido en su totalidad, de manera que sólo se vuelca una parte del mismo. Para determinar esta parte, se usan tres variables:

- Un tamaño máximo de bloque a leer.
- Un índice de desplazamiento, que marca un punto de referencia dentro del archivo.
- Una dirección de lectura, que indica si el índice es inicio o final de bloque.

Un par de botones, “Adelante” y “Atrás”, serán utilizados para recargar la página indicándole al código PHP que cargue un nuevo bloque, respectivamente posterior o anterior al actual, dándole nuevos valores a estas variables; los desplazamientos inicial y final del bloque leído estarán disponibles como variables JavaScript en la página generada.

Para aprovechar al máximo cada consulta, se intentará leer siempre el tamaño de bloque indicado completo. Por este motivo, el offset índice es orientativo; los puntos inicial o final de lectura podrán variar si el índice está muy próximo al principio o al final del archivo, como se ilustra en las figuras D.1 y D.2

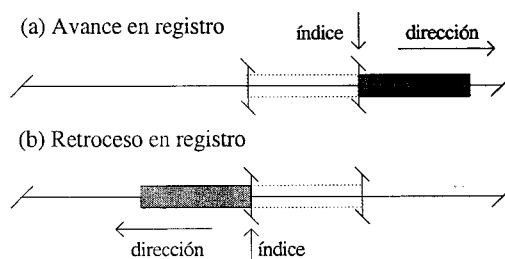


Figura D.1: Lectura de bloque sin reajuste

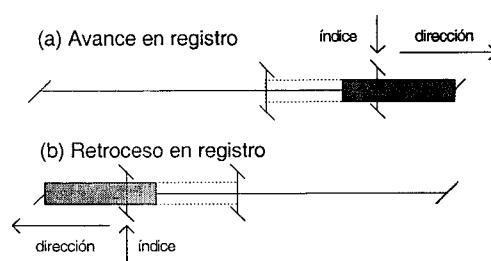


Figura D.2: Lectura de bloque con reajuste

Visor de estado

En el caso del TRM, el script PHP mostrará también la información asociada a los flujos establecidos en el dominio de transporte. Para recuperar esta información del fichero de estado,

en formato binario, y convertirla a modo texto, se ha empleado una herramienta realizada en C++, cuya salida integra el código PHP directamente en la página.

La información de estado se muestra a modo de formulario, lo que permite al administrador modificar los recursos asignados a un flujo, o liberarlos, remitiendo el formulario al servidor. Para soportar esta funcionalidad, se ha creado un cliente de interfaz QT2 de línea de comandos, capaz de enviar solicitudes `QT2.TRMQreq` y `QT2.TRMQrelreq` al TRM. El aspecto del formulario se muestra en la figura D.3.

The image shows a web-based status viewer interface. At the top, there are two buttons: 'Atras' (Back) and 'adelante' (Forward). Below these is a form with the following fields:

- Origen**: Input field with a vertical separator.
- Destino**: Input field with a vertical separator.
- Protocolo**: Input field.
- Retardo**: Input field with '(ms)' to its right.
- Jitter**: Input field with '(ms)' to its right.
- Pérdida**: Input field with '(%)' to its right.
- Ancho de banda**: Input field with '(centenas de bps)' to its right.
- Máxima PDU**: Input field with '(centenas de bit)' to its right.

At the bottom of the form, there are two radio buttons: 'A adquisición' (selected) and 'Liberación'. To the right of these is a button labeled 'Realizar petición'.

Figura D.3: Visor de estado

D.2.2. Parte Cliente

A fin de ofrecer una interfaz única, las páginas de información correspondientes a cada servidor se integran como marcos (*frames*) dentro de una página principal. Esta página incluye el código JavaScript necesario para procesar y formatear la información devuelta por los scripts PHP, y presentarla al administrador. El esquema de la parte cliente puede verse en la figura D.4.

Según se muestra en ella, la página principal está dividida en dos marcos: un marco de configuración, y otro de monitorización.

- La información generada por los servidores es susceptible de cambiar en cualquier momento, y por tanto ha de ser refrescada periódicamente. El periodo de refresco, así como cualquier otro tipo de información de configuración, se mantendrá en un marco propio, el

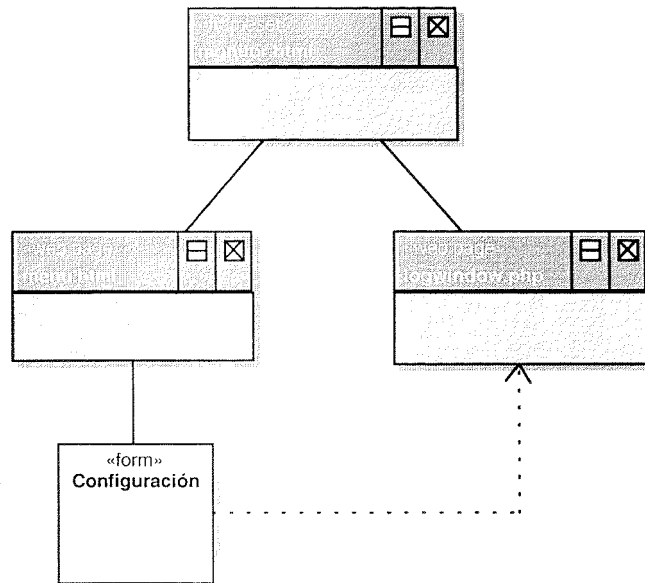


Figura D.4: Interfaz del monitor Web

marco de configuración, donde estas variables estarán almacenadas y serán accesibles al usuario, a través de un formulario.

- Las páginas web correspondientes a los servidores de QoS controlados se muestran en el marco de monitorización. Éste será a su vez dividido en tantos marcos como servidores monitorizados (únicamente el TRM hasta que se complete el desarrollo de la página del QoS).

El marco de monitorización es recargado periódicamente, según lo configure el usuario de la interfaz; cualquier información de configuración o estado que necesite almacenar entre las sucesivas recargas se mantiene como variables JavaScript en el marco de configuración, haciéndose accesibles al usuario a través de un formulario, si es conveniente.

D.3. Implementación

El aspecto de la consola puede verse en la figura D.5. El marco de la izquierda contiene el menú de selección, y el de la derecha, la ventana correspondiente al servidor TRM, donde se muestra su información de registro y de estado.

Dentro de los parámetros configurables, además del periodo de refresco, se encuentra el tamaño del bloque de cada fichero de registro que se volcará (en KBytes), el número de flujos de medios sobre los que se mostrará información, etc.

El marco de monitorización correspondiente a cada servidor consiste en un área de texto

Location: http://localhost/trmdump/monitor.html

Actualización
 manual
 cada 0 segundos

Visor de estado
 Mostrar 10 flujos

Visor de registro
 cargar 2 KBytes

Debug [v] [Atrás] [Adelante]

Event (1) Date: Sat Sep 15 16:49:35 2001
 Petición TRMQ procesada
 En mensaje (1) de -CTRMNet client -
 Aceptado añadir flujo: 2.3.4.5.6000 --> 5.6.7.8.9000 udp
 Ancho de banda: 0 (100 bps)
 Tamaño máximo de PDU: 10000 (100 bits)
 Delay, Jitter: 100, 0 (ms)
 Pérdida de paquetes: 0.9 %

Event (2) Date: Sat Sep 15 16:50:48 2001
 Petición TRMQ procesada

[Atrás] [Adelante]

Origen	2.3.4.5	:	6000
Destino	5.6.7.8	:	9000
Protocolo	udp		
Retardo	100		(ms)
Jitter	0		(ms)
Pérdida	0.9		(%)

Figura D.5: Consola de monitorización

en formato binario, y convertirla a modo texto, se ha empleado una herramienta realizada en C++, cuya salida integra el código PHP directamente en la página.

La información de estado se muestra a modo de formulario, lo que permite al administrador modificar los recursos asignados a un flujo, o liberarlos, remitiendo el formulario al servidor. Para soportar esta funcionalidad, se ha creado un cliente de interfaz QT2 de línea de comandos, capaz de enviar solicitudes **QT2.TRMQreq** y **QT2.TRMQrelreq** al TRM. El aspecto del formulario se muestra en la figura D.3.

The image shows a web-based configuration form for a state viewer. At the top, there are two buttons: 'Atrás' (Back) on the left and 'Adelante' (Forward) on the right. Below these is a form with the following fields:

- Origen**: A text input field.
- Destino**: A text input field.
- Protocolo**: A text input field.
- Retardo**: A text input field with '(ms)' to its right.
- Jitter**: A text input field with '(ms)' to its right.
- Pérdida**: A text input field with '(%)' to its right.
- Ancho de banda**: A text input field with '(centenas de bps)' to its right.
- Máxima PDU**: A text input field with '(centenas de bit)' to its right.

At the bottom of the form, there are two radio buttons: 'Adquisición' (selected) and 'Liberación'. To the right of these is a button labeled 'Realizar petición'.

Figura D.3: Visor de estado

D.2.2. Parte Cliente

A fin de ofrecer una interfaz única, las páginas de información correspondientes a cada servidor se integran como marcos (*frames*) dentro de una página principal. Esta página incluye el código JavaScript necesario para procesar y formatear la información devuelta por los scripts PHP, y presentarla al administrador. El esquema de la parte cliente puede verse en la figura D.4.

Según se muestra en ella, la página principal está dividida en dos marcos: un marco de configuración, y otro de monitorización.

- La información generada por los servidores es susceptible de cambiar en cualquier momento, y por tanto ha de ser refrescada periódicamente. El periodo de refresco, así como cualquier otro tipo de información de configuración, se mantendrá en un marco propio, el

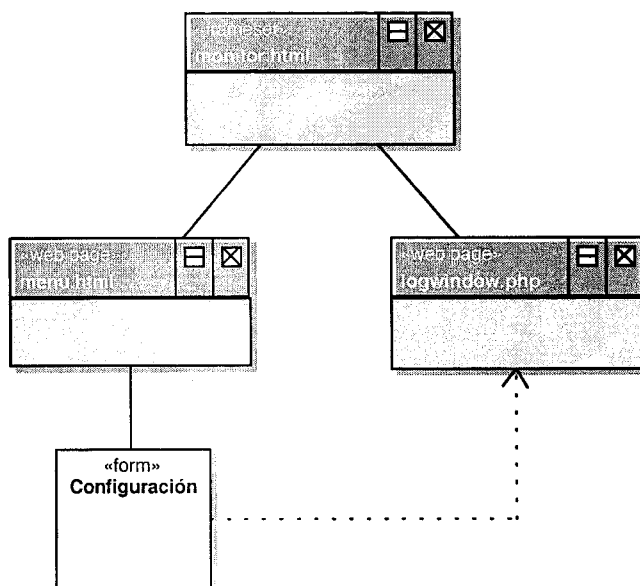


Figura D.4: Interfaz del monitor Web

marco de configuración, donde estas variables estarán almacenadas y serán accesibles al usuario, a través de un formulario.

- Las páginas web correspondientes a los servidores de QoS controlados se muestran en el marco de monitorización. Éste será a su vez dividido en tantos marcos como servidores monitorizados (únicamente el TRM hasta que se complete el desarrollo de la página del QoSM).

El marco de monitorización es recargado periódicamente, según lo configure el usuario de la interfaz; cualquier información de configuración o estado que necesite almacenar entre las sucesivas recargas se mantiene como variables JavaScript en el marco de configuración, haciéndose accesibles al usuario a través de un formulario, si es conveniente.

D.3. Implementación

El aspecto de la consola puede verse en la figura D.5. El marco de la izquierda contiene el menú de selección, y el de la derecha, la ventana correspondiente al servidor TRM, donde se muestra su información de registro y de estado.

Dentro de los parámetros configurables, además del periodo de refresco, se encuentra el tamaño del bloque de cada fichero de registro que se volcará (en KBytes), el número de flujos de medios sobre los que se mostrará información, etc.

El marco de monitorización correspondiente a cada servidor consiste en un área de texto

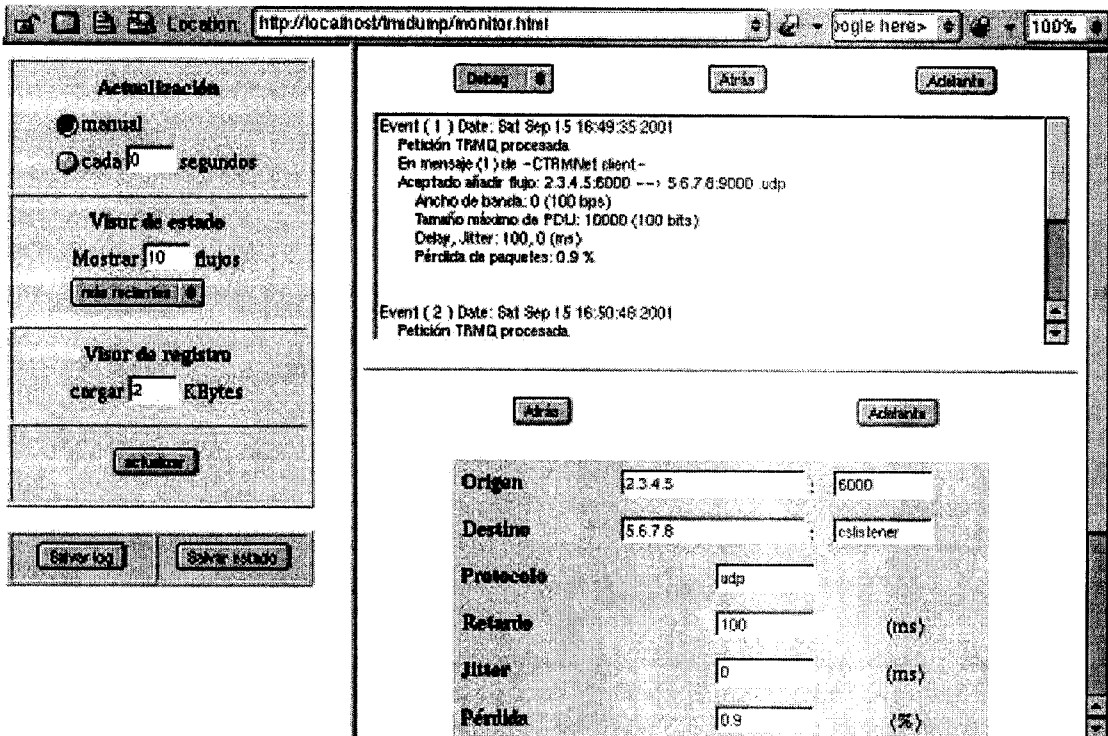


Figura D.5: Consola de monitorización

donde se muestra parte del fichero de registro, junto con un par de botones que permiten retroceder o avanzar en el mismo. Excepto en el caso del QoSM, donde el fichero de registro hereda el formato del usado por el Gatekeeper que le da soporte, el resto de aplicaciones estructuran el fichero de forma que es posible distinguir distintos niveles de urgencia dentro de un evento, y filtrar usando ese criterio la información que será mostrada en el área de texto.

En el caso del TRM, además de esto se muestra información acerca de los flujos establecidos en el dominio de transporte; Estos datos se muestran a modo de formulario, siendo posible modificar los parámetros de QoS asignados a un flujo, o liberar la reserva, remitiendo el formulario al servidor. También es posible mostrar estos datos en forma de tabla, de manera que no sean modificables.

Apéndice E

Librerías

En la implementación del modelo propuesto descrita en los anexos previos se han utilizado varias librerías existentes. En este apéndice se realizará una introducción a dichas librerías, a fin de facilitar la comprensión del diseño y las implementaciones realizadas.

E.1. PWLib

La librería PWLib (*Portable Windows Library*) nació con el objetivo de ofrecer un conjunto de clases que elevasen una capa de abstracción sobre todos los mecanismos del sistema operativo y del entorno, y permitiera realizar programas en C++ completamente portables entre las plataformas Unix, Mac y Windows.

Como parte de este proyecto, se ha desarrollado una cantidad ingente de software, que facilita el desarrollo de ciertas tareas, y sobre todo se han implementado algunas herramientas muy útiles. Entre ellas está un traductor de ASN.1 a C++ que ha simplificado enormemente la tarea de construir el código necesario para manipular y transmitir todos los objetos ASN.1 definidos. Frente a otros traductores disponibles en el mercado, éste es el único con la ventaja de estar acogido a licencias de código abierto, siendo completamente gratuito.

E.1.1. Estructura de aplicaciones PWLib

Una de las principales objeciones que puede poner un programador a esta librería es su carácter “invasivo”. La diferencia entre los estilos de programación Unix y Windows es tan grande, que la librería se ve en la necesidad de imponer un marco de aplicación (*Application Framework*) muy rígido, que abarca a todos los aspectos del programa, desde el código fuente hasta los *Makefiles*. Esto no es negativo en sí mismo, pues existen múltiples librerías que utilizan marcos de aplicación, como la *Microsoft Foundation Classes*, o la *Object Windows Library* de Borland/Inprise. Sin embargo, la – prácticamente – nula documentación de este entorno, que por otra parte es bastante complejo, hace que la curva de aprendizaje del paquete sea extremadamente elevada.

Documentar esta librería queda muy alejado de los objetivos de la presente memoria, de manera que este capítulo se limitará a una descripción somera de la misma.

El conjunto de clases que ofrece la librería PWlib se divide en tres grupos:

Containers – Contenedores: Tablas, Listas, Diccionarios...

La librería PWlib está preparada para ofrecer contenedores genéricos en compiladores que no admiten el uso de patrones (*templates*). Para ello, define un conjunto de clases base que implementan listas, diccionarios, etc, utilizando punteros, y las usa para construir sobre ellas todo su entramado de contenedores. Con el soporte del nuevo estándar de C++ en la mayoría de los compiladores actuales, puede resultar más cómodo utilizar los tipos de la STL¹.

I/O – Entrada/Salida: Colas, Pipes, Ficheros, Sockets...

Este tipo de objetos es el más ampliamente empleado en las aplicaciones. La jerarquía de clases y el conjunto completo de métodos que ofrece la PWLib para el manejo de sockets aparece en la figura E.1. Como se aprecia en dicha figura, aparte de las funciones básicas como abrir, cerrar, conectar, poner a la escucha, aceptar, etc, se incluyen una multitud de funciones adicionales.

Threads & Processes – Hilos y Procesos: Hilos, Procesos de usuario, Procesos de Servicio...

La clase *PThread* representa a un hilo de ejecución. La librería PWlib utiliza multitarea preemptiva en aquellos sistemas que la soportan (Windows NT, sistemas POSIX – pthreads –), y multitarea cooperativa en el resto. En este segundo caso, los hilos de ejecución no son tales, sino que se simula la ejecución de múltiples hilos mediante una cesión explícita del flujo de ejecución de un hilo a otro mediante llamadas a la función *PThread::Yield*.

También se ofrecen objetos de sincronización como semáforos y *mútexes*, basados en los anteriores. Sin embargo, este aspecto está ligeramente descuidado, pues no se contemplan necesidades como el establecimiento de prioridades relativas entre hilos, y se echa en falta un mecanismo similar a las *variables condición* de POSIX [49].

PWlib también ofrece facilidades para el soporte de interfaces gráficas de usuario portables, si bien, teniendo en cuenta que las aplicaciones desarrolladas en esta Tesis no tienen una interfaz gráfica muy compleja, esta categoría de utilidades no ha sido estudiada en suficiente profundidad como para ser expuesta aquí.

E.1.2. Traductor ASN.1

El elemento determinante en la elección de la librería PWlib como base para la implementación del modelo propuesto fue la existencia de esta herramienta. Este programa traduce el código ASN.1 a lenguaje PWlib-C++, siendo capaz de procesar niveles de complejidad tan elevados como los necesarios para traducir las especificaciones de H.323, que definen miles de tipos de datos.

¹Standard Template Library

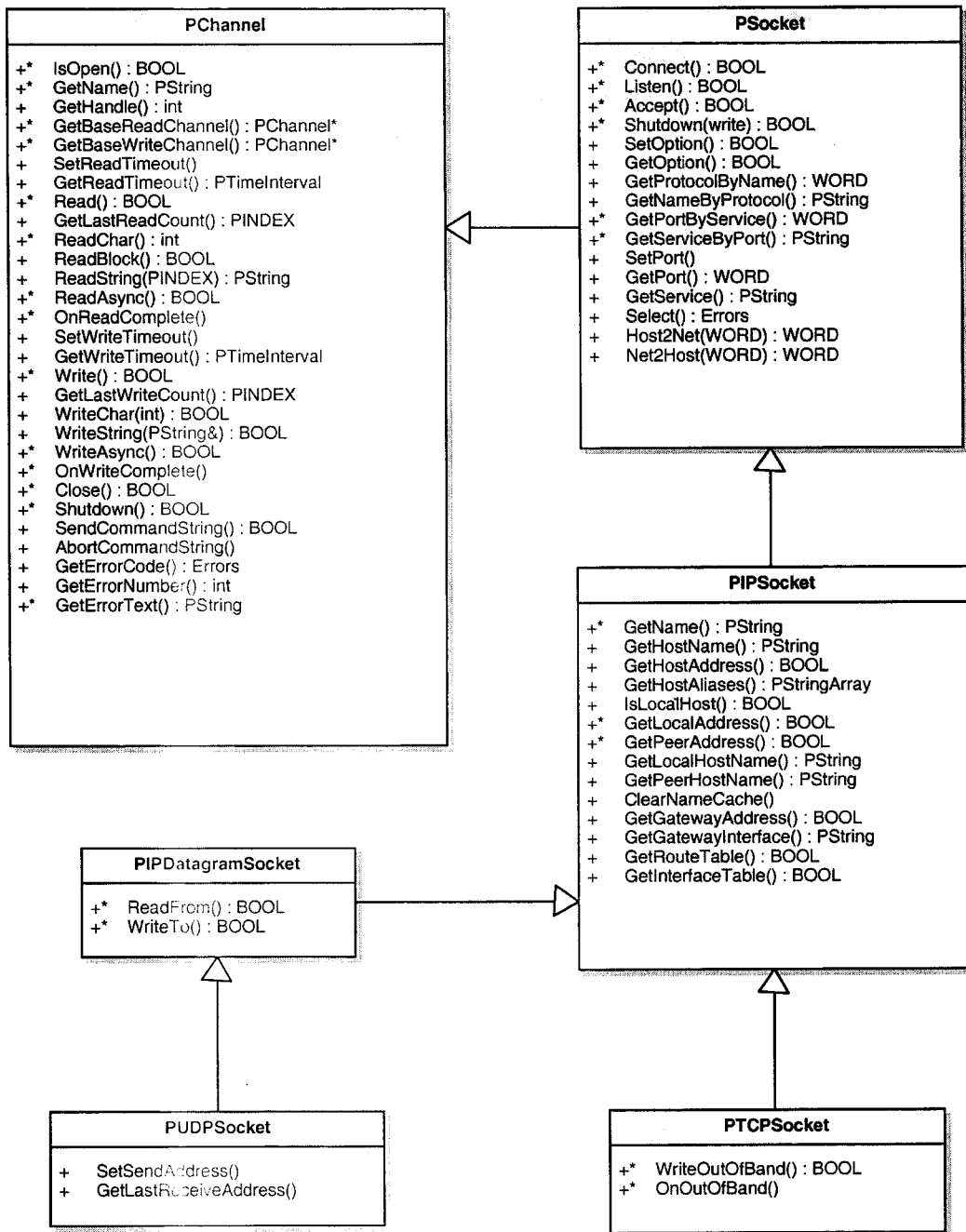


Figura E.1: Jerarquía de clases para el manejo de sockets

El proceso de traducción utiliza un grupo de tipos básicos de ASN.1, definidos mediante clases C++; esta base se muestra en la figura E.2, donde se representan los tipos más elementales, obviando otros como *BMPString* o *UniversalDate* que son también estándar de ASN.1, pero consisten en realidad en especializaciones del tipo *OctetString*.

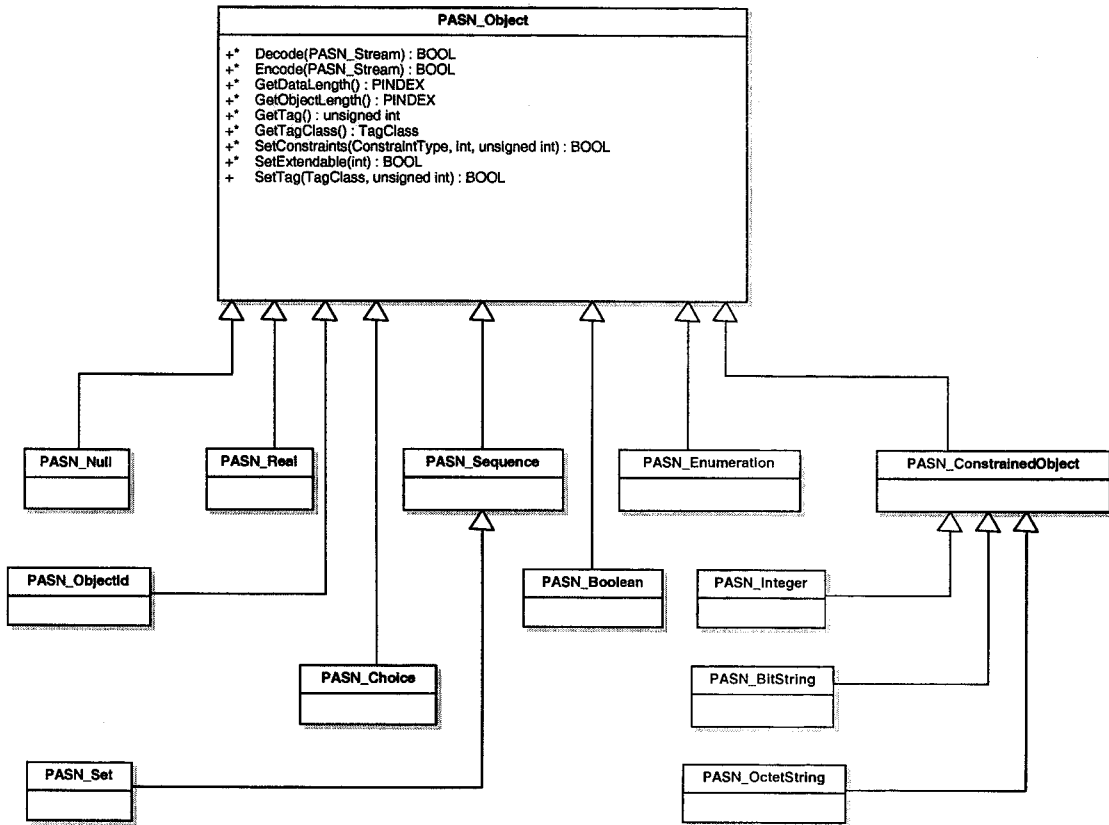


Figura E.2: Jerarquía de clases empleada por el parser ASN.1

Las estructuras definidas mediante ASN.1, y traducidas por el programa, se convierten al pasar a C++ a objetos más o menos complejos que hacen uso de estas clases para representar cada uno de sus campos simples. El traductor se encarga de la definición del tipo, del ajuste de su rango caso de ser un tipo restringido (como en el caso de los números de puerto, definidos como enteros en el rango 0–65535), y del establecimiento de las etiquetas (*tags*) necesarias.

E.2. OpenH323

OpenH323 es un proyecto que pretende facilitar la implantación de la tecnología VoIP desarrollando librerías y aplicaciones en código abierto que sirvan de apoyo para la construcción de programas conformes al estándar H.323. Ofrece todas las definiciones ASN.1 de los tipos de datos

relacionados con la especificación H.323 en formato electrónico, listas para ser procesadas por el traductor ASN.1 de PWlib y empleadas.

Dentro del software que acompaña a esta librería, se incluye un cliente de voz H.323 acogido a licencia de código abierto, sobre el cual se ha construido la implementación del cliente de voz con QoS. La estructura de la aplicación, y de las librerías, se describe en el apartado dedicado al cliente de voz (B.1), de manera que no se profundizará más sobre ella aquí.

Existe un amplio abanico de programas de código abierto que utiliza estas librerías; de entre ellos, como base para implementar el QoS, se eligió el programa OpenGate. Éste implementa un Gatekeeper H.323 con capacidad de enrutado de señalización H.225.0 CS y H.245.0 CTRLS, imprescindible para la implementación del modelo. De nuevo, se ha dedicado un apartado a su estructura y las modificaciones realizadas para añadir soporte de QoS (B.2), en el que se analiza el mismo con mayor detalle. Es posible encontrar más información sobre estas librerías en <http://www.openh323.org>.

E.3. NISTNet

NISTNet es un paquete de emulación de red que se ejecuta en plataformas Linux, permitiendo que un único PC actuando como encaminador pueda emular una amplia variedad de condiciones de red [50].

El paquete NISTNet es una herramienta de propósito general para emular dinámicas de rendimiento en un entorno IP. La herramienta ha sido diseñada para permitir la realización de experimentos controlados y reproducibles con aplicaciones sensibles o adaptables al comportamiento de la red, en un entorno simple de laboratorio. NISTNet puede emular las características críticas de rendimiento extremo a extremo que se producen en diversas situaciones, en redes de área ancha o debido al uso de ciertas tecnologías de subred (por ejemplo, la asimetría del ancho de banda con xDSL o módems de cable).

E.3.1. Características

Las posibilidades de emulación de este paquete son bastante amplias, permitiendo ajustes del retardo sufrido por los paquetes que atraviesan el encaminador, simulación de congestión y ajuste de la probabilidad de pérdida, limitación del ancho de banda disponible, y des secuenciación y duplicación de paquetes. Todos estos parámetros pueden controlarse de forma independiente para diversos flujos o agregados de flujos.

NISTNet se instala como un módulo del kernel de Linux. Una vez cargado, el emulador puede encontrarse en dos estados: activo o inactivo. Añadiendo un descriptor de flujo a las tablas de configuración del módulo, es posible instruirlo para que dispense un trato específico a los flujos que concuerden con ese descriptor, una vez que el emulador sea activado.

La comunicación de las aplicaciones con el módulo se realiza a través de un par de dispositivos especiales (entradas en */dev*). Mediante la escritura en estos dispositivos, es posible fijar

las tablas de probabilidad de la distribución estadística que se empleará para retardar o perder los paquetes. La activación, recogida de estadísticas, y adición o eliminación de nuevos descriptores de flujo en las tablas del módulo, se realiza a través de *ioctls* al dispositivo. NISTNet incluye en un conjunto de herramientas de línea de comandos y un programa monitor que se ejecuta en X-Windows, para facilitar la tarea del administrador. La distribución básica de NISTNet viene acompañada de documentación sobre el empleo de estas herramientas; también es posible encontrar información en <http://snad.ncsl.nist.gov/nistnet>.

E.4. Netfilter

El subsistema *netfilter* es una completa actualización de las precedentes implementaciones de filtrado de paquetes del núcleo de Linux, *ipfwadm* e *ipchains*. *Netfilter* permite al administrador establecer reglas para la manipulación de los paquetes IP conforme van atravesando diversas partes del núcleo. Este apartado se centra en la descripción de las características de filtrado de paquetes convencional; para características más avanzadas, se recomienda consultar [51].

E.4.1. Arquitectura netfilter

La serie 2.4 del *kernel* de Linux incluye un nuevo sistema de filtrado, *netfilter*, que viene a sustituir a los anteriores sistemas *ipchains* e *ipfwadm*. En esta arquitectura, las cadenas de filtrado se agrupan en tablas. Cada tabla está orientada a una tarea distinta.

Existen tres tablas predefinidas, cada una de las cuales puede estar o no cargada en el módulo en un determinado instante: tabla de filtrado (*filter*), tabla de NAT² y tabla de enredado (*mangling*). Cada una de estas tablas contiene un conjunto de cadenas también predefinidas, aunque pueden contener cadenas de usuario.

En la implementación realizada, el cortafuegos se limita a actuar como filtro de paquetes, de forma que se trabaja únicamente con la tabla de filtrado, en concreto con la cadena predefinida *FORWARD*, que especifica las políticas de filtrado a administrar a los paquetes que atraviesen el nodo como encaminador.

E.4.2. Librería IPTClib

Para poder llevar a cabo todas las tareas de gestión del cortafuegos, las herramientas de línea de comandos incluidas con la distribución básica de IPtables hacen uso de unas librerías, las cuales también se usan en la implementación realizada. El código fuente de las mismas puede encontrarse en <http://www.netfilter.kernelnotes.org>. La documentación relacionada se puede encontrar en http://www.linuxsecurity.com/resource_files/firewalls/iptables-hacking-howto/.

²Network Address Translation

Bibliografía

- [1] IETF RFC 1889, RTP: A Transport Protocol for Real-Time Applications. (01/96)
- [2] ITU-T Recommendation G.121. Loudness ratings (LRs) of national systems. (03/93)
- [3] ITU-T Recommendation P.64. Determination of sensitivity/frequency characteristics of local telephone systems. (09/99)
- [4] J. Janssen, D. Vleeschauwer, G. Petit. Delay and Distortion Bounds for Packetized Voice Calls of Traditional PSTN Quality. IPTEL 2000, (12-13/4/00) Berlin.
- [5] ITU-T Recommendation G.114 . One-Way Transmission Time. (02/96)
- [6] ETSI ETR 250. Speech communication quality from mouth to ear for 3,1 kHz handset telephony across networks. (7/96).
- [7] ITU-T Recommendation P.831. Subjective performance evaluation of network echo cancellers. (12/98)
- [8] ITU-T Recommendation G.169 . Automatic Level Control Devices. (06/99)
- [9] ITU-T Recommendation G.168 . Digital Network Echo Cancellers. (02/96)
- [10] ITU-T Recommendation P.830. Subjective performance assessment of telephone-band and wideband digital codecs. (02/96)
- [11] O. Hersent, D. Gurle , J. P. Petit. IP Telephony. Addison-Wesley ISBN 0-201-61910-5.
- [12] ITU-T Recommendation G.113 Appendix I. Provisional Planning Values for the Equipment Impairment Factor Ie. (09/99)
- [13] ITU-T Recommendation P.800. Methods for subjective determination of transmission quality. (08/96)
- [14] ETSI EG 201 377-1 v1.1.1. Speech Processing, Transmission and Quality Aspects (STQ);Specification and measurement of speech transmission quality;Part 1: Introduction to objective comparison measurement methods for one-way speech quality across networks. (04/99)

- [15] ITU-T Recommendation P.861. Objective quality measurement of telephone-band (300-3400 Hz) speech codecs. (02/98)
- [16] ITU-R Recommendation BS.1387. Method for objective measurement of telephone-band (300-3400 Hz) speech codecs. (01/99)
- [17] ITU-T Recommendation P.862. Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs (02/01)
- [18] A. Rix, J. Beerends, M. Hollier, A. Hekstra. PESQ - the new ITU standard for end-to-end speech quality assessment. AES 109th CONVENTION. LOS ANGELES, 2000 SEPTEMBER 22-25.
- [19] CCITT Supplement No 3. Models for predicting Transmission Quality from objective Measurements. (1988)
- [20] ITU-T Recommendation G.107. The E-Model, a computational model for use in transmission planning. (05/00)
- [21] ETSI EG 201 050. Speech Processing, Transmission and Quality Aspects (STQ); Overall Transmission Plan Aspects for Telephony in a Private Network. (02/99)
- [22] Toga, J. , Ott J. ITU-T Standardization Activities for Interactive Multimedia Communications on Packet Networks: H.323 and Related Recommendations. Computer Networks, Vol. 31 (1999) 20–23.
- [23] ITU-T Recommendation H.323. Packet-based Multimedia communications systems. (11-2000)
- [24] ITU-T Recommendation H.225.0. Call signalling protocols and media stream packetization for packet based multimedia communications systems. (1999).
- [25] ITU-T Recommendation H.245. Control Protocol for multimedia communication. (2000).
- [26] IETF RFC 2543. SIP: Session Initiation Protocol. (3-1999)
- [27] IETF RFC 2327, SDP: Session Description Protocol. (04/98)
- [28] Mathy, L. , Edwards, C. , Hutchison, D. The Internet: A global Telecommunications Solution?. IEEE Network Magazine, Vol 14-4 (2000) 46–57.
- [29] IETF RFC 1633. Integrated Services in the Internet Architecture: an Overview. (6-1994)
- [30] IETF RFC 2212. Specification of Guaranteed Quality of Service. (7-1997)
- [31] IETF RFC 2211. Specification of the Controlled-Load Network Element Service. (7-1997)
- [32] IETF RFC 2205. Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification. (7-1997)

- [33] IETF RFC 2475. An Architecture for Differentiated Services. (12-1998)
- [34] Multiprotocol label Switching (MPLS). The International Engineering Consortium.
- [35] IETF Work in Progress, Berner et al. A Framework For Integrated Services Operation Over Diffserv Networks . (5-2000)
- [36] ETSI TS 101 329, Parte 2. "Telecommunications and internet protocol harmonisation over networks (TIPHON); Definition of Quality of Service (QoS) Classes" (09/00)
- [37] ETSI TS 101 329, Parte 3. "Telecommunications and internet protocol harmonisation over networks (TIPHON); The signaling and control of end-to-end Quality of Service in TIPHON Systems" (09/00)
- [38] Pricing and Provisioning for Guaranteed Internet Services. Zhong Fan. ICN 2001 Proceeding, Part I. July 2001
- [39] Charging schemes for multi-service networks. D. Songhurst and F. Kelly. In ITC 15, 1997.
- [40] ETSI TIPHON Draft TR 101 329-7. Design Guide for Elements of a TIPHON connection from an End-to-End speech transmission performance point of view. (07/00)
- [41] ETSI TIPHON DRAFT TS 101 329-5. Quality of Service (QoS) measurement methodologies. (07/00)
- [42] ITU-T Recommendation P.76. Determination of loudness ratings; fundamental principles. (11/88)
- [43] ITU-T Recommendation P.79. Calculation of Loudness Ratings for Telephone Sets. (9/99)
- [44] ETSI TBR 8. Integrated Services Digital Network (ISDN);Telephony 3,1 kHz teleservice;Attachment requirements for handset terminals
- [45] ITU-T Recommendation P.502. Objective test methods for speech communication systems using complex test signals. (05/02)
- [46] ITU-T Recommendation P.342. Transmission characteristics for telephone band (300-3400 Hz) digital loudspeaking and hands-free telephony terminals. (05/00)
- [47] ETSI Draft TR 101 329-1 v2.4.2. General Aspects of Quality of Service (QoS). (05/00)
- [48] ITU-T Recommendation G.114 Appendix I. One-way transmission time. (05/00).
- [49] IEEE "Portable Operating System Interface (POSIX) Section 1003.1c – PThreads"
- [50] NISTNet es un paquete de emulación de red que se ejecuta sobre Linux. Véase <http://snad.ncsl.nist.gov/nistnet>
- [51] "Linux 2.4 Packet Filtering HOWTO", http://www.linuxsecurity.com/resource_files/firewalls/packet-filtering-howto/.

[52] "IPtables Hacking HOWTO", http://www.linuxsecurity.com/resource_files/firewalls-iptables-hacking-howto/.

UNIVERSIDAD DE SEVILLA

Reunido el Tribunal integrado por los abajo firmantes en el día de la fecha, para juzgar la Tesis Doctoral de D. RAFAEL M^o ESTEPA ALONSO titulada CONTRIBUCIONES AL SOPORTE DE CALIDAD EN REDES DE VOZ SOBRE IP acordó otorgarle la calificación de SOBRESALIENTE CUM LAUDE POR UNANIMIDAD

Sevilla, 24 de JUNIO 2002

El Vocal,

Manuel Alvarado

El Presidente

Manuel Alvarado

El Vocal

Vicente

El Secretario

Terencia de la Cruz

El Vocal,

Antonio

El Doctorado,

Antonio